Lecture Notes in Social Networks

Nima Dokoohaki *Editor*

Fashion Recommender Systems

Springer

Lecture Notes in Social Networks

Lecture Notes in Social Networks (LNSN) comprises volumes covering the theory, foundations and applications of the new emerging multidisciplinary field of social networks analysis and mining. LNSN publishes peer-reviewed works (including monographs, edited works) in the analytical, technical as well as the organizational side of social computing, social networks, network sciences, graph theory, sociology, Semantics Web, Web applications and analytics, information networks, theoretical physics, modeling, security, crisis and risk management, and other related disciplines. The volumes are guest-edited by experts in a specific domain. This series is indexed by DBLP. Springer and the Series Editors welcome book ideas from authors. Potential authors who wish to submit a book proposal should contact Christoph Baumann, Publishing Editor, Springer e-mail: Christoph.Baumann@springer.com

More information about this series at http://www.springer.com/series/8768

Nima Dokoohaki
Editor

Fashion Recommender Systems

Springer

Editor
Nima Dokoohaki
KTH - Royal Institute of Technology
Stockholm, Sweden

ISSN 2190-5428 ISSN 2190-5436 (electronic)
Lecture Notes in Social Networks
ISBN 978-3-030-55217-6 ISBN 978-3-030-55218-3 (eBook)
https://doi.org/10.1007/978-3-030-55218-3

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Acknowledgement

This effort has been possible with the help of the organization team of the first workshop on Recommender Systems in Fashion, namely Shatha Jaradat, Humberto Corona and Reza Shirvany. Editor would like to gratefully acknowledge support of the technical committee of the workshop, namely Steven Bourke, Soude Fazeli, Diogo Goncalves, Karl Hajjar, Nour Karessli, Ralf Krestel, Julia Lasserre, Leonidas Lefakis, Ana Peleteiro, Mirela Riveni, Roberto Roverso, Saúl Vargas and Simon Walk.

Contents

Part I Cold Start in Recommendations

Fashion Recommender Systems in Cold Start 3
Mehdi Elahi and Lianyong Qi

Part II Complementary and Session Based Recommendation

Enabling Hyper-Personalisation: Automated Ad Creative Generation and Ranking for Fashion e-Commerce 25
Sreekanth Vempati, Korah T. Malayil, V. Sruthi, and R. Sandeep

Two-Stage Session-Based Recommendations with Candidate Rank Embeddings 49
José Antonio Sánchez Rodríguez, Jui-Chieh Wu, and Mustafa Khandwawala

Part III Outfit Recommendations

Attention-Based Fusion for Outfit Recommendation 69
Katrien Laenen and Marie-Francine Moens

Outfit2Vec: Incorporating Clothing Hierarchical MetaData into Outfits' Recommendation 87
Shatha Jaradat, Nima Dokoohaki, and Mihhail Matskin

Part IV Sizing and Fit Recommendations

Learning Size and Fit from Fashion Images 111
Nour Karessli, Romain Guigourès, and Reza Shirvany

Part V Generative Outfit Recommendation

Generating High-Resolution Fashion Model Images Wearing Custom Outfits 135
Gökhan Yildirim, Nikolay Jetchev, Roland Vollgraf, and Urs Bergmann

Part I
Cold Start in Recommendations

Fashion Recommender Systems in Cold Start

Mehdi Elahi and Lianyong Qi

Abstract With the rapid growth of online market for clothing, footwear, hairstyle, and makeup, consumers are getting increasingly overwhelmed with the volume, velocity and variety of production. Fashion Recommender Systems can tackle choice overload by suggesting the most interesting products to the users. However, recommender systems are unable to generate recommendation unless some information is collected from users. Indeed, there are situations where a recommender system is requested for recommendation while no or little information is provided by users (*Cold Start* problem). In this book chapter, we investigate the different scenarios where fashion recommender systems may encounter cold start problem and review approaches that have been proposed to deal with this problem. We further elaborate potential solutions that can be applied to mitigate moderate and severe cases of cold start problem.

Keywords Fashion recommendation · Recommender systems · Cold start · New user · New item · Visual · Style · Cross domain · Active learning · Side information · Preference elicitation

1 Introduction

Fashion is defined as *"The cultural construction of the embodied identity"*.[1] Fashion is commonly described as the prevailing style of dress or behavior and it can be characterized by the notion of *change*. Fashion encompasses various forms of self-

[1]Editorial policy of Fashion Theory: The Journal of Dress, Body & Culture

M. Elahi (✉)
Department of Information Science and Media Studies, University of Bergen, Bergen, Norway
e-mail: mehdi.elahi@uib.no

L. Qi
Qufu Normal University, Shandong, China

N. Dokoohaki (ed.), *Fashion Recommender Systems*, Lecture Notes in Social Networks, https://doi.org/10.1007/978-3-030-55218-3_1

fashioning ranging form street styles to high fashion made by designers [94]. A major challenge in the fashion domain is the increasing *Variety, Volume*, and *Velocity* of fashion production which makes it difficult for the consumers to choose which product to purchase.[2] Shakespeare has -somehow- addressed this issue a long time ago by noting "that the fashion wears out more apparel than the man" [109]. This is not necessarily all negative as the more choices available the better the opportunity for consumers to choose appealing products. However, this phenomenon shall result in the problem of *choice overload*, i.e., the problem of having unlimited number of choices, especially when they do not differ significantly from each other [6, 9].

Recommender systems can mitigate this problem by suggesting a personalized selection of items (i.e., fashion products) that are predicted to be the most appealing for a target user (i.e., fashion consumer) [64, 101, 102, 111]. This is done by filtering irrelevant items and recommending a shortlist of the most relevant ones for the users. An effective filtering requires the system to (thoroughly) analyze the user preferences and (deeply) learn the particular taste and affinity of every individual user. A real world example could be analyzing the purchase history of a customer in Amazon[3] and predicting the interests of the user and ultimately generating recommendation for her. During this process, the recommender system can carefully observe the users' behaviors and elicit different forms of user preferences, in order to understand the personal needs and constrains of the users [97, 104, 114].

User preferences can be elicited in different forms, i.e., in the form of *explicit preference* or *implicit preference* [71, 113]. Explicit preference is a form of user assessment that is explicitly reported by a user (e.g., ratings for items in Zalando[4]) [44, 45]. In spite of its benefits, eliciting explicit preferences requires a certain level of user efforts [39] and may still lack to fully picture the true desires of a user [87]. Implicit preference is another form of preferences which is inferred from the actual observed activities (e.g., clicks on items in Zalando) [49, 59, 91]. Although traditional recommender systems focused on exploiting explicit preferences, however, modern recommendation platforms, particularly in e-commerce, shifted towards techniques that deal with implicit preferences.

Proven capability of recommender systems in learning different forms of user preferences and effectively dealing with choice overload has empowered them to turn to be an essential component of any modern e-commerce that needs to deal with a large catalog of items [15]. Fashion recommender systems have also shown to be effective in supporting users when making choices. Fashion recommendation deals with personalized selection and suggestion of a variety of products ranging from clothing to makeup, including recommendation of individual products or a set of products (outfits). Such a personalized recommendation is commonly generated based on the preferences of a network of consumers and computing the relationships

[2] https://www.zdnet.com/article/volume-velocity-and-variety-understanding-the-three-vs-of-big-data/

[3] https://www.Amazon.com

[4] https://www.zalando.no

and similarities among their preferences [18, 54, 88, 96, 120]. The effectiveness of fashion recommender systems has been proven in the cases where a decision support tool is needed to assist fashion customer during their interactions with an online shop. Such a support enhances the experiences of the users during the time of shopping, e.g., with surprising recommendation offered to them.

This book chapter addresses the cold start problem in fashion recommender systems. It describes different scenarios of cold start and reviews the potential solutions for this problem proposed so far. The rest of the book chapter is organized as following: Sect. 2 briefly describes the common recommendation techniques in fashion domain. Section 3 explains different scenarios of cold start problem and Sect. 4 reviews solutions for this problem, namely, solutions focused on item-based and user-based side-information (Sects. 4.1 and 4.2), solutions based on implicit preferences (Sect. 4.3), and potential solutions based on cross-domain and active learning (Sects. 4.4 and 4.5). Finally, Sect. 5 provides a conclusion for the book chapter.

2 Techniques for Fashion Recommendation

There have been a wide range of recommendation techniques that have been proposed in fashion domain [63, 76]. These techniques can be classified into the following categories:

- Collaborative Filtering (CF) [54, 58, 67] is a popular recommendation technique that aims at learning preferences of users from their ratings for fashion items and predicting the missing ratings that might be given to other products by the users. The user preferences are typically provided to the system in the form of a rating matrix where every entry represents a rating given by a user to an item (see Fig. 1). The system then recommends to a user those items that have the highest predicted ratings (i.e., missing entries in the rating matrix).
- Content-based (CB) [10, 18, 76, 121] class of recommendation techniques focuses on adopting the content of the fashion items and generating recommendation based on the content features (e.g., textual description or visual features).
- Hybrid [21, 65, 74] class of recommender systems takes advantage of a combination of techniques from multiple classes of recommender systems in order to deal with the limitations of the individual techniques.
- Machine Leaning class of recommendation techniques adopts a range of computational models with different mechanisms compared to the above-mentioned (classical) techniques. An example is [76] where a *latent* Support Vector Machines (SVM) has been developed for fashion recommendation. Another example is [62] where the authors adopted a probabilistic topic model for learning fashion coordinates that can be used for recommendation. In [63] the authors combined a deterministic and a stochastic machine learning models for recommendation. Another group of approaches implements Learn-to-Rank

Fig. 1 Schematic figure representing a rating matrix [4]

(L2R) algorithms for recommendation with three alternative variations, i.e., *pointwise, pairwise,* and *listwise* [78]. Pointwise variations predict the relevance score of each item independently. Pairwise variations aim at correctly ordering the pairs of items instead of individual items. Hence, the objective is to rank the more relevant items higher than less relevant ones. Listwise variations rely on ranked *lists* as training examples [45]. In [58], the authors proposed a technique based on tensor factorization in order to find the best clothing match. The authors of [55] proposed a technique that extends the Bayesian Personalized Ranking (BPR) by incorporating different item features. A more recent set of works developed recommender systems adopting different variations of the neural networks. An example is [56] where the authors employed Long Short-Term Memory (LSTM) cells [57] to learn temporal correlations between purchases in a fashion shop and to predict the preferred style of each user based on their past purchases.

Although the core recommendation technique plays an important role for the performance, still a recommender system will fail to generate relevant recommendation of fashion products without having certain quantity of quality preference data. This problem is known as *Cold Start* and it happens when the system has not yet acquired sufficient preference data (e.g., user ratings) to build meaningful recommendation for users (see Fig. 2). The most common cases of the cold start problem happen when the recommender system is unable to build recommendation for a new user, known as *New User* problem. Another problem happens when the recommender system is unable to recommend a new item to any user, known as *New Item* problem [4, 7, 35, 108]. In severe cases of cold start, both of the new user and new item problems may happen all together. This is a case when an online fashion shop is recently launched and the database does not contain considerable quantity and quality of data portraying the preferences of customers [7, 35, 114].

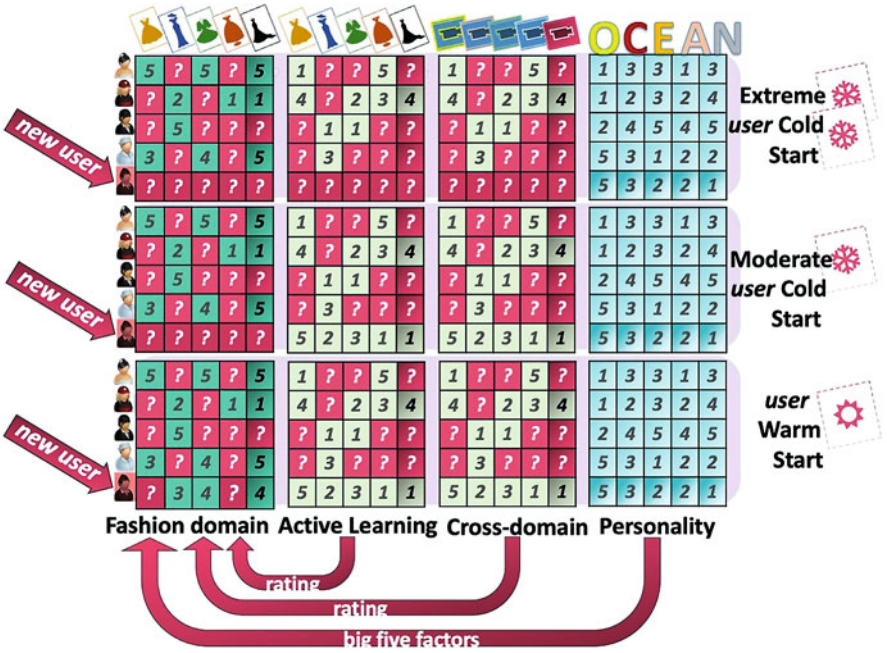


Fig. 2 User Cold Start scenarios: (i) Extreme User Cold Start, (ii) Moderate User Cold Start, and (iii) User Warm Start. Within each scenario, certain values of user-annotated features can be unknown to the recommender system. These missing features are indicated with "?" mark

3 Cold Start

Fashion recommender systems use a dataset of user preferences, typically in the form of ratings, that have been provided by a large community of customers to a catalog of fashion products. The dataset can be defined as a rating matrix where rows show the customers (users) and columns show the fashion products (see Fig. 1). Fashion recommender systems then use this dataset and compute prediction for the items that might be interesting to a target user [33, 70]. Recommender systems recognize patterns of relationships within the data and exploit them to build rating predictions that can ultimately be used for generating recommendation.

Predicted ratings are built for each unknown rating for a user-item pair within the defined rating matrix. This leads to computing a ranking list for fashion items, for a particular user. In the ranking list, the items are sorted accordingly to the predicted ratings for that user. Fashion recommender system short-lists the top items of the ranking list with the highest predicted ratings and presents them to a target user in the form of a recommendation list.

It is a fact that fashion recommender systems have already shown promising performance in dealing with particularities of this domain. However, they may still suffer from a number of challenges due to the lack of data for certain users or certain

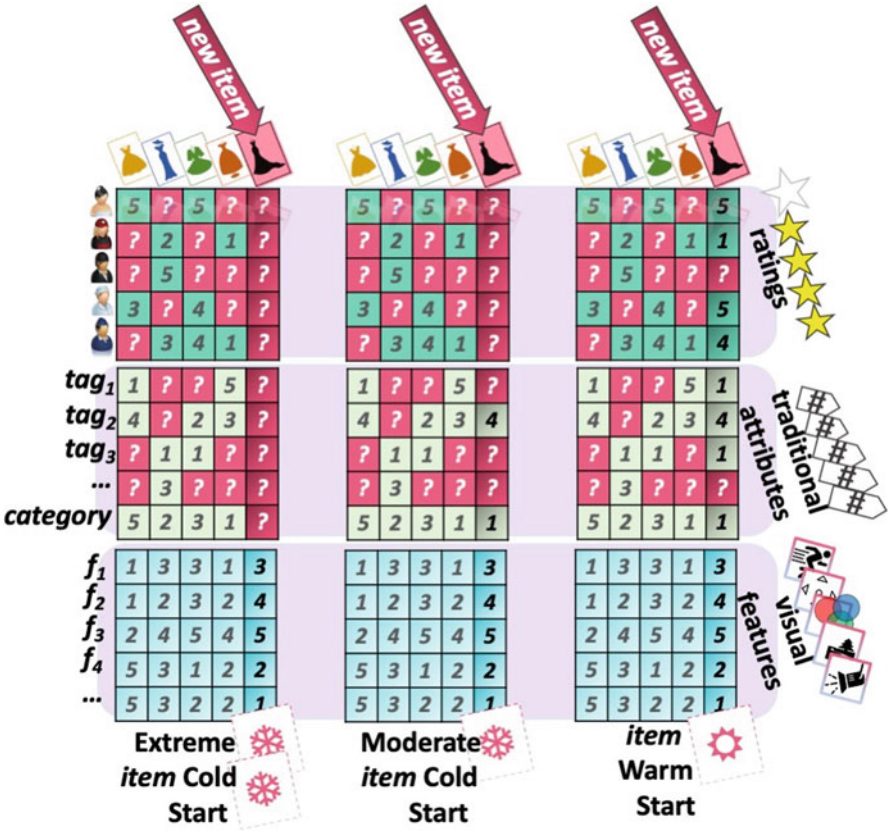


Fig. 3 Item Cold Start scenarios: (i) Extreme item Cold Start, (ii) Moderate item Cold Start, and (iii) item Warm Start. Within each scenario, certain values of user-annotated features can be unknown to the recommender system. These missing features are indicated with "?" mark

items [4, 108]. These challenges are mainly related to the cold start problem. One of the main challenges is defined as the *New User* problem which may happen when a new user enters to the fashion recommender system and requests recommendations before she has provided any preferences to any item (see Fig. 2). Another form of this challenge is defined as the *New Item* problem which happens when a new fashion product is added into the item catalog and none of the users has yet rated that new product (see Fig. 3). Similarly, the *Sparsity* of the rating dataset is known to be another related challenge. In extreme cases of sparsity problem, the performance of the fashion recommender systems will be damaged resulting in a very low quality of recommendation. In such a situation, the number of *known* ratings is extremely lower than the number of *unknown* ratings and still the fashion recommender system has to make predictions for the large number of unknown ratings [4, 11].

In real-world fashion recommender systems, different scenarios may happen, i.e., *Extreme* cold start, *Moderate* cold start, *Warm* start scenario.

- **Extreme Cold Start** in fashion recommender systems occurs when a new user registers to the system and requests recommendation before providing any data about her preferences (extreme new user problem). This scenario is illustrated in Fig. 2 (top row). This problem also occurs when a new product is added to the catalog without holding any data that can describe that item. Consequently, the system would fail to recommend that item to any user (extreme new item problem). This scenario is shown in Fig. 3 (left column). This is a serious problem and has to be tackled promptly.
- **Moderate Cold Start** occurs when a limited amount of preference data is provided by a user or a certain form of side information is collected to be used by the system for recommendation (Moderate New User Problem). This scenario is illustrated in Fig. 2 (middle row). It could also happen for a new item when some sort of semantic features are partially available (Moderate New Item Problem). This scenario is shown in Fig. 3 (middle column). Moderate cold start can happen as a mixed scenario of extreme cold start for some items and warm start for other items. Hence, it can be seen as an intermediate situation when a recommender system is in a transition phase from extreme cold start to warm start situation. However, this still means that the system has a serious problem as the user and items in the extreme cold start situation may not be well served by recommender system. This is the most common scenario and related literatures typically refer to this scenario as a cold start.
- **Warm Start** can be considered as the best possible scenario for fashion recommender systems as significant information provided by users (User Warm Start). This scenario is illustrated in Fig. 2 (bottom row). In the case of items, it refers to the situation when fashion products have already obtained considerable preference data that can be well exploited for recommendation (Item Warm Start). This scenario is presented in Fig. 3 (right column). There could be also considerable quantity of user-annotated semantic features (i.e., tags and reviews) in the dataset.

The rest of the book chapter, reviews the potential solutions to tackle the cold start problem.

4 Potential Solutions

4.1 Item Side Information Approaches

Content-based Filtering (CBF) has been one of the most popular approaches for recommendation in different application domains [47, 63, 64, 76, 121]. CBF relies on content features of items (as known as *side information*) in order to effectively

mitigate the cold start problem in recommender systems [13, 29, 30, 36]. In fashion recommender systems, when a new product is added to the item catalog, an initial profile of the item is made by using different sources of content features (see Fig. 3) [52, 65]. These content features are exploited by the system to form a *Vector Space Model (VSM)* [93], where items are represented by a multi-dimensional vector [27, 31, 80]. The system adopts machine learning models that can learn from item vectors and recognize patterns among them and ultimately generate relevant recommendation [84, 103].

Traditionally, content features exploited by recommender systems were *semantic* features based on semantic content (e.g., item description, tags, and category) [7, 27, 32, 80, 106]. However, recent approaches for fashion recommendation implement the novel idea of further enriching the item description with visual features [28]. Such visual features encapsulate the aesthetic form of the fashion products and represent the *style*. Such visual features are typically extracted from the product images based on the methodologies brought from Computer Vision and multimedia retrieval and recommendation [53, 72, 103, 105, 115].

A variety of research works have been performed on investigating usage of visual features for user and item modeling in fashion domain, e.g., for clothing matching [58, 83] and visually-aware recommendation [54, 55, 62]. Different forms of visual features have been proposed for extraction that can be classified into two big classes, i.e., (i) Hand-crafted features and (ii) Deep Learning (DL) based features [54, 55, 62, 63, 65, 77, 124].

While hand-crafted features [65] may still offer promising performance, recently, deep learning based approaches have achieved superior accuracy in comparison to them [19]. Adopting Convolutional Neural Networks (CNN) is an example of approaches based on deep learning that builds discriminative representation of fashion products [79]. Another work that pioneered this research is [76] where the authors proposed a clothing recommender system based on human body parsing detectors and latent Support Vector Machine (SVM) model. The authors in [58] proposed a technique called Functional Pairwise Interaction Tensor Factorization (FPITF) that is capable of using tensor factorization in order to predict the clothing match.

Regardless of the type of features, several works developed methodologies to exploit the content features when dealing with cold start problem. In [26], the authors proposed a content-based recommender system that constructs detailed clothing features to build up item profiles. The recommender algorithm is based on K-Nearest Neighbors (KNN) which is commonly used to make similarity-based recommendation for new items. Authors of [52] extended Collaborative Filtering and enabled it to recommend groups of fashion products (instead of individual products). The groups are formed based on certain type of features such as product category. This allowed their recommender system to tackle the new item problem. The approach has been deployed in a fashion retailer called *Rue La La*. In [56] the authors employed Long Short-Term Memory (LSTM) cells [57] that can handle cold start while learning temporal correlations between purchases in a fashion shop and ultimately to generate recommendation based on their past purchases.

In addition, there exists other works that go beyond recommendation and extend usage of visual features by performing recognition, parsing, and style extraction of clothing [65]. By common definition, clothing recognition focuses on matching clothing with clothing in an online shop and retrieving similar clothing from their photos. Clothing parsing methods aim at decomposing and labeling semantically meaningful parts of the clothing photo. Style extraction methods aim to learn the style of a product by extracting descriptive features from its visual content [65]. Among these tasks, the extracted style of clothing can be the most relevant for the fashion recommendation. For instance, the clothing style will enable the recommender systems to tackle the new item problem. Examples of approaches within this group of recommender systems are [20, 68]. Even collaborative filtering based techniques can be extended to be capable of using visual features. As an example, authors [55] proposed Visual Bayesian Personalized Ranking (VBPR) that extends the Bayesian Personalized Ranking (BPR) by incorporating the visual features. VBPR can be further extended to model the evolution of fashion trends in visually-aware recommendation [54].

4.2 User Side Information Approaches

A potential approach to deal with the cold start problem focuses on exploiting additional user attributes (as known as *side information*). A number of works have adopted this approach to build a customer profile not only in fashion domain [17, 85] but also other domains [12, 82, 86, 86]. Different forms of user attributes has been proposed in the recommender system literature. One of the important forms of side information represents the psychological characteristics of user and can be modeled by *Personality Traits*. The personality traits are defined as predictable and stable characteristics of individuals which can explain the "consistent behavior pattern and interpersonal processes originating within the individuals" [14]. Personality traits can represent the differences of individuals with respect to *emotional, interpersonal, experiential, attitudinal* and *motivational* dimensions [66].

A number of previous researches have investigated the impact of clothing attribute on the impression of personality. The results have shown that clothing (as part of fashion) can communicate comprehensive pieces of information about differences among people [122]. For instance, clothing can represent the favor, society level, attitude toward life, and personality. In fact, clothing is a language of signs, a nonverbal system of communication [81]. The authors of [25] modified the clothing colour for job applicants and changed it from light to dark. The results have shown certain level of differences in judgments of applicants.

A range of psychological models have been proposed to model the personality traits of an individual. A popular model is the *Big Five Factor model (FFM)* [22] which explains the personality of an individual in terms of five dimensions called big five traits. The list of these traits is: *Openness, Conscientiousness, Extroversion,*

Agreeableness and *Neuroticism* (as known as *OCEAN*). Figure 2 (right column) illustrates how the personality traits can be represented in a user profile.

A number of reviewed works have shown that individuals with dissimilar personality traits follow dissimilar choice making processes [41, 90]. Hence, individuals with similar personality traits are more likely to share similar choices and preferences [100, 107]. Prior works have also developed the idea of using personality traits in recommender systems to mitigate the cold start problem [11, 107]. When a new user registers in a recommender system and has not given any information about herself, personality traits can be an alternative source of information in order to build relevant recommendation (see Fig. 2). This can be done to compute the user-based similarity using personality traits or building computational (latent) models based on personality traits [41]. As an example of works within this area, the authors of [117] adopted different recommendation approaches and showed that incorporation of personality may lead to a better recommendation quality in cold-start scenario. [89] has investigated the potential of using personality and showed that personality characteristics can lead to improvement in the performance of recommender systems.

In fashion domain, limited attempts have been done on learning from effective signals obtained from people (i.e., short-term emotions as well as long-term moods and personality traits) for the potential of making recommendation. For instance, [95] integrated emotion signal of consumers for building recommendation of fashion products in the cold start situation. The work aimed at investigating the potential power of peoples' affective characteristics in predicting their preferences for fashion products. In [40] the authors proposed a recommender system that can use different forms of information, including user persona (personality) and social connections to recommend clothing. The authors of [122] has made analysis on exploring the potential relationship of personality type and how people make choices of wearing.

4.3 Approaches Based on Implicit Preferences

An alternative type of cold start solutions focuses on collecting the implicit preferences (feedback) from the users and utilizing them for generating recommendation. When a customer enters to an online shop, the system can monitor her activities and try to infer her preferences from the activities. As an example, the system can log the browsing history or purchase records and learn the actual preferences of the user which are then exploited for recommendation. In modern recommender systems, even facial expressions of the users can be used to identify emotions and ultimately the preferences and choices [1, 118, 119].

A wide range of recommender systems are capable of leveraging implicit preferences, some of which are specifically designed for that goal [50, 59, 98, 99] and the others adopt hybrid models that use both of the implicit and explicit preferences. As an example, the authors of [69] extend the standard Singular Value

Decomposition (SVD) model which originally leverages only explicit preferences. The extended model is called SVD++ and it can exploit implicit preferences in order to generate recommendation.

In fashion domain, a number of works have proposed a recommendation framework based on the implicit preferences. For example, in [55], a recommendation model has been developed that can tackle the cold start problem by incorporating both implicit preferences from users together with visual features from items. In [88], the authors assumed that no explicit preferences are available and hence developed a fashion recommendation framework that can learn from implicit preferences of users collected by a fashion app. The collected data includes user actions ranging from scrolling in the app to purchasing a product. In addition to user related data, the system also uses item price and popularity in order to generate recommendation. Also visual Bayesian personalized ranking [54], introduced before, is capable of incorporating the implicit preferences for improving the performance of fashion recommendation in cold start.

4.4 Cross-Domain Approaches

Alternative class of approaches for fashion recommender systems focuses on using cross-domain methodology [65]. This recommendation class is also referred to as *transfer learning* where a (data-driven) function is learnt to link the target domain (fashion) and auxiliary domains. This can be done by projecting the representations of the target and auxiliary domains into a common feature space.

Cross-domain recommendation has been well-studied not only in the field of recommender systems, but also in related research areas for a situation where only a limited quantity of data might be available in the target domain [61]. The reason can be due to the fact that current e-commerce web applications typically operate in multiple domains and they use mechanisms to aggregate multiple types of data from multiple domains. Availability of such data can bring benefits to a recommender system and enable it to perform *cross-selling* or coping with the cold start problem in its target domain.

There have been various algorithms developed for cross-domain recommendation [42, 92, 123]. While these algorithms may implement different mechanisms for the cross-domain recommendation, they share commonalities which enable us to classify them into two major classes, i.e., *Knowledge Aggregation* approaches [2, 8, 16, 110] and *Knowledge Transfer* approaches [24, 46, 73, 116].

The former approach aims to *aggregate* the knowledge from different auxiliary domains in order to generate recommendations in the target domain. The latter approach is based on the idea of eliciting and transferring the user ratings from auxiliary domains and *transfer* this knowledge to the target domain. In this sense, the latter approach attempts to *link* different domain knowledge in order to support the recommendation for the target domain [24, 35].

A representative example is the work of [51] where authors proposed a deep learning technique to compute the similarity among the photos taken from streets and shops. The technique is based on Convolutional Neural Networks (CNN). Transfer learning has been also adopted to mitigate the complexity of training deep learning techniques in the fashion domain. As an example, [75] proposed adopting GoogleNet architecture [115] for their training set for that task of feature extraction from clothing [65].

4.5 Rating Elicitation Approaches

An alternative set of approaches that can remedy the cold start problem are based on the idea of rating elicitation in recommender systems. These approaches are called *Active Learning*, a notion that traditionally has origin in theory of machine learning [39, 43]. This set of approaches has been adopted for designing algorithms in solving problems with scarce resources [3, 5, 112]. Active learning can be applied when a machine learning algorithm needs a large dataset for training [48] while such data is limited or expensive to acquire.

Active learning can offer a number of advantages specially in the initial phase of interaction of new users with recommender systems. It can be used to request the new users to reveal their preferences by rating a set of items [34, 37, 38], enabling the recommender system to bootstrap its knowledge about the taste and affinity of the new user (see Fig. 2). Active learner adopts a number of heuristics or rules to coordinate the process of rating elicitation. Those heuristics (as known as *strategies*) will allow the system to concentrate on eliciting the ratings that are more informative for the system to learn the user profile [97, 104]. This can be done in a single domain or multi-domain scenario [92].

In fashion domain, a minor attention has been drawn by active learning strategies. As an example, [60] presents an active learning strategy as part of clothing image retrieval and recommendation framework. The strategy is enabled to learn the preferences of users and use them for generating personalized recommendation. The proposed framework can also utilize a content-based algorithm and employ a user interaction to elicit the user feedback. The evaluation has shown the effectiveness of the developed strategy.

5 Conclusion

This book chapter discusses several scenarios related to the cold start problem in fashion recommender systems. The challenges that may happen in each of these scenarios can largely damage the performance of fashion recommender systems regardless of the quality of the core algorithm. The chapter reviews potential solutions that can be utilized to remedy these challenges.

The solutions can be classified into a number of categories, namely approaches based on *item side information* and *user side information*, approaches based on *implicit preferences*, *cross-domain* approaches, and *active learning*. Most of these approaches have been well-integrated in fashion domain while some may still have potential to be used due to their promising results in the other related domains. It is important to note that none of the approaches can necessarily offer the ultimate and conclusive solution to all of the above-mentioned cold start scenarios. In fact, every approach has a set of advantages and disadvantages which make it a unique solution that may better suit to a specific cold start scenario.

Moreover, many of the surveyed literatures have mainly viewed the fashion recommender systems from a narrow lens of classical rating-based systems with the matrix representation of users and items. Hence, the subject of the recommendation is reduced to mainly suggesting outfit to a potential fashion shopper. This is while the task of fashion recommendation may go beyond that traditional view of only finding the right outfit for a shopper and become more of modelling fashion products along with dimensions of style, design, size and fit.

Finally, it shall be noted that, additional to an efficient solution that can deal with cold start problem, a recommender system requires effective usage of interface and interaction design [23] to well serve its users and to fulfill their needs and constrains. This makes the research on cold start to be a cross-disciplinary area where various disciplines are involved, e.g., Interaction Design, Data Science, Databases and Psychology. This chapter may hopefully open up and offer a bird eye view of the cold start in fashion domain which shall be beneficial for researchers in the academia and practitioners in the industry, and as a result, advancing the knowledge in the recommender systems area.

References

1. Amazon go and the future of sentient buildings: An analysis
2. Abel F, Herder E, Geert-Jan Houben, Henze N, Krause D (2013) Cross-system user modeling and personalization on the social web. User Model User-Adap Inter 23(2–3):169–209
3. Abu-Mostafa YS, Magdon-Ismail M, Lin H-T (2012) Learning From Data. AMLBook. New York, NY, USA
4. Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng 17(6):734–749
5. Alpaydin E (2010) Introduction to machine learning, 2nd edn. The MIT Press
6. Anderson C (2006) The long tail. Random House Business. London
7. Bakhshandegan Moghaddam F, Elahi M (2019) Cold start solutions for recommendation systems. IET. England
8. Berkovsky S, Kuflik T, Ricci F (2008) Mediation of user models for enhanced personalization in recommender systems. User Model User-Adap Inter 18(3):245–286
9. Bollen D, Knijnenburg BP, Willemsen MC, Graus M (2010) Understanding choice overload in recommender systems. In: Proceedings of the Fourth ACM Conference on Recommender Systems. ACM, pp 63–70

10. Bracher C, Heinz S, Vollgraf R (2016) Fashion DNA: merging content and sales data for recommendation and article mapping. arXiv preprint arXiv:1609.02489
11. Braunhofer M, Elahi M, Ricci F (2014) Techniques for cold-starting context-aware mobile recommender systems for tourism. Intel Artif 8(2):129–143
12. Braunhofer M, Elahi M, Ricci F (2015) User personality and the new user problem in a context-aware point of interest recommender system. In: Information and Communication Technologies in Tourism 2015. Springer, pp 537–549.
13. Braunhofer M, Elahi M, Ricci F, Schievenin T (2013) Context-aware points of interest suggestion with dynamic weather data management. In: Information and communication technologies in tourism 2014. Springer, pp 87–100
14. Burger JM (2010) Personality. Wadsworth Publishing, Belmont
15. Burke R (2002) Hybrid recommender systems: Survey and experiments. User Model User-Adap Inter 12(4):331–370
16. Cantador I, Tobías IF, Berkovsky S, Cremonesi P (2015) Cross-domain recommender systems. In: Recommender systems handbook, 2nd edn. Springer, Boston, MA, USA. pp 919–959
17. Casidy R (2012) An empirical investigation of the relationship between personality traits, prestige sensitivity, and fashion consciousness of generation y in Australia. Australas Mark J 20(4):242–249
18. Chao X, Huiskes MJ, Gritti T, Ciuhu C (2009) A framework for robust feature selection for real-time fashion style recommendation. In: Proceedings of the 1st International Workshop on Interactive Multimedia for Consumer Electronics, pp 35–42
19. Chen Q, Huang J, Feris R, Brown LM, Dong J, Yan S (2015) Deep domain adaptation for describing people based on fine-grained clothing attributes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 5315–5324
20. Chen X, Chen H, Xu H, Zhang Y, Cao Y, Qin Z, Zha H (2019) Personalized fashion recommendation with visual explanations based on multimodal attention network: towards visually explainable recommendation. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 765–774
21. Claypool M, Gokhale A, Miranda T, Murnikov P, Netes D, Sartin M (1999) Combining content-based and collaborative filters in an online newspaper. In: Proceedings of the ACM SIGIR '99 Workshop on Recommender Systems: Algorithms and Evaluation, Berkeley. ACM
22. Costa PT, McCrae RR (1992) Revised NEO Personality Inventory (NEO PI-R) and NEO Five-Factor Inventory (NEO FFI): Professional Manual. Psychological Assessment Resources
23. Cremonesi P, Elahi M, Garzotto F (2015) Interaction design patterns in recommender systems. In: Proceedings of the 11th Biannual Conference on Italian SIGCHI Chapter. ACM, pp 66–73
24. Cremonesi P, Tripodi A, Turrin R (2011) Cross-domain recommender systems. In: Proceedings of the 11th International Conference on Data Mining Workshops, pp 496–503
25. Damhorst ML (1990) In search of a common thread: Classification of information communicated through dress. Cloth Text Res J 8(2):1–12
26. de Melo EV, Nogueira EA, Guliato D (2015) Content-based filtering enhanced by human visual attention applied to clothing recommendation. In: 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI). IEEE, pp 644–651
27. Degemmis M, Lops P, Semeraro G (2007) A content-collaborative recommender that exploits wordnet-based user profiles for neighborhood formation. User Model User-Adap Inter 17(3):217–255
28. Deldjoo Y, Elahi M, Cremonesi P (2016) Using visual features and latent factors for movie recommendation. CEUR-WS
29. Deldjoo Y, Quadrana M, Elahi M, Cremonesi P (2017) Using mise-en-scene visual features based on MPEG-7 and deep learning for movie recommendation. CoRR abs/1704.06109. http://arxiv.org/abs/1704.06109
30. Deldjoo Y, Elahi M, Quadrana M, Cremonesi P (2018) Using visual features based on MPEG-7 and deep learning for movie recommendation. Int J Multimed Inf Retr 7(4):207–219

31. Deldjoo Y, Quadrana M, Elahi M, Cremonesi P (2017) Using mise-en-sc\ene visual features based on mpeg-7 and deep learning for movie recommendation. arXiv preprint arXiv:1704.06109
32. Deldjoo Y, Schedl M, Elahi M (2019) Movie genome recommender: a novel recommender system based on multimedia content. In: 2019 International Conference on Content-Based Multimedia Indexing (CBMI). IEEE, pp 1–4
33. Desrosiers C, Karypis G (2011) A comprehensive survey of neighborhood-based recommendation methods. In Ricci F, Rokach L, Shapira B, Kantor PB (eds) Recommender systems handbook. Springer, Boston, MA, USA. pp 107–144
34. Elahi M (2014) Empirical evaluation of active learning strategies in collaborative filtering. Ph.D. thesis, Ph.D. Dissertation. Free University of Bozen-Bolzano
35. Elahi M, Braunhofer M, Gurbanov T, Ricci F (2019) User preference elicitation, rating sparsity and cold start. In: Collaborative recommendations: algorithms, practical challenges and applications. World Scientific, Singapore
36. Elahi M, Deldjoo Y, Bakhshandegan Moghaddam F, Cella L, Cereda S, Cremonesi P (2017) Exploring the semantic gap for movie recommendations. In: Proceedings of the Eleventh ACM Conference on Recommender Systems. ACM, pp 326–330
37. Elahi M, Ricci F, Repsys V (2011) System-wide effectiveness of active learning in collaborative filtering. In: Proceedings of the International Workshop on Social Web Mining, Co-located with IJCAI, Barcelona, July 2011
38. Elahi M, Ricci F, Rubens N (2014) Active learning in collaborative filtering recommender systems. In: E-commerce and web technologies. Springer, pp 113–124
39. Elahi M, Ricci F, Rubens N (2014) Active learning strategies for rating elicitation in collaborative filtering: a system-wide perspective. ACM Trans Intell Syst Technol 5(1):13:1–13:33
40. Etebari D (2014) Intelligent wardrobe: using mobile devices, recommender systems and social networks to advise on clothing choice. Ph.D. thesis, University of Birmingham
41. Fernández-Tobías I, Braunhofer M, Elahi M, Ricci F, Cantador I (2016) Alleviating the new user problem in collaborative filtering by exploiting personality information. User Model User-Adap Inter 26(2–3):221–255
42. Ignacio Fernández-Tobías, Iván Cantador, Kaminskas M, Ricci F (2012) Cross-domain recommender systems: a survey of the state of the art. In: Proceedings of the 2nd Spanish Conference on Information Retrieval, pp 187–198
43. Flach P (2012) Machine learning: the art and science of algorithms that make sense of data. Cambridge University Press, New York
44. Freno A (2017) Practical lessons from developing a large-scale recommender system at Zalando. In: Proceedings of the Eleventh ACM Conference on Recommender Systems, pp 251–259
45. Freno A (2019) Clothing recommendations: the Zalando case. In: Collaborative recommendations: algorithms, practical challenges and applications. World Scientific Publishing Company, Singapore. pp 687–711
46. Gao S, Luo H, Chen D, Li S, Gallinari P, Guo J (2013) Cross-domain recommendation via cluster-level latent factor model. In: Proceedings of the 2013 European Conference on Machine Learning and Knowledge Discovery in Databases, pp 161–176
47. Ge M, Elahi M, Fernaández-Tobías I, Ricci F, Massimo D (2015) Using tags and latent factors in a food recommender system. In: Proceedings of the 5th International Conference on Digital Health 2015, pp 105–112
48. Ge M, Helfert M (2007) A review of information quality research – develop a research agenda. In: ICIQ, pp 76–91
49. Gurbanov T, Ricci F (2017) Action prediction models for recommender systems based on collaborative filtering and sequence mining hybridization. In: Proceedings of the Symposium on Applied Computing, SAC '17, New York. ACM, pp 1655–1661

50. Gurbanov T, Ricci F, Ploner M (2016) Modeling and predicting user actions in recommender systems. In: Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization, UMAP '16, New York. ACM, pp 151–155
51. Hadi Kiapour M, Han X, Lazebnik S, Berg AC, Berg TL (2015) Where to buy it: Matching street clothing photos in online shops. In: Proceedings of the IEEE international conference on computer vision, pp 3343–3351
52. Harrison S, Wilson B, Case study: building a hybridized collaborative filtering recommendation engine.
53. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 770–778
54. He R, McAuley J (2016) Ups and downs: modeling the visual evolution of fashion trends with one-class collaborative filtering. In: Proceedings of the 25th International Conference on World Wide Web, pp 507–517
55. He R, Julian McAuley (2016) VBPR: visual bayesian personalized ranking from implicit feedback. In: Thirtieth AAAI Conference on Artificial Intelligence
56. Heinz S, Bracher C, Vollgraf R (2017) An LSTM-based dynamic customer model for fashion recommendation. arXiv preprint arXiv:1708.07347
57. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
58. Hu Y, Yi X, Larry S Davis (2015) Collaborative fashion recommendation: a functional tensor factorization approach. In: Proceedings of the 23rd ACM International Conference on Multimedia, pp 129–138
59. Hu Y, Koren Y, Volinsky C (2008) Collaborative filtering for implicit feedback datasets. In: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, ICDM '08, Washington, DC. IEEE Computer Society, pp 263–272
60. Huang C-M, Wei C-P, Frank Wang Y-C (2013) Active learning based clothing image recommendation with implicit user preferences. In: 2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), pp 1–4. IEEE
61. Huang J, Feris RS, Chen Q, Yan S (2015) Cross-domain image retrieval with a dual attribute-aware ranking network. In: Proceedings of the IEEE International Conference on Computer Vision, pp 1062–1070
62. Iwata T, Watanabe S, Sawada H (2011) Fashion coordinates recommender system using photographs from fashion magazines. In: Twenty-Second International Joint Conference on Artificial Intelligence
63. Jagadeesh V, Piramuthu R, Bhardwaj A, Di W, Sundaresan N (2014) Large scale visual recommendations from street fashion images. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data mining, pp 1925–1934
64. Jannach D, Zanker M, Felfernig A, Friedrich G (2010) Recommender systems: an introduction. Cambridge University Press
65. Jaradat S (2017) Deep cross-domain fashion recommendation. In: Proceedings of the Eleventh ACM Conference on Recommender Systems, pp 407–410
66. John OP, Srivastava S (1999) The big five trait taxonomy: history, measurement, and theoretical perspectives. In: Handbook of personality: theory and research, vol 2, pp 102–138
67. Kang H, Yoo SJ (2007) SVM and collaborative filtering-based prediction of user preference for digital fashion recommendation systems. IEICE Trans Inf Syst 90(12):2100–2103
68. Kang W-C, Fang C, Wang Z, McAuley J (2017) Visually-aware fashion recommendation and design with generative image models. In: 2017 IEEE International Conference on Data Mining (ICDM). IEEE, pp 207–216
69. Koren Y (2008) Factorization meets the neighborhood: A multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08, New York. ACM, pp 426–434

70. Koren Y, Bell R (2011) Advances in collaborative filtering. In Ricci F, Rokach L, Shapira B, Kantor P (eds) Recommender systems handbook. Springer, Boston, MA, USA. p 145–186
71. Koren Y, Bell R (2015) Advances in collaborative filtering. In: Recommender Systems Handbook. Springer, pp 77–118
72. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
73. Li B, Yang Q, Xue X (2009) Can movies and books collaborate? Cross-domain collaborative filtering for sparsity reduction. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence, pp 2052–2057
74. Li Q, Kim BM (2003) An approach for combining content-based and collaborative filters. In: Proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages, AsianIR '03, vol 11, Stroudsburg. Association for Computational Linguistics, pp 17–24
75. Li Z, Sun Y, Wang F, Liu Q (2015) Convolutional neural networks for clothes categories. In: CCF Chinese conference on computer vision. Springer, pp 120–129
76. Liu S, Feng J, Song Z, Zhang T, Lu H, Xu C, Yan S (2012) Hi, magic closet, tell me what to wear! In: Proceedings of the 20th ACM International Conference on Multimedia, pp 619–628
77. Liu S, Song Z, Liu G, Xu C, Lu H, Yan S (2012) Street-to-shop: cross-scenario clothing retrieval via parts alignment and auxiliary set. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp 3330–3337
78. Liu T-Y et al (2009) Learning to rank for information retrieval. Found Trends Inf Retr 3(3):225–331
79. Liu Z, Luo P, Qiu S, Wang X, Tang X (2016) Deepfashion: powering robust clothes recognition and retrieval with rich annotations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1096–1104
80. Lops P, De Gemmis M, Semeraro G (2011) Content-based recommender systems: state of the art and trends. In: Recommender systems handbook, pp 73–105. Springer, Boston, MA, USA
81. Malcolm B (2002) Fashion as communication
82. Massimo D, Elahi M, Ricci F (2017) Learning user preferences by observing user-items interactions in an iot augmented space. In: Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization, pp 35–40
83. McAuley J, Targett C, Shi Q, Van Den Hengel A (2015) Image-based recommendations on styles and substitutes. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 43–52
84. Moghaddam FB, Elahi M, Hosseini R, Trattner C, Tkalčič M (2019) Predicting movie popularity and ratings with visual features. In: 2019 14th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP). IEEE, pp 1–6
85. Mulyanegara RC, Tsarenko Y, Anderson A (2009) The big five and brand personality: investigating the impact of consumer personality on preferences towards particular brand personality. J Brand Manag 16(4):234–247
86. Nasery M, Elahi M, Cremonesi P (2015) Polimovie: a feature-based dataset for recommender systems. In: ACM RecSys Workshop on Crowdsourcing and Human Computation for Recommender Systems (CrawdRec), vol 3, pp 25–30
87. Neidhardt J, Schuster R, Seyfang L, Werthner H (2014) Eliciting the users' unknown preferences. In: Proceedings of the 8th ACM Conference on Recommender systems. ACM, pp 309–312
88. Nguyen HT, Almenningen T, Havig M, Schistad H, Kofod-Petersen A, Langseth H, Ramampiaro H (2014) Learning to rank for personalised fashion recommender systems via implicit feedback. In: Mining Intelligence and Knowledge Exploration, pp 51–61. Springer
89. Nunes MASN (2009) Recommender systems based on personality traits: could human psychological aspects influence the computer decision-making process? VDM Verlag
90. Nunes MASN, Hu R (2012) Personality-based recommender systems: an overview. In: Proceedings of the 6th ACM Conference on Recommender Systems, pp 5–6

91. Oard DW, Kim J et al (1998) Implicit feedback for recommender systems. In: Proceedings of the AAAI Workshop on Recommender Systems, pp 81–83
92. Pagano R, Quadrana M, Elahi M, Cremonesi P (2017) Toward active learning in cross-domain recommender systems. arXiv preprint arXiv:1701.02021
93. Pazzani MJ, Billsus D (2007) The adaptive web. chapter Content-based Recommendation Systems. Springer, Berlin/Heidelberg, pp 325–341
94. Person. Definition of fashion.
95. Piazza A, Kröckel P, Bodendorf F (2017) Emotions and fashion recommendations: evaluating the predictive power of affective information for the prediction of fashion product preferences in cold-start scenarios. In: Proceedings of the International Conference on Web Intelligence, pp 1234–1240
96. Quanping H (2015) Analysis of collaborative filtering algorithm fused with fashion attributes. Int J u- e-Serv Sci Technol 8(10):159–168
97. Rashid AM, Albert I, Cosley D, Lam SK, Mcnee SM, Konstan JA, Riedl J (2002) Getting to know you: learning new user preferences in recommender systems. In: Proceedings of the 2002 International Conference on Intelligent User Interfaces, IUI 2002. ACM Press, pp 127–134
98. Rendle S, Freudenthaler C, Gantner Z, Lars Schmidt-Thieme (2009) BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pp 452–461. AUAI Press
99. Rendle S, Freudenthaler C, Gantner Z, Lars Schmidt-Thieme (2012) BPR: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618
100. Rentfrow PJ, Gosling SD et al (2003) The do re mi's of everyday life: the structure and personality correlates of music preferences. J Pers Soc Psychol 84(6):1236–1256
101. Resnick P, Varian HR (1997) Recommender systems. Commun ACM 40(3):56–58
102. Ricci F, Rokach L, Shapira B, Kantor PB (2011) Recommender systems handbook. Springer, Boston, MA, USA
103. Rimaz MH, Elahi M, Bakhshandegan Moghadam F, Trattner C, Hosseini R, Tkalcic M (2019) Exploring the power of visual features for the recommendation of movies. In: Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization, pp 303–308
104. Rubens N, Elahi M, Sugiyama M, Kaplan D (2015) Active learning in recommender systems. In: Recommender systems handbook – chapter 24: Recommending active learning. Springer Boston, MA, USA. pp 809–846
105. Savian S, Elahi M, Tillo T (2020) Optical flow estimation with deep learning, a survey on recent advances. In: Deep biometrics. Springer, Boston, MA, USA. pp 257–287
106. Schedl M, Zamani H, Chen C-W, Deldjoo Y, Elahi M (2017) Current challenges and visions in music recommender systems research. arXiv preprint arXiv:1710.03208
107. Schedl M, Zamani H, Chen C-W, Deldjoo Y, Elahi M (2018) Current challenges and visions in music recommender systems research. Int J Multimedia Inf Retr 7(2):95–116
108. Schein AI, Popescul A, Ungar LH, Pennock DM (2002) Methods and metrics for cold-start recommendations. In: SIGIR '02: Proceedings of the 25th Annual international ACM SIGIR Conference on Research and Development in Information Retrieval, New York. ACM, pp 253–260
109. Shakespeare W (1997) Much ado about nothing, vol 2. Cambridge University Press, England
110. Shapira B, Rokach L, Freilikhman S (2013) Facebook single and cross domain data for recommendation systems. User Model User-Adap Inter 23(2–3):211–247
111. Shardanand U, Maes P (1995) Social information filtering: Algorithms for automating "word of mouth". In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '95, New Yor. ACM Press/Addison-Wesley Publishing Co., pp 210–217
112. Sipser M (1996) Introduction to the theory of computation, 1st edn. International Thomson Publishing
113. Stern DH, Herbrich R, Graepel T (2009) Matchbox: Large scale online bayesian recommendations. In: Proceedings of the 18th International Conference on World Wide Web, WWW '09, New York. ACM, pp 111–120

114. Su X, Khoshgoftaar TM (2009) A survey of collaborative filtering techniques. Adv Artif Intell 2009:4:2–4:2.
115. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1–9
116. Tiroshi A, Berkovsky S, Kâafar MA, Chen T, Kuflik T (2013) Cross social networks interests predictions based on graph features. In: Proceedings of the 7th ACM Conference on Recommender Systems, pp 319–322
117. Tkalcic M, Kunaver M, Košir A, Tasic J (2011) Addressing the new user problem with a personality based user similarity measure. In: Proceedings of the 1st International Workshop on Decision Making and Recommendation Acceptance Issues in Recommender Systems, p 106
118. Tkalčič M, Maleki N, Pesek M, Elahi M, Ricci F, Marolt M (2017) A research tool for user preferences elicitation with facial expressions. In: Proceedings of the Eleventh ACM Conference on Recommender Systems. ACM, pp 353–354
119. Tkalčič M, Maleki N, Pesek M, Elahi M, Ricci F, Marolt M (2019) Prediction of music pairwise preferences from facial expressions. In: Proceedings of the 24th International Conference on Intelligent User Interfaces, pp 150–159
120. Tu Q, Dong L (2010) An intelligent personalized fashion recommendation system. In: 2010 International Conference on Communications, Circuits and Systems (ICCCAS), pp 479–485. IEEE
121. Tuinhof H, Pirker C, Haltmeier M (2018) Image-based fashion product recommendation with deep learning. In: International Conference on Machine Learning, Optimization, and Data Science, pp 472–481. Springer
122. Wei Z, Yan Y, Huang L, Nie J (2017) Inferring intrinsic correlation between clothing style and wearers' personality. Multimedia Tools Appl 76(19):20273–20285
123. Winoto P, Tang TY (2008) If you like the devil wears prada the book, will you also enjoy the devil wears prada the movie? A study of cross-domain recommendations. N Gener Comput 26(3):209–225
124. Yamaguchi K, Kiapour MH, Ortiz LE, Berg TL (2012) Parsing clothing in fashion photographs. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp 3570–3577

Part II
Complementary and Session Based Recommendation

Enabling Hyper-Personalisation: Automated Ad Creative Generation and Ranking for Fashion e-Commerce

Sreekanth Vempati, Korah T. Malayil, V. Sruthi, and R. Sandeep

Abstract Homepage is the first touch point in the customer's journey and is one of the prominent channels of revenue for many e-commerce companies. A user's attention is mostly captured by homepage banner images (also called Ads/Creatives). The set of banners shown and their design, influence the customer's interest and plays a key role in optimizing the click through rates of the banners. Presently, massive and repetitive effort is put in, to manually create aesthetically pleasing banner images. Due to the large amount of time and effort involved in this process, only a small set of banners are made live at any point. This reduces the number of banners created as well as the degree of personalization that can be achieved. This paper thus presents a method to generate creatives automatically on a large scale in a short duration. The availability of diverse banners generated helps in improving personalization as they can cater to the taste of larger audience. The focus of our paper is on generating a wide variety of homepage banners that can be made as an input for a user-level personalization engine. Following are the main contributions of this paper: (1) We introduce and explain the need for large scale banner generation for e-commerce companies (2) We present on how we utilize existing deep learning based detectors which can automatically annotate the required objects/tags from the image. (3) We also propose a Genetic Algorithm based method to generate an optimal banner layout for the given image content, input components and other design constraints. (4) Further, to aid the process of picking the right set of banners, we designed a ranking method and evaluated multiple models. All our experiments have been performed on data from Myntra (http://www.myntra.com), one of the top fashion e-commerce players in India.

S. Vempati (✉) · K. T. Malayil · R. Sandeep
Myntra Designs, Bengaluru, India
e-mail: sreekanth.vempati@myntra.com; korah.malayil@myntra.com; sandeep.r@myntra.com

V. Sruthi
Microsoft, Bengaluru, India (Work done while at Myntra)

N. Dokoohaki (ed.), *Fashion Recommender Systems*, Lecture Notes in Social Networks, https://doi.org/10.1007/978-3-030-55218-3_2

Keywords Deep Learning · Machine learning · Computer Vision · E-commerce · Fashion e-commerce · Automated cropping · Automated ads · Ads

1 Introduction

In the current e-commerce era, content on the homepage plays an important role in a customer's journey for most of the e-commerce companies. Significant amount of the homepage on online shopping apps/websites is dedicated to banners. Examples of homepage banners are shown in Fig. 1. These banners play a key role in visual communication of various messages to customers such as sale events, brand promotion, product categories and new product launches. The primary components of a banner are the image, also called a creative, and the landing page associated with it.

Any banner image is an amalgamation of an underlying theme, a background image, the brand logo in focus and affiliated text phrases. The text style consists of font colours, size and typography. Banners essentially come down to being one particular permutation of these components which is visually appealing and can deliver the message. Given a requirement, a designer typically takes these various elements from a library, places these elements as per aesthetics and applies few image transformations to create a banner image.

In this paper, we show how we can automate the process that designers follow, using a set of deep learning and image processing techniques. Usually designers pick images from a large image collection called photo-shoot images. Most of these images are provided by brands or other associated entities. These images are also used on other platforms like public hoardings, offline stores etc. Designers also use product catalogue images on few occasions. These photo-shoot images are taken in good lighting conditions showcasing highlights of the brands.

Due to the notably large amount of manual hours invested in creating these banners, only a small set is made live at any point of time. This reduces both the variety of output produced and the degree of personalization that can be achieved. But if the range of available banners increases, the personal taste of a larger audience can be satisifed. From a wider spectrum of options, we can now accomplish targeted banners that cater to different sectors of customers, instead of generic ones made for all. In this paper, we do not focus on the user personalization engine, which is already internally built, but we focus on large scale generation of the inputs to this engine.

We present a method which generates banner images using a library of design elements. Examples of design elements include background content image, text phrases, logo etc., One of the key design elements is the underlying layout of the banner which determines the spatial arrangement of various design elements like text, logo etc., A layout can be used to describe a group of banners, and can also be used to create new banners with the same blueprint. We use a Genetic algorithm based method which can generate a banner layout given a background image, logo and text. It takes into account, design considerations like the symmetry, overlap between elements, distance between elements etc for assessing layout quality.

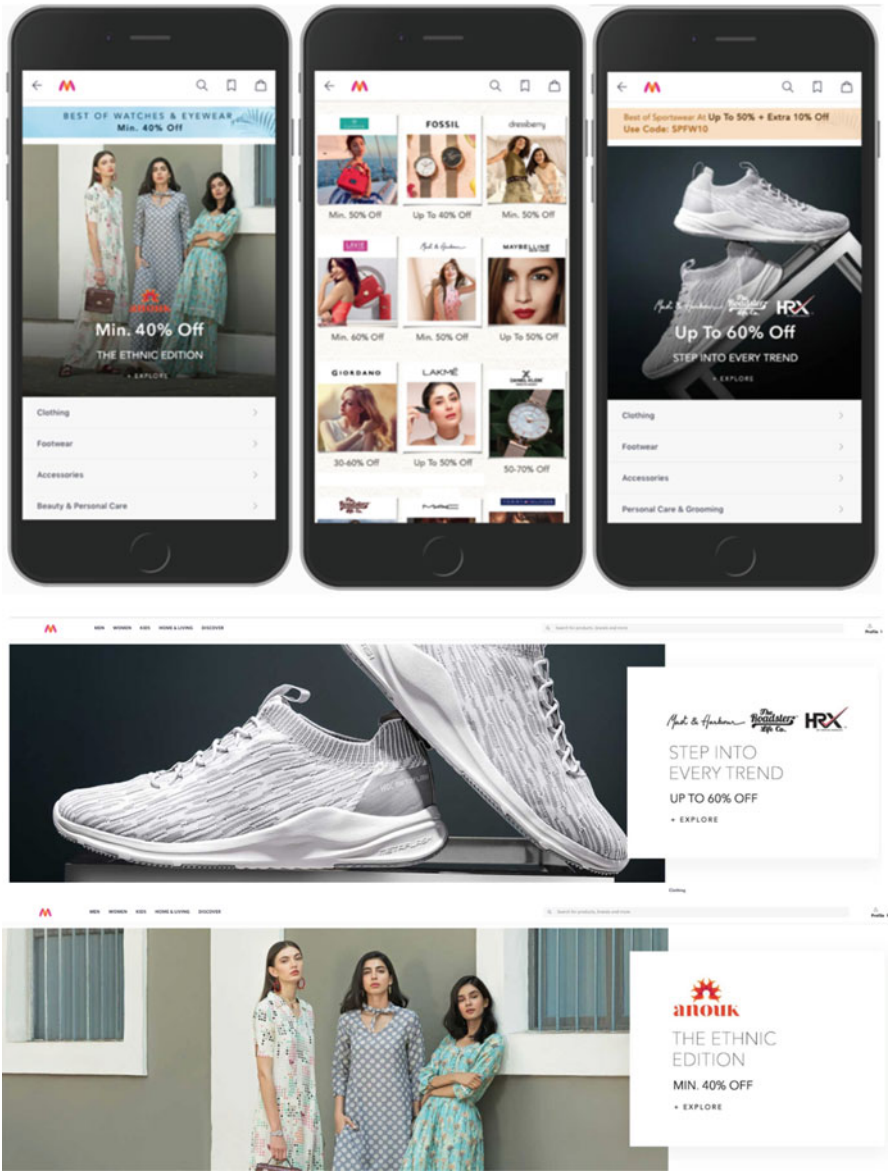


Fig. 1 First row shows examples of Homepage banner/ad images on the Myntra app. Second row shows how same banner image can appear in different formats on desktop

Overall, we can generate the banners using a library of photoshoot images and tag lines for any given themes. As input, we make use of photo-shoot images that are available to the designers in e-commerce companies. We further present an offline process to rank these generated banners. This method constitutes of machine learning models built on banner meta-data.

The use cases of this proposed method are not restricted to e-commerce websites only. It can further be extended to automating banners for social messaging platforms and movies or online video content providers.

In the next sections, we discuss some of the related work to this problem and then we talk about the method for creating the banners, touch upon few challenges and solutions. Further, we present and evaluate the methods for evaluating the banners generated using different design elements.

2 Related Work

As this is an attempt to solve a new problem, there is very limited related work available for the same. We do have some research being done to solve similar problems in other domains. In the past, there have been papers which solve the problem of automated layout design for photo-book [18] and magazine cover [11].

Photo-book design system [18] was built to automatically generate photo compositions such as collages or photo-books using design and layout principles. In their paper, images and text elements are pre-processed and then content is distributed across pages using pre-defined rules. Then, a layout is generated using a set of design principles. Further, genetic algorithm is used to evaluate the layout, whose fitness function takes into account the borders, text elements and overall visual balance of the page.

There is some work on generating new layouts and also transferring the layout using an example [14, 15, 21]. In [14], Generative Adversarial Networks are used to generate new layouts for UX design and clip-art generation. The generator takes randomly placed 2D elements and produce a realistic layout. These elements are represented as class probabilities and bounding boxes. A CNN based discriminator is used as the aim was to be similar to the human eye and spatial patterns could be extracted.

Alibaba's Luban [10, 20] is the most relevant one to our work, but we do not have any published work regarding this. As per the blog, the design team established a library of design elements and allowed Luban to extract and cluster features from raw images. Then, given a blank canvas, it places elements on it in a random fashion and then uses reinforcement learning to learn the good designs.

Photo-book design system [18] was built to automatically generate photo compositions such as collages or photo-books using design and layout principles. In their paper, images and text elements are pre-processed and then content is distributed across pages using pre-defined rules. Then, a layout is generated using a set of design principles. Further, genetic algorithm is used to evaluate the layout,

whose fitness function takes into account the borders, text elements and overall visual balance of the page.

Zhang et al. [21] tackles the problem of multi-size and multi-style designs, i.e. modifying a single design to suit multiple styles and sizes. Automation of layout design by optimizing an energy function based the fitness of a layout style which measures factors such as margin, relative position, etc. In [15], a new banner is being generated based on a given examples using energy function optimization.

Another energy based model is built by [16] targeting single page graphic designs. The images are analyzed to get hidden variables such as importance of different elements and alignment. The energy based model is then based on positions of different elements, balance, white space, overlap area, etc. Design re-targeting is also presented, i.e, transferring the same design to a different aspect ratio. We have adopted some of the energy terms in generating the layout in our work.

In the movies domain, Netflix [1] blogs talk about generating art work which is personalized to the user. The focus here is primarily on finding the best thumbnail for a given movie and user by using reinforcement learning approaches.

For predicting the Click-Through-Rate (CTR) of online advertisements, [6] has trained various models on a large dataset of images. In Deep CTR system [4], a deep neural network is proposed which uses convolution layers to automatically extract representative visual features from images, and nonlinear CTR features are then learned from visual features and other contextual features by using fully-connected layers. There is also work on finding the quality of native ads using image features [23].

3 Methodology

Given that the integral constituents of a banner are the image, logo, text and additional elements, structuring them optimally results in the end product. This structure is defined by a layout, which is essentially the positions and sizes of the various components. Hence the input to automated generation would primarily be a layout and also the image in focus, the brand's logo, text and other associated details. We could use both human provided layout as well as generate a layout based on the given photoshoot image and design elements. We present a layout generation method in the further sections. Different layouts are generated and the best ones are utilized for final creatives generation.

For automation to be possible, we need automated tags to pick the right image for the right theme/purpose. A bare photo-shoot image is composed of various fashion articles spread across it. Examples of fashion articles are "Men-Tshirts", "Women-Jeans" etc. Information regarding this is required for filtering and selecting appropriate ones. Thus the first step would be to perform large scale automated annotation of all images and tag each of them with relevant data.

Once annotation is complete, this newly obtained information is given as input to the layout generation module and further to creation module. The region of interest

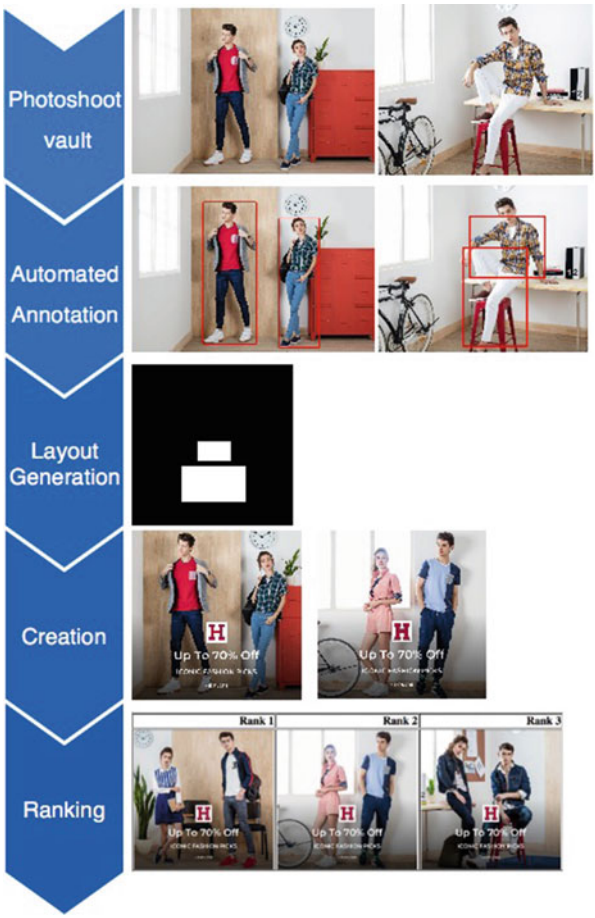


Fig. 2 End-to-end pipeline for automated creation of banners

is extracted, and the different elements are stitched together according to the layout specifications. The final result produced is a large number of banners for the given theme. Since only a few from this pool of options would actually be required , these are further re-ranked by a model built on historical data. End to end steps involved in the banner creation and ranking can be found in the Fig. 2.

3.1 Automated Annotation of Photo-Shoot Images

Automated annotation involves extracting the meta-data of an image. Simpler attributes like brand name and season name are given as a label. For all the other attributes, we need visual understanding of the image which is done by using a set of detectors for each attribute. The constituents of the images are tagged based

on categories or bounding boxes. A bounding box is a rectangle that completely encloses the object in focus and is described by coordinate points.

The different aspects of annotation ranges from the significant objects such as people or the main article, along with their types to the secondary level details such as a person's gender, the type of scenery of the image, and the number of articles present in each category. We explain more details for each of the aspects below.

3.1.1 Object and Person Detection

The various objects present in the image were detected using the MaskRCNN [9] object detector. MaskRCNN was built for the purpose of instance segmentation, which is essentially the task of detecting and delineating distinct objects of interest in an image. The state of the art instance segmentation techniques are R-CNN [8], Fast-RCNN [17], Faster-RCNN [17] and MaskRCNN [9]. MaskRCNN was chosen as it addresses the issues of Faster-RCNN and also provides pixel-to-pixel alignment.

In particular, detecting people is of more interest to us as most of the times the article in focus is on or near them. Thus images were tagged with the bounding boxes of people present and additional information such as total number of people, dominant person, etc. We have used pre-trained detector for our person/object annotations.

3.1.2 Fashion Category Detection

Though detecting the people in image narrows down the region of interest, the actual focus is always on the product that the banner was originally meant to be created for. Tagging this product will help us give significant importance to it. Thus, to identify the various fashion categories present in an image, an in-house detector was used. This fashion category detector was built using Mask RCNN architecture [9] and was trained on fashion specific categories like shoes, watches, etc., The training data for this detector contained manually labelled bounding boxes and class labels for the fashion categories. Example detections are illustrated in Fig. 3. The mean average precision of this detector (mAP) is 67.9%. This value is inline with the mAP obtained for the famous Pascal VOC 2007 dataset which is 63.4%.

This detector provides a bounding box of the category along with its type. The types of categories include top-wear, shoes, watches, etc. Additionally, the image was also tagged with the dominant article present in terms of area. This would later be useful in filtering images based on the articles present.

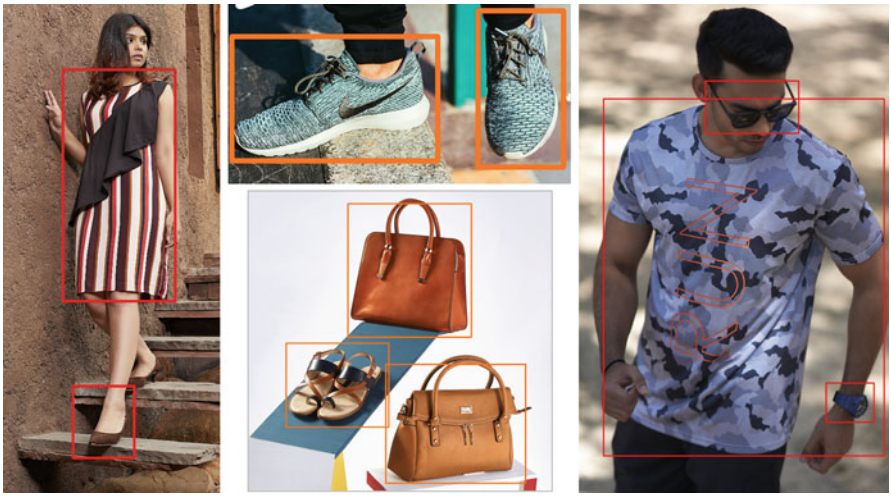


Fig. 3 Example images showing how the Fashion Categories detected from the photoshoot images. We can see the detected objects includes regions covering wide range of categories such as footwear, clothing, bags, etc.

3.1.3 Gender and Face Detection

Apart from the entire person's body, detecting the face will be of more use. This is due to the fact that in certain cases it is okay to overlap other elements on a person's body, but the design elements should not be present on the face. Tagging the gender of the people present will again help filter images based on any particular theme requirement. A CNN-based gender detection model [19] trained on the IMDB-Wiki dataset was used for this purpose.

3.1.4 Scene Detection

Using the scene detector [22], we obtain the various attributes of the background of an image. The categories of scenes include indoor, outdoor as well as details like restaurant, garden, etc and features such as lighting level and man-made area. This level of annotation will help filtering images for both theme based creatives generation and for better personalization.

3.1.5 Text Detection

We perform text detection on photoshoot images so as to remove few images which have too much text area in the image and are not suitable for generation of creatives. Text was detected using the OpenCV [2] East Text detector.

3.2 Layout Generation

A layout is defined as the set of positions/placements for each of the design elements like "brand logo", "text callouts" etc., on a given content image consisting of people and/or objects along with their bounding boxes. A layout L can be defined as $\{\theta_1, \theta_1 \ldots \theta_n\}$ where θ_i represents the co-ordinates of the bounding box for the ith design element. Our objective is to find the co-ordinates θ_i which form the layout with highest aesthetic value.

Layout generation involves evaluating aesthetic compatibility for all combinations of possible locations to properly place and scale the design elements in the banner. Since there is a very large number of combinations of the possible coordinates, it is time-consuming to compute a feasible solution out of all the combinations. For this purpose, we have used Genetic algorithms, as they are suitable for problems involving large combinations of variables. Genetic algorithms have been proven to help in converging quickly to a workable solution and the random mutations also take care of generating new unseen layouts.

Genetic algorithms [13] simulate the natural process of evolution and uses the following techniques to reach the best possible/fittest solution. It starts with an initial population and performs the following steps to reach the best possible solution. In our case, each of the population corresponds to one layout configuration of all the design elements.

(a) **Selection:** Chooses the best parents to produce the next generation from a population
(b) **Crossover:** Combines two parents to create new individuals. In our case, this involves swapping coordinates between the various design elements like text boxes/logo.
(c) **Mutation:** Randomly changes genes/points on individuals to create new individuals. This helps in evaluating new coordinate points.
(d) **Fitness Function:** Uses a fitness score to evaluate the population. Individuals having a higher fitness score are chosen as the next parents.

The Distributed Evolutionary Algorithms in Python (DEAP) library [7] was used for the implementation. The algorithm generates a random population with x and y coordinates for the logo and text bounding boxes. The bounding boxes of the persons and objects in the photoshoot image is considered as fixed. These coordinates are the inputs for the model. The algorithm performs a series of selection, crossovers and mutations on the logo and text coordinates to come up with the best solution based on the fitness function and the constraints provided. The fitness function incorporates the fundamentals of graphic design by aggregating scores for each of the design aspects. Once a specific number of generations are produced, the individual corresponding to the best score is chosen as the output.

The final fitness/energy score for a layout, $E(X, \theta)$, is the weighted sum of individual scores, $E_i(X, \theta)$. One such layout assessment is showcased in [16]. The weights for the individual fitness scores were obtained by regression of these scores

on the CTR of historical banners. We've used CTR, as it helps in decoding user preferences, there by helping us in mapping the weights for different design aspects to user preferences.

$$E(X, \theta) = \sum_{i=1}^{n} w_i E_i(X, \theta)$$

X is the background layout which includes the x and y positions of the bounding box for the person/object in the image. It is denoted as an array $[x_{left}, y_{top}, x_{right}, y_{bottom}]$ which corresponds to the top-left and bottom-right coordinates of the bounding box. θ represents the coordinates for each element and hence is an array of 4 elements. w_i represents the weight for the ith fitness term. Key individual fitness scores are explained below. Note that the overall fitness function can be easily modified to incorporate more design rules.

Alignment Alignment is one of the core designing considerations while creating a banner. We calculate the misalignment metric which penalizes layouts where the elements are misaligned. For cases with left alignment, we have a lower penalization.

Overlap Overlapping of any two of the elements, significantly reduces the aesthetic value of the banner. We calculate the overlap percentage for all pairs of elements and penalize them in the fitness score.

$$Overlap\% = \frac{Area_{overlap}}{Area_{total}}$$

Distance Between Elements Even in cases of zero overlap between the elements, there can be cases where they are placed very close to each other. This is especially discomforting when the logo or text are placed very close to the face of the person or the important region of an object in the background image. Hence layouts with elements placed farther apart are preferred. The euclidean distance is calculated between pairs and added:

$$Distance(i, j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

$$\forall (i, j)\ i \neq j$$

$$i, j \in \{Person, Logo, Text, \ldots\}$$

Symmetry Symmetry of the layout is a key factor in stabilizing the overall balance of the final layout. To account for this, we calculate the asymmetry for all elements in the image layout and add this as a penalization term in the fitness score.

$$X_{center} = \frac{(X_{left} + X_{right})}{2}$$

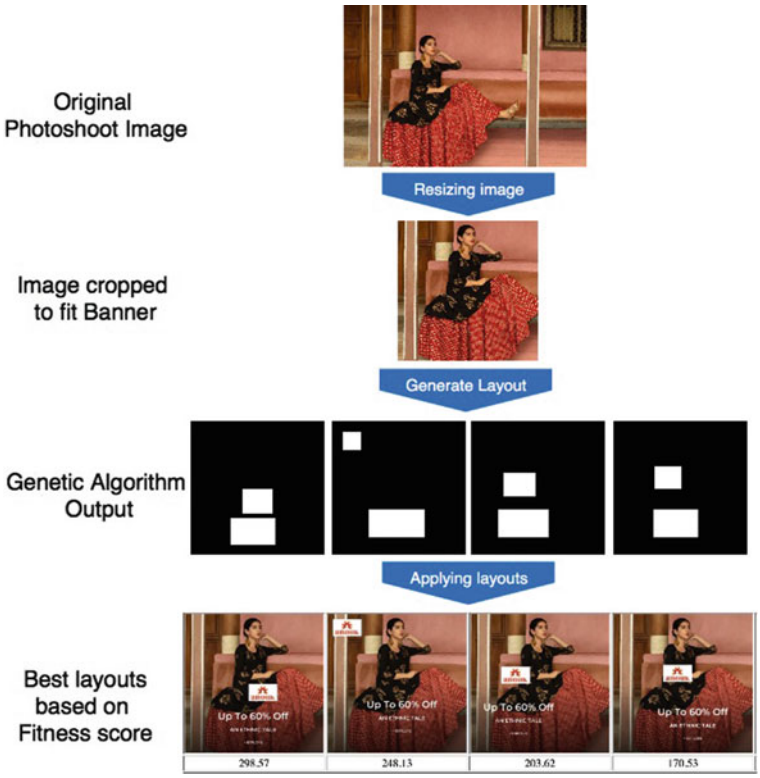


Fig. 4 The steps involved in layout generation, and applying them on creatives. The generated layouts and their respective fitness scores are observed. The scores are found to be congruent with the aesthetic value of the banner

$$Asymmetry_{horizontal} = |2 * X_{center} - Width_{layout}|$$

Constraints To make sure that all elements are of reasonable size and not too large, the bounding boxes for the elements are assigned a buffer region. All layouts where the dimensions fall outside this region, we term those as in-feasible solutions.

For illustration purposes, we have tried to maintain uniformity within a single brand/category level creative and hence the layout that performs best on majority of the cases is shown in the examples. An example result is illustrated in Fig. 4.

3.3 Creative Generation

Combining everything together, creative generation involves the following steps for a given library of photoshoot images with annotations, brand logos, text callouts:

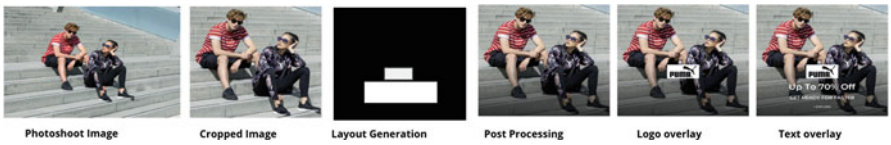


Fig. 5 Above figure illustrates the prominent steps involved in creating the banner creative. Given a photoshoot image, it is automatically cropped according to the given requirement of size and category, then best layout is computed using the genetic algorithm based method. After applying image post processing steps, design elements like logo and text call-outs are placed as per the layout specifications

- Filtering the relevant photoshoot images for the given brand or category using automated tags. Brand name is already provided for the images as labels.
- Automatic cropping of the selected photoshoot images to find the region of interest using annotations.
- Generating best layouts for the given cropped image, logo and text call-outs.
- Overlaying of the different design elements as per the layout specification. Details on how the text is selected and overlaid along with few post-processing steps are explained below.

All the steps involved in creative generation are illustrated for an example in the Fig. 5.

3.3.1 Cropping and Scaling Input Image

The input images are most often of a different size or aspect ratio than the final banner that we require. We need a technique that crops and scales the input image such that the person/object of interest is focused and centered and the quality of the image is not reduced. We have tried out the popular seam-carving technique and also designed our own algorithm to solve this problem. A comparison of the techniques is illustrated in Fig. 6.

1. **Seam Carving [24]**: It is an energy based approach which supports content aware image resizing by deleting or adding seams (a path of minimum energy cost) on the input image to arrive at the desired aspect ratio
2. **Smart Crop [25]**: It is an algorithm that detects the edges, regions of high saturation, etc. and generates a set of candidate crops using sliding windows that are ranked by an importance function. The one with the highest rank is chosen as the output.
3. **Our approach**: We have implemented a recursive cropping and resizing algorithm that progressively adjusts the bounding box containing the person/object of interest such that we reach the required aspect ratio and size. We have come up with the constraints and cut-off values by trial and error, and have implemented

Fig. 6 Top: Original image, Bottom from left to right: Cropped using Seam carving, Smart Crop and Using our approach. Our approach yields the best output image with the person of interest being centered and focused

this algorithm in the final approach. It has been found that our algorithm crops and resizes an input image, better than the above approaches.

3.3.2 Overlaying Text Callouts

The text on a banner signifies the core message and intention for creating it. Be it a sale or a new arrival, catchy phrases are used to attract the customer. We have a collection of text phrases for various new launches, brand call-outs, sale events. We select the appropriate text to be overlayed from this collection.

Algorithm 1 Image cropping algorithm: Our approach

```
Load input image and output dimensions
while Image dimensions/Aspect-ratio not less than cut-off value do
    Adjust image boundaries without cropping the area of interest
    if Aspect ratio different from required then
        Continue
    else
        if Image dimensions different from required then
            Resize image to output size
        else
            Result obtained
        end
    end
end
```

Text Formatting We have a few pre-defined standard text colours, size and fonts based on banner images served on the platform in the past. These were obtained using certain existing design rules, such as golden ratio and the rule-of-thirds. The golden ratio(divine proportion) acts as a typography ratio that relates font size, line height, and line width in an aesthetically pleasing way.

Post Processing While adding text on an image, we have to ensure that the font color is appropriate. To allow even lighter font colors to be visible, a slight darker gradient is applied on the image.

3.4 Ranking Creatives

So far, we have explained ways of automatically generating the banners. Out of these numerous banners generated, we would prefer the superior ones to go live. In order to eliminate the manual effort put in picking these banners, we have designed a ranking methodology based on historical live data-set which had images along with their associated Click-Through-Rate (CTR) as labels. This methodology can be used to rank manual as well as automatically generated creatives. We consider CTR as a signal for goodness of a creative. Note that we do not have any ground truth labels for the generated creatives. We test the goodness of our creatives with the help of the ML model trained on the historical dataset (creatives which are manually designed) which already has a label in terms of CTR. We have trained different models on this dataset to predict CTR given a generated creative.

Note that the layout generation algorithm explained in earlier sections does the job of finding the best layout whereas the method explained in this section helps in ranking all the creatives and considers features which not only explains about layout, but also the content in the image and their historical performance.

Feature Engineering The model was built on both image embeddings, explicit features designed from the layout of the image and a set of features determining aesthetic score of the image.

VGG Embeddings The VGG convolutional network [3] that has been pre-trained on the ImageNet database provides embeddings for an image of dimension 4096. Input to the network requires the image to be re-sized to 224×224. VGG is just one of the pre-trained models that can be used for getting the image features, other models can also be used for this purpose.

Layout Extracted Features Using the various annotation means, we can engineer features from different components of an image.

1. Position specific features: The coordinates of bounding boxes for people, text, faces and articles.
2. Area: The relative area percentage covered by each of the dominant objects.
3. Gender: Number of women present, Number of men present, Total number of people present.
4. Category Type: The types of articles detected are one hot encoded (present or not) . Types include topwear, bottomwear, watches, etc.
5. Environment Type: Indoor or outdoor background setting.
6. Scene Categories and Attributes: Frequently occurring categories (restaurant, garden, etc) and attributes(lighting,man-made, etc.) were picked and one-hot encoded.
7. Overlapping objects: When text is overlayed on the image, it will overlap on the existing components such as a articles or a person's body. This overlap is tolerable as long as the main article of focus or a face is not hidden. To account for this, the relative area percentage of overlap between each of the following components are calculated:
   - Text regions and Faces
   - Text regions and People
   - Text regions and Articles
8. Text Quadrants: One hot encoded information for every quadrant if it contains a text component or not.

Aesthetic Features We have used Neural Image Assessment (NIMA scores) [5] which computes scores representing aesthetics of an image. This is obtained by training a deep CNN on a dataset containing images along with human judgment on aesthetics. In our experiments, we have obtained the score by using this publicly available pre-trained model [5]. For a given image, the aesthetic scores predicted by this model was used as additional feature.

Ranking Models Apart from the simple Logistic Regression model, the tree based classifiers were chosen as they are popular methods for CTR prediction problems. Note that other methods based on deep learning [4] could further improve the ranking methodology.

Here are the methods that we have experimented along with the optimal parameters picked.

1. Logistic Regression
2. Decision Trees
3. Random Forest Classifier

As there are fewer clicks compared, we have balanced the data by providing higher weights to samples with clicks.

Ad Personalization Engine In order to provide a personalized experience, relevant ads are being chosen and shown to the user. For this purpose, all the active ads will be ranked using prediction scores of a click-prediction model. This model is trained on historical click-stream data with user and ad features as input variables. This model is already deployed on production and is not the focus of this work.

4 Experiments and Results

4.1 Qualitative Evaluation of Generated Layouts

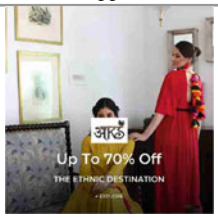
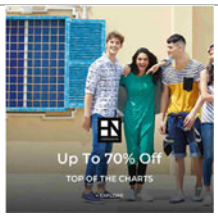
The generated layouts are sorted according to their fitness scores and the top one is selected. We carried out an internal survey asking users to identify the designer created layouts and the machine generated ones. It was observed that 60% of the people were not able to distinguish between them. An example result is illustrated in Fig. 4.

4.2 Qualitative Evaluation of Cropping and Scaling Algorithm

4.2.1 Baseline Approach

As a baseline approach for generating creatives, we can crop the center region of the given photoshoot image with required aspect ratio. Further steps include pasting brand logo and text in the same layout as used in the creative generated. Some image processing is done onto the banner creative. Note that this approach doesn't consider regions of any objects/people present in the image. Results using this approach can be seen in Table 1.

Table 1 Few qualitative results with baseline and our cropping and scaling approach

Photoshoot image	Baseline approach	Our approach
		
		
		
		

4.3 Qualitative Evaluation of Generated Creatives

Figure 7 shows examples of banners which were generated by designers and our approach. Figure 8 demonstrates how article-type based creatives are generated. To evaluate the goodness of the generated creatives w.r.t designer created banners, we had conducted an internal survey. In the survey, a user was presented with two images, a designer created banner and automated banner. We found out that only 45% of the people were able to judge correctly, showing that we were able to create banners which are realistic.

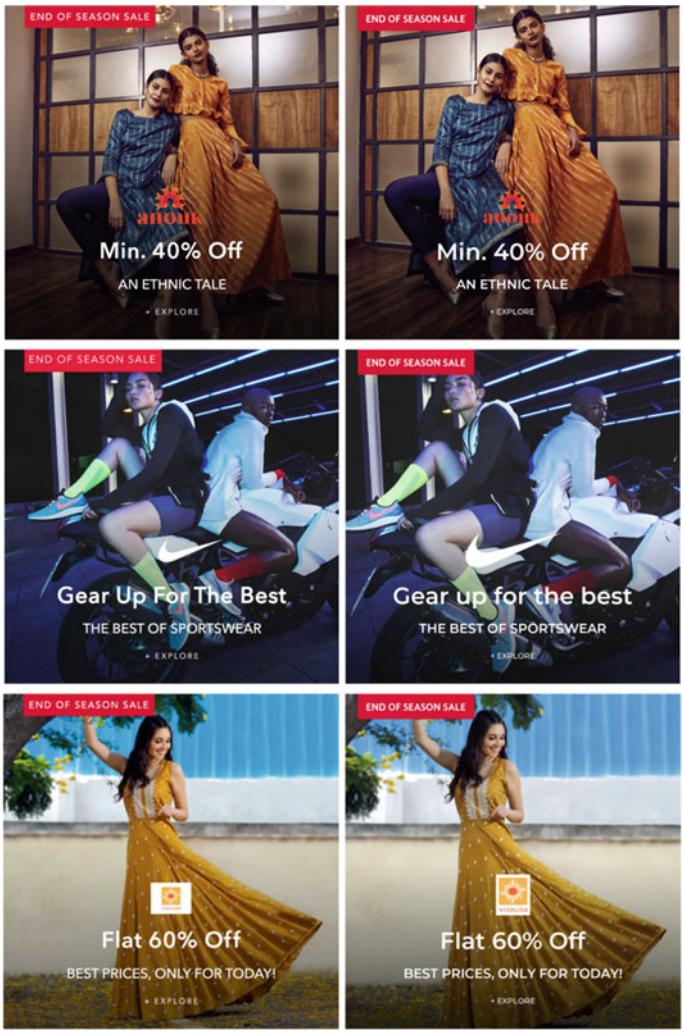


Fig. 7 The image on the left was created by a designer while the one on the right was automatically generated.

4.4 Evaluation of the Ranking Model

In order to evaluate the goodness of the proposed ranking approach, we have conducted few offline experiments. Details are explained below.

Dataset All the images and data used for our experiments are taken from Myntra (http://www.myntra.com), one of the top fashion e-commerce players. We have trained the models on nearly 20,000 existing production banner images which

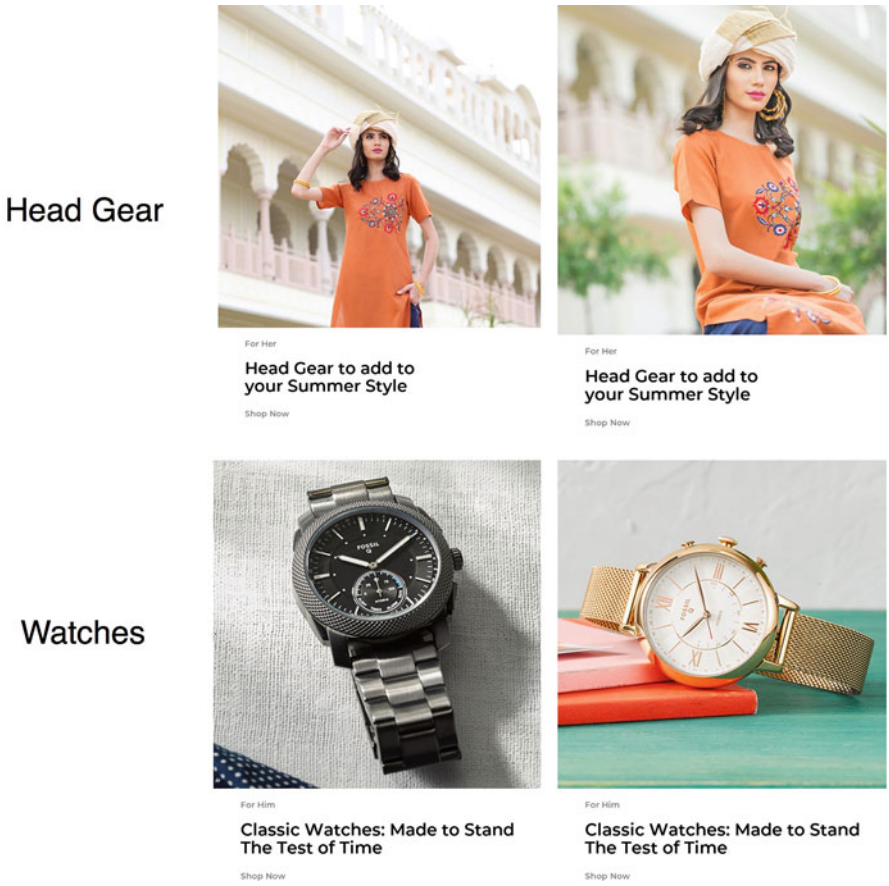


Fig. 8 Creatives generated for different article types. First row correspond to the "head gear" category and second row shows creatives generated for "watches" category

already have labels. We perform the feature extraction on all the images in the dataset and then use the target variable as is_clicked. The models were trained on 75% of the data and tested on 25%.

4.4.1 Evaluation Metrics

Classification models were evaluated using both AUC (Area Under Curve) and NDCG (Normalized Discounted Cumulative Gain) [12]. NDCG is a standard information retrieval measure [12] used for evaluating the goodness of ranking a set. For computing NDCG, we have used CTR as relevance score. Evaluation Metrics on test data using different models is presented in the Tables 2 and 3.

Table 2 Evaluation metrics for different feature sets experimented on best performing model (Random Forests)

Features	AUC	NDCG
VGG Embeddings	0.71	0.17
Layout extracted features	0.74	0.14
NIMA	0.71	0.24
VGG Embeddings + Layout Features	0.71	0.17
Layout extracted features + NIMA	0.72	0.56
VGG + Layout features + NIMA	0.71	0.22

Table 3 Evaluation metrics for different models experimented on combined feature set i.e., VGG embeddings, NIMA and Layout extracted features

Model	AUC	NDCG
Logistic regression	0.60	0.03
Decision trees	0.70	0.05
Random forests	0.71	0.22

4.4.2 Quantitative Evaluation

From the various models and features experimented with, the promising results are shown in Tables 2 and 3. Apart from training on VGG embeddings, NIMA scores [5] and layout extracted features individually, a combination of all was also attempted. Since the best performing model was the Random Forests Classifier, the results for it are present in Table 2. Performance metrics using different models for combined feature set is present in Table 3. We can see that Random Forests using Layout and NIMA features gives the best NDCG.

It is also interesting to observe that when the model was trained only on the layout extracted features, the most important features were : area of the image, overlap area between text and objects, certain text region coordinates, overlap area between text and people, etc. This further reiterates the fact that the position and orientation of text defines the layout of an image, and it is useful to generate banners without assumptions about their positions.

4.4.3 Qualitative Evaluation

When the model trained on historical data was used to predict CTR on the newly generated creatives, the results were quite similar to a what the human eye would observe.

Figure 9 is an example of images that have high predicted CTR. This seems to be a meaningful outcome as these images have good color contrast ratio, optimal placement of the components and visible text. In Fig. 10, we notice how the images with poor lighting, faces turned away and ones with unnecessary extra padding space are all pushed down the ranking scale due to much lower predicted CTR.

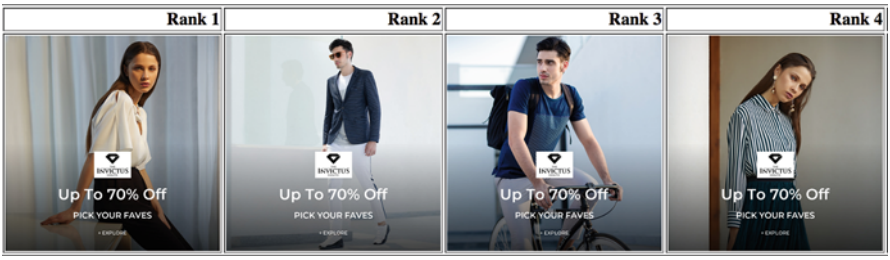


Fig. 9 Examples of images with high predicted CTR for the ones generated under a single brand

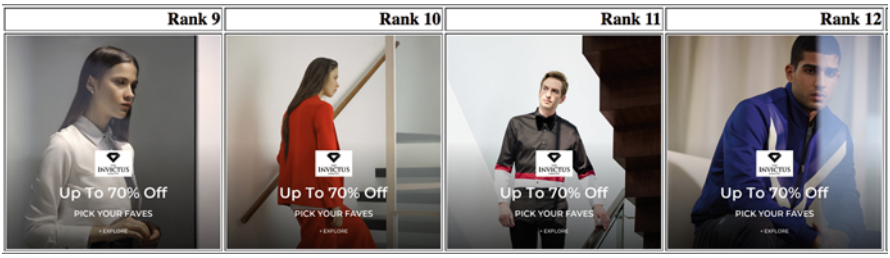


Fig. 10 Examples of images with low predicted CTR for the ones generated under the same brand. We found that ranking model is inline with human judgment

4.5 Evaluation of the Complete Approach

Online Evaluation We performed an online A/B test of the auto-generated creatives along with the manually designed creatives for the same brands. As the main hypothesis was that having multiple options increases personalization, we had a larger set of automated banners in the test set compared to the control set. For both buckets of the A/B test (equal proportion of user traffic), we have used the same personalisation engine that considers both user preferences and banner features. The control group contains manually designed banners and the test group contains automated banners. The results have shown that CTR has increased by 72% for the test set compared to control set (relative increment), with high statistical significance.

5 Applications

We can utilize the above approach for multiple use cases like

1. Brand campaigns: Generating numerous banners for a Brand , from a given corpus of photo-shoot images with variations of images and messaging.

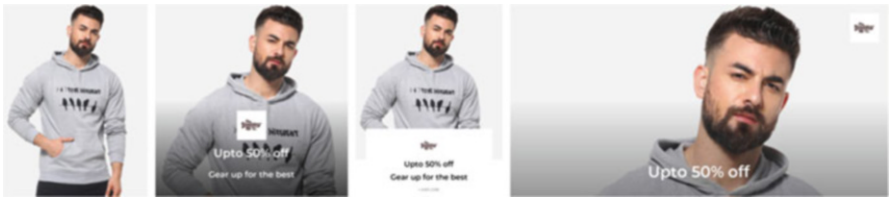


Fig. 11 Multiple templates for a catalogue image. From left to right: (**a**) Raw photoshoot image, (**b**) Square banner, (**c**) Carousel banner, (**d**) Horizontal banner

Fig. 12 Multiple templates for a photo-shoot image. From left to right: (**a**) Raw image, (**b**) Template 1, (**c**) Template 2, (**d**) Template 3

2. Category banners: Searching for category type in the annotated data, and creating banners using the images identified.
3. Dynamic banners: We can generate banners based on user's recent browsing activity by extracting brand, category and other attributes information from the browsing data and feeding into our approach.
4. Multiple templates: Creating banners of different templates for the same image, for using in different pages (homepage, search pages, product listing pages etc.,) on the app. Examples can be seein for catalogue images in Figs. 11 and 12
5. Digital Marketing: The banners can also be used for digital marketing purposes on social media platforms like Instagram, Facebook and ad platforms like Google Ads etc.,

6 Conclusion

In this paper, we have presented a method to generate banner images in an automatic manner on a large scale. This helps in reducing the massive effort and manual hours spent by designers currently and also to produce a wider variety of options to pick from. The broader spectrum of banners will help in catering to wide spectrum of users, instead of showing common banners to all users. We have presented end to end steps involved in generating creatives given few input constraints using automated annotation of photoshoot images. We described a genetic algorithm based method to generate a layout given cropped image and other design elements.

We have shown how a ranking model can be trained on historical banners to rank these generated creatives by predicting their CTR. We observed that when the best performing model was tested on these automatically produced banners, the ranking results was very similar to what a human would've picked, with well positioned, optimal images having higher CTR than the rest. Apart from this offline method of ranking them, future work would be to perform online ranking via reinforcement learning, which will also further boost the personalisation by showing the right set of banners from the vast amount of banners created.

References

1. Amat F, Chandrashekar A, Jebara T, Basilico J (2018) Artwork Personalization at Netflix. In: Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18). ACM, New York, p 487–488. https://doi.org/10.1145/3240323.3241729
2. Bradski G (2000) The OpenCV Library. In: Dr. Dobb's J Soft Tools (2000), 25:120,122–125
3. Chatfield K, Simonyan K, Vedaldi A, Zisserman A (2014) Return of the devil in the details: delving deep into convolutional nets. CoRR abs/1405.3531. arXiv:1405.3531. http://arxiv.org/abs/1405.3531
4. Chen J, Sun B, Li H, Lu H, Hua X-S (2016) Deep CTR prediction in display advertising. In: Proceedings of the 24th ACM International Conference on Multimedia. ACM, pp 811–820
5. Esfandarani HT, Milanfar P (2017) NIMA: neural image assessment. In: CoRR abs/1709.05424 (2017). arXiv:1709.05424. http://arxiv.org/abs/1709.05424
6. Fire M, Schler J (2015) Exploring online ad images using a deep convolutional neural network approach. In: arXiv e-prints abs/1709.05424. arXiv:1509.00568. https://arxiv.org/abs/1509.00568
7. Fortin F-A, De Rainville F-M, Gardner M-A, Parizeau M, Gagné C (2012) DEAP: evolutionary algorithms made easy. J Mach Learn Res 13:2171–2175
8. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 580–587
9. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask R-CNN. arXiv e-prints. Article arXiv:1703.06870. arXiv:1703.06870
10. Hua X-S (2018) Challenges and practices of large scale visual intelligence in the real-world. In: Proceedings of the 26th ACM International Conference on Multimedia (MM '18). ACM, New York, pp 364–364. http://doi.acm.org/10.1145/3240508.3267342
11. Jahanian A, Liu J, Lin Q, Tretter D, O'Brien-Strain E, Lee SC, Lyons N, Allebach J (2013) Recommendation system for automatic design of magazine covers. In: Proceedings of the 2013 International Conference on Intelligent User Interfaces. ACM, pp 95–106
12. Järvelin K, Kekäläinen J (2002) Cumulated gain-based evaluation of IR techniques. ACM Trans Inf Syst 20(4):422–446
13. Man KF, Tang KS, Kwong S (1996) Genetic algorithms: concepts and applications. IEEE Trans Ind Electron 43(5):519–534
14. Li J, Xu T, Zhang J, Hertzmann A, Yang J (2019) LayoutGAN: generating Graphic Layouts with Wireframe Discriminator. In: International Conference on Learning Representations. https://openreview.net/forum?id=HJxB5sRcFQ
15. Maheshwari P, Bansal N, Dwivedi S, Kumar R, Manerikar P, Balaji Srinivasan V (2019) Exemplar based experience transfer. In: Proceedings of the 24th International Conference on Intelligent User Interfaces (IUI '19). ACM, New York, pp 673–680. https://doi.org/10.1145/3301275.3302300

16. O'Donovan P, Agarwala A, Hertzmann A (2014) Learning layouts for single-page graphic designs. IEEE Trans Vis Comput Graph 20(8):1200–1213
17. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp 91–99
18. Sandhaus P, Rabbath M, Boll S (2011) Employing aesthetic principles for automatic photo book layout. In: International Conference on Multimedia Modeling. Springer, pp 84–95
19. Uchida Y (2018) Age-gender-estimation. https://github.com/yu4u/age-gender-estimation
20. Xu R (2017) EAI visual design is already here. https://medium.com/@rexrothX/
21. Zhang Y, Hu K, Ren P, Yang C, Xu W, Xian-Sheng Hua (2017) Layout style modeling for automating banner design. In: Proceedings of the on Thematic Workshops of ACM Multimedia 2017. ACM, pp 451–459
22. Zhou B, Lapedriza A, Xiao J, Torralba A, Oliva A (2014) Learning deep features for scene recognition using places database. In: Advances in neural information processing systems, pp 487–495
23. Zhou K, Redi M, Haines A, Lalmas M 2016. Predicting pre-click quality for native advertisements. In: Proceedings of the 25th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, pp 299–310
24. Avidan S, Shamir A (2007) Seam carving for content-aware image resizing. In: ACM SIGGRAPH 2007 papers, 10–es.
25. Smart crop implementation https://github.com/smartcrop/smartcrop.py

Two-Stage Session-Based Recommendations with Candidate Rank Embeddings

José Antonio Sánchez Rodríguez, Jui-Chieh Wu, and Mustafa Khandwawala

Abstract Session-based recommender systems have gained attention recently due to their potential for providing real-time personalized recommendations with high recall, especially when compared to traditional methods like matrix factorization and item-based collaborative filtering. Two recent methods are Short-Term Attention/Memory Priority Model for Session-based Recommendation (STAMP) and Neural Attentive Session-based Recommendation (NARM). However, when we applied these two methods to the similar-item recommendation dataset of Zalando, they did not outperform a simple collaborative filtering baseline.

Aiming for improving similar-item recommendation, in this work we propose to re-rank a list of generated candidates, by employing the user session information encoded by an attention network. We confirm the efficacy of this strategy when using a novel **Candidate Rank Embedding** that encodes the global ranking information of each candidate in the re-ranking process. Offline and online experiments show significant improvements over the baseline in terms of recall and MRR, as well as improvements in click-through rate. Additionally, we evaluate the potential of this method on the next click prediction problem, where, when applied to STAMP and NARM, it improves recall and MRR on two publicly available real-world datasets.

Keywords E-commerce · Recommender systems · Neural networks · Session-based recommendations

1 Introduction

Recommender systems have become a useful tool for users to explore e-commerce sites, helping users narrow down a vast assortment that includes thousands to millions of products. These sites offer different types of recommendations, tailored

J. A. Sánchez Rodríguez (✉) · J.-C. Wu · M. Khandwawala
Zalando SE, Berlin, Germany
e-mail: jui-chieh.wu@zalando.de; mustafa.khandwawala@zalando.de

N. Dokoohaki (ed.), *Fashion Recommender Systems*, Lecture Notes in Social Networks, https://doi.org/10.1007/978-3-030-55218-3_3

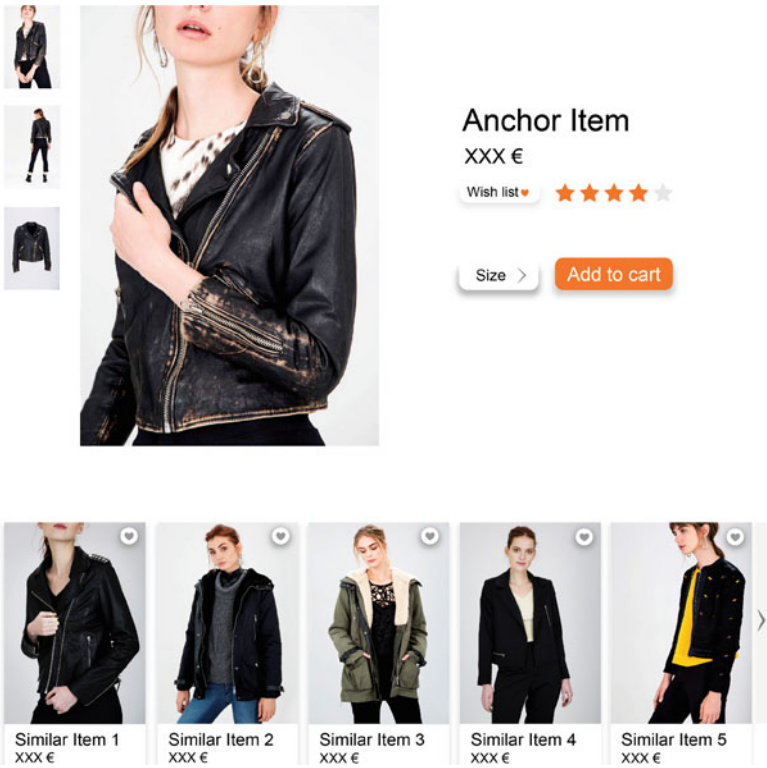


Fig. 1 Example of a Product Detail Page. The similar-item recommendations are shown under the heading "How about these similar items?". This type of recommendations allows customers to find alternatives to the product they are currently seeing

to meet the specific needs of the user, such as finding items similar to a given product [10], or finding a product that complements a given one [18].

In fashion e-commerce platforms such as Zalando,[1] one of the most widely used recommendation tools is the similar-item recommendations carousel. These recommendations are displayed on each Product Detail Page (as shown in Fig. 1), which is one of the pages with more traffic in the platform. These similar-item recommendations offer each user with different alternatives related to the product that the user is currently browsing. For example, the similar-item recommendations carousel displayed in the bottom part of Fig. 1 shows several jackets that might be similar to the product displayed in the top-left part of the Product Detail Page, which is called anchor product.

In this work we present how we improve the similar item recommendation in a production system by employing a two-stage approach: borrowing from recent work

[1] https://en.zalando.de

on session-based recommendations to rank the item candidates generated by a base recommender. Recent advances in session-based recommendations [5, 6, 9, 11, 17] have the capability of encoding information about the temporal evolution of user actions. We propose to extract the ranking information from a pre-trained/pre-calculated model – already adapted for the recommendation task – by using Candidate Rank Embeddings (CRE) in a session-based recommender to have a more powerful personalized recommendation system. In addition we show that using the proposed Candidate Rank Embeddings approach to re-rank the recommendation list generated by some of the session-based recommendation algorithms improves the performance of a next click prediction task, compared to the baselines.

Recent studies on session-based recommendations have shown significant improvements compared to collaborative filtering approaches in several publicly available datasets [5, 6, 9, 11, 17]. We implemented two of the best-performing session-based recommenders, STAMP [11] and NARM [9] to improve our existing similar-item recommendations. We evaluated these algorithms in a proprietary dataset for the task of predicting the next clicked item in the similar-item recommendation presented to users. In the initial evaluation, we were unable to obtain any improvement (in terms of recall@20) over the results with a collaborative filtering baseline [1]. This counter-intuitive phenomenon has also been observed and reported by other recent studies in the literature [12, 16]. Full analysis of this disparity in performance is multi-faceted and complicated. For example, one possible cause could be the bias from the feedback loop existing in the production system. Alternatively, it could be caused by the intrinsic nature of the user behavior presented in a different dataset. Figuring out the exact reasons and applying counter-measures to fight against this phenomenon belongs to another research area, and is therefore not the focus of this paper. The objective of this work is to improve the similar item recommendation in an existing production system, which has been proven successful previously by an online A/B test.

Our hypothesis is that with the proper use of personalized information, the resulting model should be able to outperform the baseline collaborative-filtering algorithms in both offline and online evaluations. To effectively use the personalized information of user sessions, we have to overcome the performance issue of directly applying session-based models to the Zalando Fashion-Similar dataset. Earlier results indicate that a single session-based model seems to struggle in capturing global information between items in the dataset. To address this issue, we propose to use Candidate Rank Embeddings (CRE) together with a session-based recommender to collect such information from the pre-trained/pre-calculated model.

Following the approach of Covington et al. [2], in the first stage a **candidate generator** was employed to take a given user's click history X_u as input to generate a sorted list of candidates C of size k. Specifically, item-to-item collaborative filtering (I2I-CF) was used as the candidate generator. In the second stage, C and X_u are fed to the **re-ranker** to produce fine-tuned recommendations. Borrowing the user session encoder of STAMP E_{STAMP}, X_u is encoded and used in the Re-ranker together with an innovative Candidate Rank Embeddings (CRE). CREs learns the

personalized candidate rank preference along with their item preference in the process of training the model to optimize the prediction of the next item of interest. By using CREs, the Re-ranker was enabled to incorporate implicit information from the candidate generator as prior knowledge that helps to guide and calibrate the training of the Re-ranker with the objective of improving the order of the original recommendation.

The offline experiments showed that our CRE-enhanced model can indeed outperform the collaborative filtering baseline. The good performance on the Fashion-Similar dataset suggests the CRE trick may also be applicable to other recommendation tasks such as general next click prediction. This hypothesis was confirmed by combining the CRE-enhanced model with two baselines, STAMP and NARM, and evaluate on the YooChoose 1/4 and Diginetica next click prediction datasets.

To summarize, the two main contributions of this study are as follows. Firstly, we suggest Candidate Rank Embeddings together with a session-based recommender to enhance the performance of I2I-CF in a two-stage approach. The model outperforms the baselines on a Fashion-Similar dataset in terms of Recall and Mean Reciprocal Rank (MRR) at 20. Also, the improvement was confirmed with an online test where we observed significant improvements in Click-through Rate. Secondly, we compared the baselines I2I-CF, STAMP and NARM, and the proposed method using those baselines as candidate generators on the task of next-click prediction. The results show that the model with CREs improves these baselines with respect to both Recall@20 and MRR@20 on two publicly available datasets.

The rest of this paper is organized as follows. In Sect. 2 we discuss the related work. Section 3 states the problem. Section 4 describes our proposed approach. Section 5.1 presents the public and proprietary datasets in which we evaluate our proposed approach, and Sect. 5 presents the experimental methodology and the results obtained. Finally, Sect. 6 summarizes the conclusions and proposes future work.

2 Related Work

2.1 Session-Based Recommender Systems

Factorized Personalized Markov Chains (FPMC) [14] is an early example of a recommender system model that takes the sequence of user actions into account to predict a user's next actions. Before the advent of such models, user interactions were mainly used in simpler methods, such as item-based collaborative filtering [1, 10] or matrix factorization [8]. Later, recurrent neural networks (RNN) became the focus of research on sequence-based recommender systems. An early neural network attempt at using the click sequences of users as input data and considering recommendation as a next-target prediction was [3]. The authors of DREAM [20]

used pooling to summarize the current basket combined with using vanilla RNN later to recommend the next basket. Because Gated Recurrent Units (GRU) have been proven to work better than vanilla RNNs for longer sequences, GRU-based recommender systems have been proposed with different loss functions, such as Bayesian Personalized Ranking (BPR) [13], TOP-1 [6] and their enhancements [5]. On publicly available sequence prediction datasets, these models have shown better results than matrix factorization based methods.

Various authors have proposed different improvements to the GRU approach. In [17], two techniques were proposed to address data sparsity and behavior distribution shift. NARM [9] further increased the predictive power of GRU-based session recommenders by adding an attention layer on top of the output. While achieving a higher recall and mean reciprocal rank (MRR), training of NARM takes longer than pure GRU approaches.

The authors of the Short-Term Attention Model (STAMP) [11] managed to substantially reduce training and prediction time by replacing RNNs with simpler components such as the feature averaging layer and feed-forward networks. On top of that, STAMP applies an attention mechanism on the embeddings of the items in the user history. However, despite promising results on several public datasets, there were studies reporting that some state-of-the-art session-based models do not outperform simple collaborative filtering methods on certain datasets [12, 16]. We found the same phenomenon in our real-world dataset, and that motivated us to design a solution that benefits from personalization without performance degradation.

2.2 Two-Stage Approaches

Two-stage approaches are widely adopted in different recommendation tasks in various domains. Paul Covington et al. proposed using two neural networks for video recommendation on YouTube [2]. With a clear separation of candidate selection and ranking generation, their solution targets especially at multimedia recommendations. They use this cascading approach to solve performance issues, mainly due to the enormous amount of videos on YouTube.

Other studies have employed this technique to improve accuracy. Rubtsov et al. [15] applied a two-stage architecture to improve the quality of music playlist continuation. Similar to us, they used collaborative-filtering to generate candidates and a more complicated algorithm based on gradient boosting for the final prediction. Likewise, we found that the use of the two-stage architecture makes it easy to improve the model performance when applying session-based recommendations to the fashion similar item recommendation. Besides applying a two stage approach to use the session information, we modeled the user-rank preference with Candidate Rank Embeddings.

3 Problem Statement

Session-based recommendation is usually modeled as predicting the next item of interest for the user, given their interaction history. Given an assortment of items I where all possible items come from, the short-term history X_u of a user u consists of a sequence of l items $x_0, x_1, \cdots, x_{l-1} \in I$ that u has interacted with so far. Session-based recommenders aim to shift the next item $x_l = y_{u^+}$ that the user will interact with to the top position of the recommendation list when given X_u. We denote the collection of user interaction sequences as a dataset $\mathcal{D}$, composed of N pairs of user sequence and target item $\{(X_u, y_{u^+}), \text{for } u = 1, 2, \cdots, N\}$.

A session-based recommender produces a sorted list L_Y from all items in a subset $Y \subseteq I$. In most cases, Y is equivalent to I, but it can also come from a selected set of candidates from another recommender. To obtain L_Y, a score s_y for each item $y \in Y$ is calculated and all the items from Y are ranked by their scores in a descending order. The scores of items in Y are denoted as S_Y and the function $rank$ yields the item list L_Y with the new order according to S_Y:

$$L_Y = rank(S_Y) \tag{1}$$

In the following sections, we use $V_x \in \mathbb{R}^d$ to represent the latent vector for an item x. To represent a matrix of feature vectors of items, the notation V_Q is used; where Q is a list or a set of items. The shape of V_Q is $\mathbb{R}^{d \times |Q|}$.

4 Two-Stage Recommender with Candidate Rank Embeddings

Using the naming convention from [2], we call the first recommender of the cascade **candidate generator G**, while the second, known as the **re-ranker R**, ranks the most relevant items from the output of **G**. Both **G** and **R** take a candidate set and a specific user history as input parameters, and later assign a score to each candidate. The final recommendation of the proposed method L_Y is calculated as follows:

$$\begin{aligned} L_{Y_G} &= rank(S_{Y_G}) = rank(G(Y, X_u, \theta)) \\ C, \overline{C} &= L_{Y_G}[:k], L_{Y_G}[k:] \\ L_{C_R} &= rank(S_{C_R}) = rank(R(C, X_u, w)) \\ L_Y &= L_{C_R} : \overline{C} \end{aligned} \tag{2}$$

Where C denotes the k most relevant candidates computed by G and $\overline{C}$ indicates the rest in Y. θ and w denotes the trainable parameters of G and R respectively. In this study, we set θ and w to be independent and do not share parameters. The

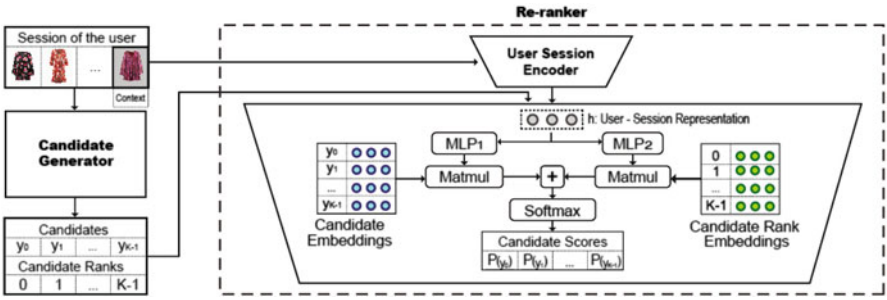


Fig. 2 The model architecture: The candidate generator is trained first and treated as a black box. Then, the re-ranker is trained to re-score the candidates provided by the generator. For calculating these scores, two components are considered: the candidates and the rank preference of them

training process of the proposed model is also two-staged. G is trained first (in case of training needed). The training of R starts after a well-trained G is obtained, and only considers the top-ranked k candidates coming from G, as described in Eq. 2. The parameters θ and w are optimized using $\mathcal{D}$. Figure 2 illustrates the overall architecture of the model.

At inference time, given a specific user interaction history X_u, both, G and R, take it as an input and operate sequentially. We describe more details of the candidate generator and the re-ranker in Sects. 4.1 and 4.2.

4.1 The Candidate Generator

The candidate generator G can be an arbitrary recommender that takes a user session X_u as input and ranks the set Y_G. For training G we consider $Y_G = I$.

The selection of the algorithm for G depends mostly on the characteristics of the dataset $\mathcal{D}$ and the performance of the algorithm on generating high quality candidates.

4.2 The Re-ranker

The re-ranker R takes the same user click sequence X_u, as G does, but concentrates on ranking a smaller set of candidates C determined by the candidate generator it connects with.

We employ a variant of STAMP in R to include the candidate rank information in the model. Specifically, the encoder for the user click history E_{STAMP} is reused. E_{STAMP} is a simple element-wise multiplication between the history representation h_s and the anchor representation item h_t described in [11].

$$h_u = E_{STAMP}(X_u) = h_s \odot h_t \tag{3}$$

We are aware of other possible encoders that might be able to process more user features than the latest interactions, or represent the short and long click history of the user in a better way. Still, we chose the STAMP encoder E_{STAMP} for two reasons. On one hand, with it we benefit from architectural simplicity and promising performance in predicting user preferences. On the other hand, our main focus in this work is not about finding a better user encoder, but the scoring of the candidate items.

Before defining the score calculation, we introduce Candidate Rank Embeddings. Given a sequence X_u, we obtain a sorted list of candidates $C = [y_0, y_1, \cdots, y_{k-1}]$ of size k from G. The **candidate rank** r_i is an integer ranging from $[0, k)$ which denotes the position of the ith item in C. Each candidate rank is associated with a Candidate Rank Embedding. Therefore, CREs are positional embeddings shared among different candidate lists produced by G. The CREs for a re-ranker that takes k candidates into account can be represented as a matrix W_{CR} of shape $\mathbb{R}^{k \times d_{CRE}}$.

With the CRE defined the scores are specified in the following equation:

$$\begin{aligned} h_e &= MLP_1(h_u) = tanh(W_{e1}^\intercal tanh(W_{e2}^\intercal h_u + b_{e2}) + b_{e1}) \\ h_r &= MLP_2(h_u) = tanh(W_{r1}^\intercal tanh(W_{r2}^\intercal h_u + b_{r2}) + b_{r1}) \\ D_R(h_u, C) &= softmax(V_C^\intercal h_e + W_{CR}^\intercal h_r), \end{aligned} \tag{4}$$

where W_{e1}, b_{e1}, W_{e2}, b_{e2}, W_{r1}, b_{r1}, W_{r2}, b_{r2} are learnable weight matrices and bias terms of the feed-forward network layers, V_C is the item embeddings of the candidates and W_{CR} is the candidate rank embedding matrix. Note that W_{CR} is the same for all user sequences X_u because they depend only on the rank of the candidates. We initialize all embeddings randomly and train them together with the model.

We train R to predict the next click that is in C by using the cross-entropy loss.

The main difference between our solution and STAMP is the use of W_{CR} and the two non-linear projections MLP_1 and MLP_2. These projections approximate the click probability of the candidates and the rank preference of users. The first projection focuses on predicting the embedding of the target item, while the second one focuses on predicting the embedding of the position of the target item in the ranked candidate list.

The intuition behind learning the rank preference is that the information from the output of G can flow into the model, and a balance between the newly-learned item preference and the old rank can be obtained by summing up two user preferences.

Furthermore, since the ranking score comes from the dot product from the candidate rank embeddings and a projection from the user representation, this allows the model to learn personalized relationships between the user and the position of the target. For example, it gives the model the capability to recognize which users like to click the top positions or the items that frequently co-occur with the anchor

when G produces its candidates using co-occurrence information. Such behavior can be difficult to learn with a model that considers only user-item preferences.

In Sect. 5.7, we present an analysis of the importance of using the candidate rank information; we compare our two-stage approach against one without CREs.

So far, we have only tried using a one-to-one mapping between candidate ranks or positions and CREs. In applications with a large candidate set, having multiple ranks share one CRE could be beneficial because training signals can be shared among several unpopular positions.

5 Experiments and Analysis

5.1 Datasets

We evaluated the proposed method on three datasets representing two common use cases of session-based recommenders.

- The **Fashion-Similar** dataset is from Zalando, the biggest e-commerce fashion platform in Europe. We collected user interaction sequences from the Zalando Fashion Store over several days in early 2018. Each sequence contains two parts: the input sequence and the target item. The input sequence consists of the last l items a specific user interacted with. The last item in the input corresponds to a product detail page view (see Fig. 1). On the bottom part of that page is a carousel of similar items, showing alternative items that the user might also like. The target item is one of the recommendations the user clicked on. In the case of multiple clicks, we created multiple sequences with identical input but different targets.

  Sequences with target user clicks that happen on the last day are included in the test set. We use a time-based train-test split because we want to simulate employing the recommender in a real-world production environment.
- The **YooChoose 1/4** dataset was used in the RecSys'15 challenge. We followed the same pre-processing steps as proposed in [17] to speed up training. Specifically, we filtered the dataset by keeping only sessions with at least two interactions and items that appear in at least 5 different sessions. The filtered dataset is then divided into training and test sets by a time split. Note that we followed the suggestions proposed by Tan et al. to use the most recent 1/4 of the training data, for this arrangement yields faster training procedure and similar results compare to using the whole training set.
- The **Diginetica** dataset from CIKM Cup 2016 contains only transactional data [9]. We pre-processed this dataset similarly as we pre-processed the YooChoose Dataset with the exception of dividing the training dataset. Also, for this dataset we sorted the session events by time, as [19] suggests. The difference in training data preprocessing yields different experiment results compared to those reported by other studies [9, 11].

Table 1 Statistics of the datasets used

Dataset	Fashion-Similar	YooChoose 1/4	Diginetica
# train sequences	8,353,562	5,917,746	719,470
# test sequences	624,559	55,898	60,858
# unique items in training	650,228	30,232	43,097
# unique items in test	259,784	6,751	21,131
avg. len. train sequences	9.692	4.112	3.910
avg. len. test sequences	8.486	5.445	4.035

The Fashion-Similar dataset is used to evaluate how well a session-based recommender predicts the next similar item on the carousel that users would click. This specialized next-item prediction problem is especially of concern for industry applications, where the algorithms are designed to support the functionality of a product i.e. items similar to a certain base item, or items belonging to certain colors, brands, etc. To evaluate the performance of CREs in a more general setting aligned with the research community, we use the two general next-item prediction datasets YooChoose and Diginetica. The YooChoose and Diginetica datasets are publicly available, while the Fashion-Similar dataset is proprietary. The sessions from the public datasets are augmented by using all the prefixes of each sequence, as described in[17].

Table 1 shows the statistics of all three datasets.

5.2 Evaluation Metrics

We use the following metrics to evaluate the proposed method and the baselines:

- **Recall@K** measures the proportion of test cases where the target is among the top-k recommendations. For this study, we consider **Recall@5** and **Recall@20**. Particularly, we take Recall@5 into account because on many recommendation platforms users see only a few items. Therefore it is important to put the most relevant content at the top. Recall@20 is a widely used performance metric to compare between recommender systems from other studies.
- In contrast to Recall@K, **MRR@K** (mean reciprocal rank) measures how high the target items T from a dataset $\mathcal{D}$ are ranked at the top positions. This metric is defined as:

$$MRR@K = \frac{1}{|T|} \sum_{y_u^+ \in T} \begin{cases} \frac{1}{rank(y_u^+)}, & \text{if } rank(y_u^+) \leq k \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

We measure **MRR@5** and **MRR@20** for each model in the experiments.

5.3 Baselines

We use the following recommendation algorithms as baselines:

- **Item-Item Collaborative Filtering (I2I-CF)**: This method considers only the most similar items for the last-seen item x_{l-1} in the user interaction/click sequence. We use a variant of the cosine similarity function described in [1].

$$s_{y_u} = \frac{\Omega(y_u, x_{l-1})}{\Omega(y_u)^{\alpha} * \Omega(x_{l-1})^{1-\alpha}}, \tag{6}$$

  where α is a constant chosen between 0 and 1. Approaching 0, the similarity function becomes the conditional probability of y_u given x_{l-1}. When $\alpha = 0.5$, s_{y_u} is the cosine similarity between the user clicks received by y_u and x_{l-1}.

  Ω is a co-occurrence function in case of receiving two parameters and count in case of one.
- **Short-Term Attention Model (STAMP)** [11]: This approach utilizes a neural network to compute the similarity between a user and a candidate item with a tri-linear combination of three different pieces of information: (1) The summarized user history representation, calculated by a simple attention network with user history item embeddings and the anchor item embedding taken into account. (2) A non-linear transformation of the embedding of the anchor item x_{l-1}, and (3) a non-linear transformation of the candidate item embedding. Recommendations are made by ranking the scores of candidates.
- **Attention-based GRU (NARM)** [9]: Similar to STAMP, this model summarizes the user history by computing a user embedding and ranks all the items by computing the dot product of the user and item embeddings. The difference is that the user history encoder employs two recurrent neural networks (RNN) to generate the user Embedding. A simple GRU is used to generate an overview of the sequence and the other GRU with attention captures prominent local intentions of the user from the sequence. The output vectors of two RNNs are combined into the representation of a user by concatenation. The authors showed that the model is able to capture the local and global preference of the user.

5.4 Experimental Setup

Each of the baselines is used as a candidate generator in two sets of separate experiments. Before training the re-ranker, we first train the corresponding baseline and keep them fixed as candidate generators. Not all the training sequences are used in training, the re-ranker only considers those whose target item falls within the candidate set C from G. We randomly selected five percent of the training examples for validation. During training, the model performance is checked every 1000 steps and the best model is selected by their performance in terms of Recall@5. Adam

[7] is used to train R for 5 epochs with a learning rate of **0.001** and batch size of 512. We set the number of candidates k being returned by G to be **100**. The item embeddings and model weights of E_{STAMP} are initialized w.r.t. the best settings reported in [11].

The other hyper-parameters considered when training the Re-ranker as listed below. Note that in the case of multiple parameters, the ones that worked better for us are marked in bold:

- Embedding dimension for items: 100 [11].
- Candidate Rank Embedding dimension: 100.
- Candidate Rank Embedding initialization: Xavier [4].
- Weight initializer for the re-ranker decoder: Xavier.

The best models are selected by their performance in the validation set in terms of Recall@5.

For both tasks, we use RRCRE-X as an abbreviation for re-ranking the output of approach X with CREs, e.g. RRCRE-I2I-CF means re-ranking the candidates generated by I2I-CF with CREs.

5.5 Predicting Fashion-Similar Target Clicks

In this task, our goal is to predict the clicked products in the similar-item-recommendation carousel, given the latest l actions of a user.

Only the **Fashion-Similar** dataset was used because the other datasets are mainly used for general next click prediction tasks. Every record in this dataset consists of a click on the similar-item recommendation, which is the target, along with the latest 12 items the user browsed before interacting with the similar-item recommendation.

The training set contains 9 days of user sequences and the test set consists of sequences with target clicks happening on the last day. With this arrangement, the evaluation metrics measure how well the model performs on the next day assuming we retrain our algorithms daily. The results are shown in Table 2.

Table 2 Performance of the baselines and the proposed method on predicting similar items with 5 and 20 recommendations

	Recall@5	MRR@5	Recall@20	MRR@20
I2I-CF	**0.4086**	**0.2211**	**0.8106**	**0.2611**
STAMP	0.3918	0.2092	0.7244	0.2443
NARM	0.3816	0.2183	0.6989	0.2510
RRCRE-I2I-CF	**0.4692**	**0.2593**	**0.8381**	**0.2981**
RRCRE-STAMP	0.4290	0.2383	0.7416	0.2716
RRCRE-NARM	0.4241	0.2361	0.7184	0.2675

Table 3 Performance of the baselines and the proposed method on the task of predicting the next click with 5 recommendations

	YooChoose 1/4		Diginetica	
	Recall@5	MRR@5	Recall@5	MRR@5
I2I-CF	0.3006	0.1772	0.1490	0.0820
STAMP	**0.4594**	**0.2752**	**0.4093**	0.2486
NARM	0.4499	0.2660	0.4075	**0.2493**
RRCRE-I2I-CF	0.3635	0.2243	0.2255	0.1379
RRCRE-STAMP	**0.4773**	**0.2889**	**0.4253**	**0.2641**
RRCRE-NARM	0.4652	0.2831	0.4160	0.2561

Table 4 Performance of the baselines and the proposed method on the task of predicting the next click with 20 recommendations

	YooChoose 1/4		Diginetica	
	Recall@20	MRR@20	Recall@20	MRR@20
I2I-CF	0.5259	0.2001	0.3760	0.1211
STAMP	**0.6983**	0.2915	0.4834	0.1588
NARM	0.6973	**0.2921**	**0.5015**	**0.1599**
RRCRE-I2I-CF	0.5586	0.2446	0.3773	0.1220
RRCRE-STAMP	**0.7086**	**0.3133**	0.5046	**0.1677**
RRCRE-NARM	0.7029	0.3082	**0.5116**	0.1675

5.6 Predicting the Next Click

For this task, the objective is to predict the next user interaction given the past click history. We used the datasets **YooChoose 1/4** and **Diginetica**.

The results are listed in Tables 3 and 4. Note that the Recall and MRR scores we report for STAMP and NARM are based on the results of a corrected dataset preprocessing step, as described in [19]. Therefore there is a gap between the numbers reported in [11].

5.7 Offline Results and Analysis

Table 2 indicates that STAMP and NARM do not yield superior results in the task of predicting the similar item in the offline evaluation on the Fashion-Similar dataset. Among all the candidate generators, I2I-CF has the highest recall and MRR at both positions 5 and 20. With the proposed re-ranking method, the performance of NARM and STAMP improves and is comparable with I2I-CF. However, the best performing combination is RRCRE-I2I-CF, with which we are able to improve I2I-CF on all metrics. It is because the model is capable of utilizing the hidden information in the ranking of the baseline together with the session

Table 5 Recall@100 of I2I-CF for different datasets

Dataset	Recall@100
Fashion-Similar	0.99
YooChoose 1/4	0.79
Diginetica	0.55

information captured by the attention network from E_{STAMP}. We performed an online test to confirm the improvement of RRCRE-I2I-CF over I2I-CF, described in Sect. 5.8. It seems to be counter-intuitive that simple approaches like item-item collaborative filtering outperform neural-network based session recommenders, however, experiments performed on another industry dataset [16] showed similar trends in that GRU4Rec was outperformed by Item-kNN. One possible reason for the superior performance of RRCRE-I2I-CF is that the Fashion-Similar dataset records mainly the users' response to a non-personalized algorithm similar to item-item collaborative filtering, and thus grants more advantage to similar algorithms that captures global dynamics between the relationship of two items.

For the next-item prediction task, as shown in Tables 3 and 4, STAMP and NARM perform significantly better than I2I-CF in the next-click prediction task. We also applied our method to use STAMP and NARM as candidate generator in this task, the evaluation result shows that it is able to slightly improve the Recall@20 and MRR@20 of STAMP and NARM on both YooChoose 1/4 and Diginetica. Re-ranking the output of I2I-CF is, in contrast, not as effective as session-based candidate generators.

RRCRE-I2I-CF has almost no improvements over I2I-CF on Diginetica for both top-5 and top-20 recommendations. The inferior performance of RRCRE-I2I-CF on Diginetica can be explained by Table 5. As depicted, the recall of the first 100 recommended items from I2I-CF only covers 55% of the sequences, and it becomes an upper bound for any re-ranking algorithms applied on top of it.

To understand the improvement obtained by applying the Candidate Rank Embeddings, we compared the model performance between the proposed approach (denoted with a prefix *RRCRE*) and a variant from which we remove the CREs (denoted as a prefix *RR*). Specifically, the final output scores of candidates from the variant become $D_R(h_u, C) = softmax(V_C^\intercal h_e)$. The result is illustrated in Fig. 3. For the Fashion-similar dataset, we can observe that simply re-ranking the most relevant candidates from I2I-CF with E_{STAMP} (RR-I2I-CF) does not lead to better results compared to the proposed solution (RRCRE-I2I-CF). On the other hand, when training with CREs we obtain a better result listed in Table 2. We also compare RR-STAMP and RRCRE-STAMP in one of the next click prediction datasets. It turns out that RRCRE-STAMP outperforms the baseline from epoch 1, while RR-STAMP requires more iterations. This is because RR-STAMP has to learn the next clicked target from randomly-initialized model parameters without the rank information from STAMP being present.

It is important to note that the main difference between STAMP and NARM is how they encode the user history. In contrast, the proposed approach RRCRE-

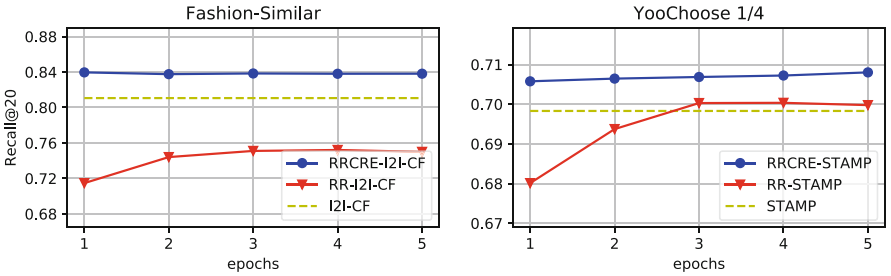


Fig. 3 Our approach with and without candidate rank information

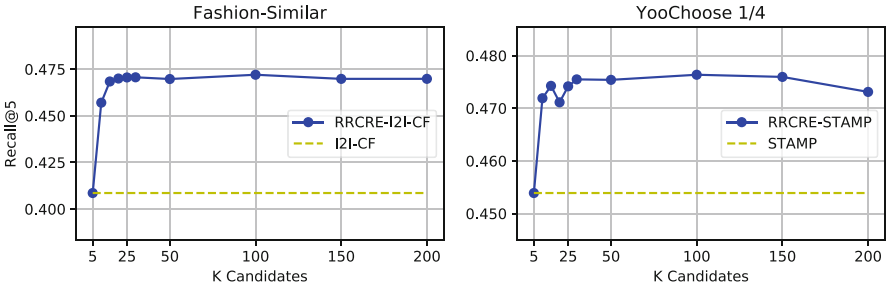


Fig. 4 Recall@5 of the bests approaches on each dataset given different number of candidates to re-rank

X focuses on how to incorporate the information coming from a baseline ranking into the model regardless of the approach used to rank the items. Therefore, testing different user encoders is out of the scope of this study.

Additionally, we illustrate the behavior of the proposed method with respect to the number of candidates to re-rank. In Fig. 4 we can observe improvements in recall@20 even with a small k. In addition, we found the recall plateaus or decreases when k exceeds a certain threshold. One possible reason could be that the candidates associated with the low ranks rarely appear as targets in the dataset. As a result, the CREs for these ranks could not have been well-trained and could have captured misleading information. However, since using a small k simplifies the effort of training and serving in production, we have not investigated this further.

5.8 Online Experiment at Zalando

In the previous section, we observed an improvement in the offline metrics with respect to I2I-CF. To further assess the actual effect of this approach on Zalando products, we conducted an online test that compares I2I-CF and RRCRE-I2I-CF.

RRCRE-I2I-CF was served using CPU machines running TensorFlow-Serving. For this algorithm, the recommendations are calculated in real-time. Specifically,

the user's interactions within a session are updated and fed into the algorithm when they proceed with their browsing. On the contrary, I2I-CF was served by using a static table stored in memory and only the base item is used as key to retrieve the most similar items.

I2I-CF provides static non-personalized similar recommendations and RRCRE-I2I-CF provides the adjusted similar item list depending on the user's previous l actions.

We updated the models every day to adapt to the latest user behavior, and we compared their performance in major European markets for several days.

To ensure that the recommendations satisfy the similarity constraint of the product, some filters based on category trees were applied to the output of both methods.

The results showed relative improvements in engagement based on a significant $+\mathbf{2.84}\%$ ($p - value \leq 0.05$) increase in click-through rate. It proves that there is a positive effect of using the session of users to generate a personalized ranking of similar items and supports the findings of our offline experiments.

6 Conclusion and Future Work

We explored the possibility of improving similar item recommendation in the fashion domain with a two-staged re-ranking approach that is able to benefit from the candidate rank information, the session of the user and a small set of candidates.

With this approach, we improved Recall@20 and MRR@20 of item-based collaborative filtering on the Fashion-Similar dataset, and the success in the offline evaluation was confirmed by an online test.

We also confirmed that the proposed approach improves the performance of two advanced session-based recommendation algorithms, STAMP and NARM on the next click prediction datasets YooChoose 1/4 and Diginetica. Despite the success in the offline evaluation, further experiments are needed to confirm the impact of the proposed method in the context of session-based recommendation.

Acknowledgments The authors are immensely grateful to Alan Akbik, Andrea Briceno, Humberto Corona, Antonino Freno, Zeno Gantner, Francis Gonzalez, Romain Guigoures, Sebastian Heinz, Bowen Li, Max Moeller, Roberto Roverso, Reza Shirvany, Julie Sanchez, Hao Su, Lina Weichbrodt, and Nana Yamazaki for their support, revisions, suggestions, ideas and comments that greatly helped to improve the quality of this work.

References

1. Aiolli F (2013) A preliminary study on a recommender system for the Million Songs Dataset Challenge. In: CEUR Workshop Proceedings. http://ceur-ws.org/Vol-964/paper12.pdf
2. Covington P, Adams J, Sargin E (2016) Deep neural networks for YouTube recommendations. In: Proceedings of the 10th ACM Conference on Recommender Systems, RecSys'16. ACM, New York, pp 191–198. https://doi.org/10.1145/2959100.2959190
3. Devooght R, Bersini H (2016) Collaborative filtering with recurrent neural networks. CoRR abs/1608.07400. http://arxiv.org/abs/1608.07400
4. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: Teh YW, Titterington M (eds) Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol 9, 13–15 May 2010. PMLR, Chia Laguna Resort, Sardinia, pp 249–256. http://proceedings.mlr.press/v9/glorot10a.html
5. Hidasi B, Karatzoglou A (2018) Recurrent neural networks with top-k gains for session-based recommendations. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM'18. ACM, New York, pp 843–852. https://doi.org/10.1145/3269206.3271761
6. Hidasi B, Karatzoglou A, Baltrunas L, Tikk D (2015) Session-based recommendations with recurrent neural networks. CoRR abs/1511.06939. http://arxiv.org/abs/1511.06939
7. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. CoRR abs/1412.6980. http://arxiv.org/abs/1412.6980
8. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. Computer 42(8):30–37. https://doi.org/10.1109/MC.2009.263
9. Li J, Ren P, Chen Z, Ren Z, Lian T, Ma J (2017) Neural attentive session-based recommendation. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM'17. ACM, New York, pp 1419–1428. https://doi.org/10.1145/3132847.3132926
10. Linden G, Smith B, York J (2003) Amazon.com recommendations item-to-item collaborative filtering. IEEE Internet Comput 1(February):76–80. http://www.academia.edu/download/33248546/Amazon-Recommendations.pdf
11. Liu Q, Zeng Y, Mokhosi R, Zhang H (2018) STAMP: short-term attention/memory priority model for session-based recommendation. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'18. ACM, New York, pp 1831–1839. https://doi.org/10.1145/3219819.3219950
12. Ludewig M, Jannach D (2018) Evaluation of session-based recommendation algorithms. CoRR abs/1803.09587. http://arxiv.org/abs/1803.09587
13. Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L (2009) BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI'09. AUAI Press, Arlington, pp 452–461. http://dl.acm.org/citation.cfm?id=1795114.1795167
14. Rendle S, Freudenthaler C, Schmidt-Thieme L (2010) Factorizing personalized Markov chains for next-basket recommendation. In: Proceedings of the 19th International Conference on World Wide Web, WWW'10. ACM, New York, pp 811–820. https://doi.org/10.1145/1772690.1772773
15. Rubtsov V, Kamenshchikov M, Valyaev I, Leksin V, Ignatov DI (2018) A hybrid two-stage recommender system for automatic playlist continuation. In: Proceedings of the ACM Recommender Systems Challenge 2018. RecSys Challenge'18. ACM, New York, pp 16:1–16:4. https://doi.org/10.1145/3267471.3267488
16. de Souza Pereira Moreira G, Ferreira F, da Cunha AM (2018) News session-based recommendations using deep neural networks. In: Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems, DLRS'18. ACM, New York, pp 15–23. https://doi.org/10.1145/3270323.3270328

17. Tan YK, Xu X, Liu Y (2016) Improved recurrent neural networks for session-based recommendations. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, DLRS'16. ACM, New York, pp 17–22. https://doi.org/10.1145/2988450.2988452
18. Wu JC, Rodríguez JAS, Pampín HJC (2019) Session-based complementary fashion recommendations. In: Workshop on Recommender Systems in Fashion
19. Wu S, Tang Y, Zhu Y, Wang L, Xie X, Tan T (2019) Session-based recommendation with graph neural networks. In: Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, vol 33, pp 346–353. http://arxiv.org/abs/1811.00855
20. Yu F, Liu Q, Wu S, Wang L, Tan T (2016) A dynamic recurrent model for next basket recommendation. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'16. ACM, New York, pp 729–732. https://doi.org/10.1145/2911451.2914683

Part III
Outfit Recommendations

Attention-Based Fusion for Outfit Recommendation

Katrien Laenen and Marie-Francine Moens

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

Keywords Outfit recommendation · Item understanding · Attention · Attention-based fusion

1 Introduction

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

In this work, we tackle the problem of outfit recommendation. The goal of this task is to compose a fashionable outfit either from scratch or starting from an incomplete set of items. Outfit recommendation has two main challenges. The

K. Laenen (✉) · M.-F. Moens
KU Leuven, Leuven, Belgium
e-mail: katrien.laenen@kuleuven.be; sien.moens@kuleuven.be

N. Dokoohaki (ed.), *Fashion Recommender Systems*, Lecture Notes in Social Networks, https://doi.org/10.1007/978-3-030-55218-3_4

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

first is *item understanding*. Fine details in the garments can be important for making combinations. For example, the items in Fig. 1 match nicely because of the red heels of the sandals, the red flowers on the dress and the red pendants of the bracelet. These fine-grained product details should be captured in the item representations. Moreover, usually there is also a short text description associated with the product image. These descriptions point out certain product features and contain information which is useful for making combinations as well. Hence, there is a need to effectively integrate the visual and textual item information into the item representations. The second challenge in outfit recommendation is *item matching*. Item compatibility is a complex relation. For instance, assume items A and B are both compatible with item C. In that case items A and B can be, but are not necessarily, visually similar. Moreover, items A and B can be, but are not necessarily, also compatible with each other. Furthermore, different product features can play a role in determining compatibility depending on the types of items being matched, as illustrated in [14].

This work will focus on *item understanding*. Our outfit recommender system operates on region-level and word-level representations to bring product features which are important to make item combinations to the forefront as needed. The contributions of our work are threefold. Firstly, our approach works on a finer level of image regions and words. In contrast, previous approaches to outfit recommendation work on a more coarse level of full images and sentences. Secondly, we explore

different attention mechanisms and propose an attention-based fusion method which fuses the visual and textual information to capture the most relevant product features into the item representations. Attention mechanisms have not yet been explored in outfit recommender systems to improve item understanding. Thirdly, we improve state-of-the-art outfit recommendation results on three benchmark datasets.

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

2 Related Work

The task of outfit fashionability prediction requires to uncover which items go well together based on item style, color and shape. This can be learned from visual data, language data or a combination of the two. Currently, two approaches are common to tackle outfit fashionability prediction. The first one is to infer a feature space where visually compatible clothing items are close together. Veit et al. [16] use a Siamese convolutional neural network (CNN) architecture to infer a compatibility space of clothing items. Instead of only one feature space, multiple feature spaces can also be learned to focus on certain compatibility relationships. He et al. [5] propose to learn a compatibility space for different types of relatedness (e.g., color, texture, brand) and weight these spaces according to their relevance for a particular pair of items. Vasileva et al. [14] infer a compatibility space for each pair of item types (i.e., tops and bottoms, tops and handbags) and demonstrate that the embeddings specialize to features that dominate the compatibility relationship for that pair of types. Moreover, their approach also uses the textual descriptions of items to further improve the results. The second common approach to outfit fashionability prediction is to obtain outfit representations and to train a fashionability predictor on these outfit representations. In [13] a conditional random field scores the fashionability of a picture of a person's outfit based on a bag-of-words representation of the outfit and visual features of both the scenery and person. Their method also provides feedback on how to improve the fashionability score. In [7] neural networks are used to acquire multimodal representations of items based on the item image, category and title, to pool these into one outfit representation and to score the outfit's fashionability. Other approaches to outfit fashionability prediction also exist. In [3] an outfit is treated as an ordered sequence and a bidirectional long short-term memory (LSTM) model is used to learn the compatibility relationships among the fashion items. In [6] the visual compatibility of clothing items is captured with a correlated topic model to automatically create capsule wardrobes. Lin et al. [9] build an end-to-end learning framework that improves item recommendation with co-supervision of item generation. Given an image of a top and a description of the requested bottom (or vice versa) their model composes outfits consisting of

one top piece and one bottom piece. In [2] an encoder-decoder framework based on the Transformer architecture is proposed which generates personalized outfits by taking into account both item compatibility and user behaviour. Their approach uses self-attention to learn the compatibilities between each item and all other items within an outfit.

None of the above approaches use attention on the regions of a product image and the words in a product description to improve item understanding. In contrast, we use attention to bring fine-grained product features in the image and description to the forefront in the item representation as needed for the outfit recommendation task.

3 Methodology

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

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

3.1 Common Space Fusion

The baseline model is the method of [14]. The model receives two triplets as input: a triplet of image embeddings $(\boldsymbol{x}_{(u)}, \boldsymbol{x}^{+}_{(v)}, \boldsymbol{x}^{-}_{(v)})$ of dimension d_i and a triplet of corresponding description embeddings $(\boldsymbol{t}_{(u)}, \boldsymbol{t}^{+}_{(v)}, \boldsymbol{t}^{-}_{(v)})$ of dimension d_t. How these image and description embeddings are obtained is detailed in Sect. 4.5. Embeddings $\boldsymbol{x}_{(u)}$ and $\boldsymbol{x}^{+}_{(v)}$ represent images of items of respectively type u and type v which are compatible. Compatible means that the items represented by $\boldsymbol{x}_{(u)}$ and $\boldsymbol{x}^{+}_{(v)}$ appear together in some outfit. Meanwhile $\boldsymbol{x}^{-}_{(v)}$ represents a randomly sampled image of an item of the same type as $\boldsymbol{x}^{+}_{(v)}$ that has not been seen in an outfit with $\boldsymbol{x}_{(u)}$ and is therefore considered to be incompatible with $\boldsymbol{x}_{(u)}$.

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

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

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

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

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

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

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

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

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

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

with $W_i \in \mathbb{R}^{d_g \times d_i}$ the image projection to $\mathcal{S}$, $W_c^{(u,v)} \in \mathbb{R}^{d_c \times d_g}$ the projection associated with $\mathcal{C}_{(u,v)}$, and ℓ the standard triplet loss of Eq. 3. As such, we use cosine similarity f to measure both similarity in $\mathcal{S}$ and compatibility in $\mathcal{C}_{(u,v)}$ since it is a very popular metric in image and text analysis.

Finally, the complete training loss is:

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

with λ_1, λ_2 and λ_3 scalar parameters.

3.2 *Attention-Based Fusion*

The downside of the baseline model is that the item representations are quite coarse and the interaction between the visual and textual modality is quite limited.

Instead, we would like to highlight certain regions of an image or words in a description which correspond to important product features for making fashionable item combinations, and integrate these into a multimodal item representation. We propose to use attention, which is a mechanism used in neural networks to focus on certain parts of the input (i.e., regions in an image or words in a text) in order to bring fine-grained product features to the forefront as needed. Attention was originally introduced for neural machine translation [1] and has not yet been explored in outfit recommendation to improve item understanding. Further, attention has the advantage that it increases interpretability and leads to explainability through visualisation of the attention weights in the image.

We obtain our proposed attention-based fusion model by making a few adjustments to the baseline model. Firstly, the first input to the attention-based fusion model is a triplet of region-level image features $(\boldsymbol{x}_{\mathbf{1:N}(u)}, \boldsymbol{x}^{+}_{\mathbf{1:N}(v)}, \boldsymbol{x}^{-}_{\mathbf{1:N}(v)})$ of dimension d_i, where N denotes the number of regions. Depending on the attention mechanism used, the other input is either a triplet of description-level features $(\boldsymbol{t}_{(u)}, \boldsymbol{t}^{+}_{(v)}, \boldsymbol{t}^{-}_{(v)})$ as before or a triplet of word-level features $(\boldsymbol{t}_{\mathbf{1:M}(u)}, \boldsymbol{t}^{+}_{\mathbf{1:M}(v)}, \boldsymbol{t}^{-}_{\mathbf{1:M}(v)})$ of dimension d_t, where M denotes the number of words. Details on how these features are obtained can be found in Sect. 4.5. Since $\mathcal{L}_{vsim}$ and $\mathcal{L}_{vse}$ are formulated at the level of full images, we obtain image-level features by simply taking the average of the region-level features, i.e., $\boldsymbol{x}_{(u)} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_{\boldsymbol{i}(u)}$. In the same way we obtain description-level features from word-level features for $\mathcal{L}_{tsim}$.

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

We experiment with different attention mechanisms to fuse the visual and textual information. These are described in Sects. 3.2.1, 3.2.2, 3.2.3, and 3.2.4. Furthermore, we also tried out some other attention mechanisms [10–12] which did not improve performance.

3.2.1 Visual Dot Product Attention

Given region embeddings $X \in \mathbb{R}^{N \times d_g}$ and description embedding $\boldsymbol{t} \in \mathbb{R}^{d_g}$, visual dot product attention produces attention weights a_i based on the dot product of the embeddings of the description and each region:

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

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

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

with $\boldsymbol{x_i}$ the i'th row of X. Finally, multimodal item representation $\boldsymbol{m}$ is obtained by concatenating the visual context vector $\boldsymbol{c}$ and description embedding $\boldsymbol{t}$, i.e., $\boldsymbol{m} = [\boldsymbol{c}; \boldsymbol{t}]$ with ; the concatenation operator.

3.2.2 Stacked Visual Attention

Given region embeddings $X \in \mathbb{R}^{N \times d_g}$ and description embedding $\boldsymbol{t} \in \mathbb{R}^{d_g}$, stacked visual attention [17] produces a multimodal context vector in multiple attention hops, each extracting more fine-grained visual information. In the r'th attention hop, the attention weights and context vector are calculated as:

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

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

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

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

This process is repeated R times, with R the number of attention hops. Afterwards, multimodal item representation $\boldsymbol{m}$ is obtained by concatenating the final guidance vector $\boldsymbol{g}^{(R)}$ with description embedding $\boldsymbol{t}$, i.e., $\boldsymbol{m} = [\boldsymbol{g}^{(R)}; \boldsymbol{t}]$.

3.2.3 Visual L-Scaled Dot Product Attention

Visual L-scaled dot product attention [15] is a self-attention mechanism where the query Q, keys K and values V are the region embeddings $X \in \mathbb{R}^{N \times d_g}$. L-scaled dot product attention computes the scaled dot product of the query and keys in L different h-dimensional spaces. This way, each space can focus on particular data characteristics. Hence in space l, the attention weights $\mathcal{A}^l$ and weighted values C^l are calculated as:

$$\mathcal{A}^l = \text{softmax}(\frac{(QW^l)(KW^l)^T}{\sqrt{h}}) \tag{13}$$

$$C^l = \mathcal{A}^l (VW^l) \tag{14}$$

with $W^l \in \mathbb{R}^{d_g \times h}$. Finally, all C^l are concatenated across their columns and transformed back to dimension d_g:

$$C = [C^1; C^2; \ldots; C^L]W_f \tag{15}$$

with $W_f \in \mathbb{R}^{Lh \times d_g}$. Next, we add a residual connection to C and apply layer normalization as suggested in [8]:

$$O = \text{LayerNorm}(C + V) \tag{16}$$

Finally, we reduce O to a vector through average pooling:

$$\boldsymbol{o}_f = \sum_{i=1}^{N} \boldsymbol{o}_i \tag{17}$$

with $\boldsymbol{o}_i$ the i'th row of O. The resulting feature $\boldsymbol{o}_f$ is a fine-grained visual representation. Finally, $\boldsymbol{o}_f$ is concatenated with description embedding $\boldsymbol{t}$ to obtain multimodal item representation $\boldsymbol{m}$, i.e., $\boldsymbol{m} = [\boldsymbol{o}_f; \boldsymbol{t}]$.

3.2.4 Co-attention

The co-attention mechanism of [18] attends to both the region embeddings $X \in \mathbb{R}^{N \times d_g}$ and word embeddings $Y \in \mathbb{R}^{M \times d_g}$ as follows.

First, the description words are attended independent of the image regions. The assumption here is that the most relevant words of the description can be inferred independent of the image content, i.e., words referring to color, shape, style and brand can be considered relevant independent of whether they are displayed in the image or not. Given word embeddings Y, the textual attention weights $\boldsymbol{a}^t$ and textual context vector $\boldsymbol{c}^t$ are obtained as:

$$\boldsymbol{a}^t = \underset{in=d_g,out=1,k=1}{\text{Convolution1D}_{t,2}}(\text{ReLU}(\underset{in=d_g,out=d_g,k=1}{\text{Convolution1D}_{t,1}}(Y))) \tag{18}$$

$$\boldsymbol{c}^t = \boldsymbol{\alpha}^t Y, \text{ with } \boldsymbol{\alpha}^t = \text{softmax}(\boldsymbol{a}^t) \tag{19}$$

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

Next, the image regions are attended in R attention hops. In the r'th attention hop, the textual context vector $\boldsymbol{c}^t$ is merged with each of the region embeddings in X using multimodal factorized bilinear pooling (MFB). MFB consists of an *expand stage* where the unimodal representations are projected to a higher dimensional space of dimension $p2d_g$ (with p a hyperparameter) and then merged with elementwise multiplication followed by a *squeeze stage* where the merged feature is transformed back to a lower dimension $2d_g$. For a detailed explanation of MFB the

reader is referred to [18]. The MFB operation results in a multimodal feature matrix $M \in \mathbb{R}^{N \times 2d_g}$. Then, the visual attention weights $\boldsymbol{a}^{v,(r)}$ and context vector $\boldsymbol{c}^{v,(r)}$ are calculated based on this merged multimodal feature matrix M:

$$\boldsymbol{a}^{v,(r)} = \underset{in=d_g,out=1,k=1}{\text{Convolution1D}^{(r)}_{v,2}}(\text{ReLU}(\underset{in=2d_g,out=d_g,k=1}{\text{Convolution1D}^{(r)}_{v,1}}(M))) \tag{20}$$

$$\boldsymbol{c}^{v,(r)} = \boldsymbol{\alpha}^{v,(r)} M, \text{ with } \boldsymbol{\alpha}^{v,(r)} = \text{softmax}(\boldsymbol{a}^{v,(r)}) \tag{21}$$

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

$$\boldsymbol{c}^v = [\boldsymbol{c}^{v,(1)}; \boldsymbol{c}^{v,(2)}; \ldots; \boldsymbol{c}^{v,(R)}] W_f \tag{22}$$

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

4 Experimental Setup

4.1 Experiments and Evaluation

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

FC questions and FITB questions that consist of images without a description are discarded to keep evaluation fair for all models. Also note that if a pair of items has a type combination that was never seen during training, the model has not learned a type-specific compatibility space for that pair. Such pairs are ignored during evaluation. Hence, we also use the training set to determine which pairs of types do not effect outfit fashionability.[1]

[1] Alternatively, we could compare such item pairs in the semantic space instead. This has a negligible effect on experimental results.

4.2 Baselines

We compare our attention-based fusion models with the following baselines:

- **Baseline common space fusion** The baseline common space fusion model as described in Sect. 3.1.
- **Standard common space fusion** The common space fusion model as described in Sect. 3.1, with the adjustment that not the triplet of image embeddings $(\boldsymbol{x}_{(u)}, \boldsymbol{x}^{+}_{(v)}, \boldsymbol{x}^{-}_{(v)})$ but a triplet of multimodal item embeddings $(\boldsymbol{m}_{(u)}, \boldsymbol{m}^{+}_{(v)}, \boldsymbol{m}^{-}_{(v)})$ is projected to the type-specific compatibility spaces (Eq. 6). These multimodal item embeddings are obtained by concatenating the image and description embeddings in the semantic space, i.e., $\boldsymbol{m}_{(u)} = [W_i \boldsymbol{x}_{(u)};\ W_s \boldsymbol{t}_{(u)}]$. We use this model to be able to distinguish what is the effect of respectively including text in the compatibility spaces and of using attention.
- **Image only** This model only uses the image embeddings for outfit recommendation. It is similar to the common space fusion model in Sect. 3.1, except that it does not enforce the visual-semantic loss nor the textual similarity loss in the semantic space.
- **Text only** Analogous to the image only model, this model only uses the description embeddings for outfit recommendation.

4.3 Datasets

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

4.3.1 Polyvore68K

The Polyvore68K dataset[2] [14] originates from Polyvore. Two different train-test splits are defined for the dataset. Polyvore68K-ND contains 53,306 outfits for training, 10,000 for testing, and 5,000 for validation. It consists of 365,054 items, some of which occur both in the training and test set. However, no outfit appearing in one of the three sets is seen in the other two. The other split, Polyvore68K-D, contains 32,140 outfits, of which 16,995 are used for training, 15,145 for testing and 3,000 for validation. It has 175,485 items in total, where no item seen during training appears in the validation or test set. Both splits have their own FC questions and FITB questions.

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

[2] https://github.com/mvasil/fashion-compatibility

4.3.2 Polyvore21K

Another dataset collected from Polyvore is the Polyvore21K dataset[3] [3]. It contains items of 380 different item types, however not all are fashion related, e.g., furniture, toys, skincare, food and drinks, etc. We delete all items with types unrelated to clothing, clothing accessories, shoes and bags. The remaining 180 types are all fashion related, but some of them are very-fine grained. We make the item types more coarse to avoid an abundance of type-specific compatibility spaces, i.e., more than 5,000, which is unfeasible. The remaining 37 types can be found in Table 2 in Appendix A. Eventually, this leaves 16,919 outfits for training, 1,305 for validation and 2,701 for testing. There are no overlapping items between the three sets. Each item has an associated image and description.

During evaluation we use the FC questions and FITB questions supplied by [14] for the Polyvore21K dataset, after removal of fashion unrelated items.

4.4 Comparison with Other Works

This work uses a slightly different setup than the work of [14] and therefore our results are not exactly comparable with theirs. Firstly, we do not evaluate our models on the same set of FC and FITB questions. This is because we discard questions consisting of images without a description (Sect. 4.1). Secondly, the item types used for the Polyvore21K dataset are different. It is unclear from [14] how they obtain and use the item types of the Polyvore21K dataset, as these have only been made public recently. In this work, we used the publicly available item types after cleaning as detailed in Sect. 4.3.2. Furthermore, we train our own description embeddings (Sect. 4.5) and do not perform metric learning as in [14] as this degraded our results.

4.5 Training Details

All images are represented with the ResNet18 architecture [4] pretrained on ImageNet. More precisely, as in [14] we take the output of the $7 \times 7 \times 256$ *res4b_relu* layer. Since we take the output of this layer we obtain 49 image regions, each with a dimension d_i of 256, which form a grid segmentation across the image. For the models working with full images instead, we use an additional average pooling layer to obtain one image-level representation, also with a dimension d_i equal to 256. The text descriptions are represented with a bidirectional LSTM of which the forward and backward hidden state at timestep M are concatenated, with M the number of words in the descriptions. For models operating on the level of words instead

[3] https://github.com/xthan/polyvore-dataset

of full descriptions, we concatenate the forward and backward hidden state of the bidirectional LSTM at each timestep j to obtain the representation for the j'th word. In both cases the forward and backward hidden state have dimension 256, resulting in a dimension d_t of 512. The parameters of the ResNet18 architecture and the bidirectional LSTM are finetuned on our dataset during training. Dimensions d_g, d_c and h are equal to 512. For the attention mechanisms the number of attention hops R equal to 2, a hyperparameter p for MFB of 2, and L for L-scaled dot product attention equal to 1 were found to work well based on the validation set.

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

5 Results

Table 1 shows the results of the discussed models on the Polyvore68K dataset versions and the Polyvore21K dataset. We make the following observations.

Firstly, the image only model outperforms the text only model. Hence as expected, a product's image expresses more relevant product features for outfit recommendation than its description. However, on the FC as well as the FITB task

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

	Polyvore68K-ND		Polyvore68K-D		Polyvore21K	
	FC	FITB	FC	FITB	FC	FITB
No fusion						
Image only	85.83	56.76	85.02	56.68	86.83	59.06
Text only	74.53	43.01	70.88	41.49	74.39	46.54
Common space fusion						
Baseline [14]	85.62	56.55	85.07	56.91	86.28	58.35
Standard	**89.74**	61.68	87.01	60.33	88.30	62.35
Attention-based fusion						
Visual dot product attention	89.43	61.55	86.85	60.12	88.59	63.11
Stacked visual attention	89.68	**61.92**	87.25	60.48	88.89	62.52
Visual L-scaled dot product attention	89.06	61.14	**87.93**	**61.46**	**88.97**	**63.55**
Co-attention	89.58	61.20	86.25	59.00	85.04	58.20

the text only model does far better than random guessing. Clearly, product descriptions contain valuable information for outfit recommendation as well. Further, comparison of the image only and baseline common space fusion model shows little to no effect of aligning images and their descriptions in the semantic space through the visual-semantic loss. This observation together with the previous one suggests that the images and descriptions contain relevant, complementary information for outfit recommendation.

The attention-based fusion models outperform baseline common space fusion on all three datasets for both the FC and FITB tasks. Comparison of baseline and standard common space fusion shows the performance increase due to the use of multimodal representations instead of visual representations in the type-specific compatibility spaces. Hence, using the text in an effective way is responsible for a large portion of the performance increase. Comparison of the attention-based fusion models with standard common space fusion shows the performance increase due to the attention. It seems that attention is responsible for only a small portion of the total performance increase. On the Polyvore68K-ND dataset the fusion method based on stacked visual attention outperforms standard common space fusion on the FITB task only. On the Polyvore68K-D and the Polyvore21K datasets the best results for both tasks are achieved with the fusion method based on visual L-scaled dot product attention. Further, the attention contributes more to performance on the FITB task than on the FC task. The FITB task, i.e., outfit completion, is more subtle and difficult than the FC task, i.e., distinguishing between randomly generated and human-generated outfits. For the latter, there will be more and clearer visual and/or textual clues that signal an outfit is randomly composed. However for the former, finding the best matching item among four candidates requires detailed information of each candidate's product features which is obtained through the attention.

Figures 2 and 3 shows some FITB questions and answers generated by the baseline common space fusion model and our fusion model based on stacked visual attention for the Polyvore68K-ND dataset. For stacked visual attention-based fusion the attention maps of both attention hops are shown to provide insight in which regions are considered important. For each of these FITB questions, the ground truth item needs to be selected because of some small details in other items of the outfit which are picked up by our model but not by the baseline model. More precisely, in the first example the light blue handbag matches especially well with the light blue buckle of the pump. In addition the light blue handbag has the same brand as the items in the outfit as can be inferred from the product descriptions. The attention maps of the pump show that the blue buckle is indeed noticed. The buckle is probably attended since it is mentioned in the description. The attention maps for both hops look very similar. In the second example, the green belt matches nicely with the green accents on the clutch and slippers. It also provides a nice contrast with the white clothes, unlike the white belt selected by the baseline model. Looking at the attention maps of the clutch and slippers, we see a focus on the relevant green accents of the clutch and the right slipper. Further, there is also a clear focus on the heels of the slippers which might enable to capture the heel height of the slippers. This is an important product feature which is not mentioned in the slipper's

Fig. 2 Examples: Part 1. Fill-in-the-blank questions on the Polyvore68K-ND dataset and answers generated by the baseline common space fusion model and our attention-based fusion model based on stacked visual attention. For stacked visual attention-based fusion the attention maps of both attention hops are shown for some items. The attention maps show a focus on both shared and complementary product features

Fig. 3 Examples: Part 2. Fill-in-the-blank questions on the Polyvore68K-ND dataset and answers generated by the baseline common space fusion model and our attention-based fusion model based on stacked visual attention. For stacked visual attention-based fusion the attention maps of both attention hops are shown for some items. The attention maps show a focus on both shared and complementary product features

description and could evidence the ability of stacked visual attention-based fusion to focus on complementary product features. In the third example, the striped pattern of the ground truth handbag returns in the slippers and the yellow of the flower on the handbag returns in the sunglasses. The attention maps of the slippers do not show a focus on the striped pattern but on the heels, which again could be to capture the heel height. The attention maps of the sunglasses show a clear focus on the glasses and frame and the attention maps of the handbag show a focus on the striped pattern and flower. Hence, from the attention maps we can interpret that the ground truth handbag is probably selected because it matches well with the sunglasses rather than the slippers. In the last example, the print of the ground truth handbag returns in the dress. The attention maps of the handbag and dress also seem to show a focus on their prints. Further, we somewhat observe a focus on the dress' sleeve length which is a relevant product feature not expressed in the description. Hence, both quantitative and qualitative results demonstrate that highlighting certain product features in the item representations for making outfit combinations is meaningful and can be achieved with attention. We expect that results could be further improved with novel attention mechanisms specifically designed to focus on both shared and complementary product features and to optimally integrate them.

6 Conclusion

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

As future work and to further improve the results, we would like to investigate neural architectures that still better recognise fine-grained fashion attributes in images, to benefit more from the attention-based fusion. Furthermore, we would like to design novel fusion mechanisms which better integrate fine-grained visual and textual attributes given also their complementary nature in the context of outfit recommendation. Finally, while item understanding is a key initial step in building a high-performance outfit recommender system, in the future we would also like to improve our system by including personalization as well.

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

Appendix

A Dataset Item Types

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

Table 2 Item types kept in the Polyvore68K and Polyvore21K datasets

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

References

1. Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate. CoRR abs/1409.0473. http://arxiv.org/abs/1409.0473
2. Chen W, Huang P, Xu J, Guo X, Guo C, Sun F, Li C, Pfadler A, Zhao H, Zhao B (2019) POG: personalized outfit generation for fashion recommendation at Alibaba iFashion. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, pp 2662–2670
3. Han X, Wu Z, Jiang YG, Davis LS (2017) Learning fashion compatibility with bidirectional lstms. In: ACM International Conference on Multimedia (ACM-MM)
4. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 770–778
5. He R, Packer C, McAuley J (2016) Learning compatibility across categories for heterogeneous item recommendation. In: IEEE International Conference on Data Mining (ICDM)
6. Hsiao W, Grauman K (2018) Creating capsule wardrobes from fashion images. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 7161–7170
7. Li Y, Cao L, Zhu J, Luo J (2017) Mining fashion outfit composition using an end-to-end deep learning approach on set data. IEEE Trans Multimedia 19:1946–1955
8. Li X, Song J, Gao L, Liu X, Huang W, Gan C, He X (2019) Beyond RNNS: positional self-attention with co-attention for video question answering. In: AAAI Conference on Artificial Intelligence
9. Lin Y, Ren P, Chen Z, Ren Z, Ma J, de Rijke M (2019) Improving outfit recommendation with co-supervision of fashion generation. In: The World Wide Web Conference, pp 1095–1105

10. Lu J, Yang J, Batra D, Parikh D (2016) Hierarchical question-image co-attention for visual question answering. In: Advances in Neural Information Processing Systems (NIPS), pp 289–297
11. Nam H, Ha JW, Kim J (2017) Dual attention networks for multimodal reasoning and matching. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017
12. Seo MJ, Kembhavi A, Farhadi A, Hajishirzi H (2017) Bidirectional attention flow for machine comprehension. In: International Conference on Learning Representations (ICLR)
13. Simo-Serra E, Fidler S, Moreno-Noguer F, Urtasun R (2015) Neuroaesthetics in fashion: modeling the perception of fashionability. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 869–877
14. Vasileva MI, Plummer BA, Dusad K, Rajpal S, Kumar R, Forsyth DA (2018) Learning type-aware embeddings for fashion compatibility. In: The European Conference on Computer Vision (ECCV)
15. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: Advances in Neural Information Processing Systems (NIPS), pp 5998–6008
16. Veit A, Kovacs B, Bell S, McAuley J, Bala K, Belongie S (2015) Learning visual clothing style with heterogeneous dyadic co-occurrences. In: IEEE International Conference on Computer Vision (ICCV), pp 4642–4650
17. Yang Z, He X, Gao J, Deng L, Smola AJ (2016) Stacked attention networks for image question answering. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 21–29
18. Yu Z, Yu J, Fan J, Tao D (2017) Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In: IEEE International Conference on Computer Vision (ICCV), pp 1839–1848

Outfit2Vec: Incorporating Clothing Hierarchical MetaData into Outfits' Recommendation

Shatha Jaradat, Nima Dokoohaki, and Mihhail Matskin

Abstract Fashion Personalisation is emerging as a major service that online retailers and brands are competing to provide. They aim to deliver more tailored recommendations to increase revenues and satisfy customers by providing them options of similar items according to their purchase history. However, many online retailers still struggle with turning customers' data into actionable and intelligent recommendations that reflect their personalised and preferred taste of style. On the other hand due to the ever increasing use of social media, fashion brands invest in influencers' marketing to advertise their brands to reach a larger segment of customers who strongly trust their influencers' choices. In this context the textual and visual analysis of social media can be used to extract semantic knowledge about customers' preferences that can be further applied in generating tailored online shopping recommendations. As style lies in the details of outfits, recommendation models should leverage the fashion metadata ranging from clothing categories and subcategories to attributes such as materials and patterns to overall style description in order to generate fine-grained recommendations. Recently, several recommendation algorithms suggested to model the latent representations of items and users with neural word embeddings approaches which showed improved results. Inspired by Paragraph Vector neural embeddings model, we present **Outfit2vec** and **PartialOutfit2vec** in which we leverage the complex relationship between user's fashion metadata while generating outfits' embeddings. In this paper, we also describe a methodology to generate representative vectors of hierarchically-composed fashion outfits. We evaluate our models using different strategies in comparison to the paragraph embedding models on an extensively-annotated Instagram dataset on recommendation and multi-class style classification tasks. Our models achieve better results specially in whole outfits' ranking evaluations with an average of 22% increase.

S. Jaradat (✉) · N. Dokoohaki · M. Matskin
KTH Royal Institute of Technology, Stockholm, Sweden
e-mail: shatha@kth.se; nimad@kth.se; misha@kth.se

N. Dokoohaki (ed.), *Fashion Recommender Systems*, Lecture Notes in Social Networks, https://doi.org/10.1007/978-3-030-55218-3_5

Keywords Outfits recommendation · Documents embedding · Fashion personalisation

1 Introduction

Personalisation is becoming a leading factor for the current and future success of E-commerce sector. According to a recent study [1], 43% of online purchases are influenced by personalised recommendations, 75% of customers prefer personalised brands messaging, 94% of retail companies are perceiving personalisation as a critical strategy for their success. With the E-commerce fashion industry's worldwide revenue predicted to rise to $712.9 billion by 2022 [1], it is expected that online retailers will direct more efforts into delivering the best perosnalisation services to online fashion shoppers. This is becoming more important with the increasing number of fashion brands and online retail competitors. Customers should experience more recommendation services tailored to their needs and fashion taste. This in turn could be reflected in increased revenue possibilities for retailers. Personalisation can also address one of the leading threats to online fashion which is the unsatisfied customers who frequently return items. Learning more about customers' style preferences can provide them with more satisfying options and possibly reduce the frequency of returns.

Social Media platforms and specifically Instagram is becoming a driving force in the fashion market [2]. Its real power is coming from the smart integration of the increasing popularity of fashion models and the attractive trends of fashion brands. Many followers don't view influencers as paid sales representatives, but rather real people who share their everyday experiences with them. As brands are increasingly investing in the use of social media, influencer marketing has grown into a multibillion-dollar industry [2]. Brands believe that the benefit is beyond liking or sharing a single outfit's post of a fashionista in Instagram. Many followers browse the product and search for it or for a similar one in online shops. This enhances the fast growth of these brands. It is not just for the top influencers, some brands also target Micro-Influencers who have between 1–10 K followers. They believe that because they have a smaller number of followers, this in turn could help them to build more personal relationships with people. As a consequence, their engagement rates are higher, and they are a cost-effective solution [3]. With product tagging feature in Instagram, direct channels for buying can be facilitated. Tagging might be for one or some of the main items in the outfit. Other influencers advertise the product using Hashtags or description in the post and comments. Some followers are curious about all the details in the outfit or the overall style of the fashionistas. This can be noticed in the comments of fashionistas' posts where many conversations start between followers about the outfit's details. Here text and image analysis and recommendation technologies could play a significant role in extracting semantic knowledge about followers' fashion preferences. Building fashion profiles

for these users could enhance the recommendations, with smarter options that are closer to their real taste.

In the last couple of years, a promising new class of neural recommendation models that can generate user and item embeddings by modeling the nonlinearity in **Matrix Factorization** (MF) latent factor models has emerged and has shown improving results. **Word2Vec** model [4] has inspired multiple neural recommendation models such as Prod2Vec [5] and Item2Vec [6]. In the context of Word2Vec [4], the model is able to learn word representations in a low-dimensional continuous vector space using a surrounding context of the word in a sentence. Then as a result, semantically similar words will be close to each other in the resulting embedding space. Researchers have realized that just like word embeddings can be trained by treating a sequence of words in a sentence as context, the same can be done for training embeddings of user's actions (purchases for example) by treating such sequence of user actions as context [7]. This gives us inspiration that a training data that consists of user's outfits which consist of a combination from different clothing categories and subcategories along with their attributes (material/pattern), style and brands can be treated as sequences and learnt by the model. The vocabulary in the model can then be the sequence of outfit details rather than just clothing categories. However, we still deal here with the hierarchical nature of data which is a more complex scenario than the previously described scenarios in literature (where they just consider products or items). In this paper we describe a methodology to generate representative vectors of hierarchically-composed items such as outfits. These representations are then used in our **Outfit2Vec** and **PartialOutfit2Vec** models in which we handle the complex relationship between fashion metadata in order to generate outfits embeddings. We evaluate our models on an extensively-annotated Instagram dataset on recommendation and multi-class style classification tasks. Extensive details about the dataset are available in Sects. 3 and 4.1.

2 Related Works

Some of the recent neural recommendation models have focused on user-to-item recommendations and others brought into focus the value of item-to-item recommendations to avoid problems such as cold start and sparsity of data. Neural Network-based Collaborative Filtering **(NCF)** [8] is an example of a model in which a neural architecture is used to replace the traditional inner product on the latent features of users and items in matrix factorization. In their architecture, they use a multi-layer perceptron (MLP) to learn the user-item interaction function. Their assumption is coming from the fact that the inner product in matrix factorization that simply combines the multiplications of latent features linearly, may not be sufficient to capture the complex structure of user interaction data. A very similar model to NCF is Neural Tensor Factorization **(NTF)** [9] with the addition of including a long-short term memory architecture to model the dynamicity in data that represent how

the users' preferences change over time. **Content2Vec** [10] is a recent model where image and text embeddings are joined together for product recommendation.

Inspired by the famous word embedding model Word2Vec [4], **Item2Vec** was proposed in [6] as a neural item embedding to learn latent representations of items in collaborative filtering scenarios. The main goal of Item2Vec is to learn item-item relations even when the user's information is absent for the purpose of inferring items' similarity. Learning items' similarity is done by embedding items in a low-dimensional space. Another similar model is **Prod2vec** [5], which generates product embeddings from sequences of purchases and performs recommendation based on most similar products in the embedding space. Prod2vec embeds products into real-valued, low-dimensional vector space using a neural language model applied to a time series of user purchases. As a result, products with similar contexts (their surrounding purchases) are mapped to vectors that are nearby in the embedding space. To be able to make meaningful and diverse suggestions about the next product to be purchased, they further cluster the product vectors and model transition probabilities between clusters to use the closest products in the embedding space. **Meta-Prod2Vec** [11] enhances Prod2Vec by injecting the items' metadata (items' attributes and users' past interactions) as side information while computing the low-dimensional embeddings of items. The assumption behind their model is that the item representations that merges the items metadata can lead to better performance on recommendation tasks. **Search2Vec** [12] is one example where multiple embeddings are considered in the learning process. In Search2Vec, the authors have addressed the idea of learning low-dimensional representations of "different" inputs in a common vector space. In their model, search queries and sponsored advertisements are examined together to decide the best matching and/or relevant advertisements to be presented to the user while performing search.

An increasing number of research papers started to focus on applying embeddings approaches for learning outfits' compatibility and similarity for the purpose of recommendation. In [13], Bidirectional LSTMs are used to train sequences of outfits to predict the next item within an outfit that is described from top to bottom. In [14], multimodal representations are learnt from fashion images and their text, and then fed to a neural network that combines the embeddings for all items within an outfit and outputs a score of compatibility based on the relationship between the embeddings of the different items. All this work depends on the idea that compatible items will be closer to each other in the embedding space. In [15] they use Siamese CNNs to learn features of images, and then with Nearest Neighbour retrievals compatible items can be generated. Attention-based techniques were applied in [16] where their primary goal is to complete an outfit based on the context which is the scene in the image (e.g. sea). Personalized Outfit Generation (POG) [17] is a another example of a model that uses a Transformer encoder-decoder architecture with self-attention in order to model embedding representations of the signals of user's preferences that are captured through their latest clicks on items, the visual and textual features of the clothing items and the items' compatibility scores for generating an outfit in a masked item prediction approach.

3 Methodology

It can be noticed that all the previously described neural recommendation models have focused on one type of inputs such as "product" or "item" for finding low-dimensional word representations. However, considering more than one type of input is a more challenging task. In our work, we have sequences of outfits' descriptions such that each outfit is composed of multiple clothing items. Furthermore, each item consists of clothing category, subcategory, with materials and patterns details. Style labels are attached to the whole outfit. Moreover, brand names and hashtags are also available at the outfit's level and they can be used as additional contexts to enhance the outfits' recommendations. We care about having the materials and patterns details describing the items at the instance level rather than the whole sequence. This becomes more important while generating partial recommendations, as the recommendation should describe an instance of clothing including the materials and patterns, and should not be just the category of the clothing item or the material as an example. The clothing metadata was generated from our framework described in [18] by analysing the text and comments of Instagram fashion images using an unsupervised embedding approach. An example of the output of our text mining approach is shown in Fig. 1. For the full details of the followed approach, we refer the reader to [18]. The fashion taxonomy on top of which we based our work was based on the one used in Zalando online shop.[1] An example of this taxonomy is shown in Fig. 2.

A more detailed description of the problem is given as follows: given a set of outfit choices obtained from a defined number of users, we define the following components of each outfit:

- List of clothing categories that exist in the outfit. For example: an outfit that consists of a dress, a coat, shoes and a bag.
- List of clothing subcategories as an additional level of details. For example: the dress is a cocktail dress, the coat is a trench coat, the shoes is a Stiletto heel pumps and the bag is a handbag.
- List of materials from which the clothing items are made. For example: the dress is made from lace as a main component, the coat is made from polyester, the pumps and the handbag are made from leather.
- List of patterns that describe the clothing items. For example: the dress is floral, the coat is plain, the pumps is plain, and the handbag is camouflage.
- List of styles that describe the overall outfit. For example: classic chic style.
- List of brands of the clothing items. For example: the dress and the coat are from Zara, the bag is from Gucci and the pump is from Tamaris.

The objective is to find a D-dimensional representation (*D is the embedding dimensional size*) $v_o \in R^D$ of each outfit o such that similar outfits based on their

[1] https://zalando.com

item_category	bags:48.61%, jackets:35.08%, all_accessories:9.41%, shoes:6.90%
item_sub_category	bag:58.92%, blazer:29.46%, jacket:7.56%, purse:4.06%
materials	leather:40.19%, denim:28.77%, lace:18.15%, polyester:12.88%
patterns	striped:25.58%, herringbone:25.20%, checked:25.03%, print:24.18%
styles	trendy / creative / unique/ fashion-forward -style:33.78%, sporty / casual / easy/ practical -style:24.33%, classic / conservative / timeless/ traditional / crisp -style:23.92%, chic style:17.97%

Fig. 1 An example of the output of the text mining framework we developed in [18]. The unsupervised analysis of the post's text and comments resulted in the identification of the most probable clothing categories in the image along with other attributes that are linked to the clothing items

vectors' similarity can be recommended to the user. To address the challenges of dealing with the hierarchical levels in each outfit's item description, we propose a methodology for generating outfits' representative vectors in the coming subsection. Then, we describe our Outfit2vec model.

3.1 Methodology for Generating Outfits' Representative Vectors

In this section we describe a methodology that can be followed to generate representative vectors of hierarchically-composed items such as outfits. As each outfit is composed of multiple clothing categories, and each of which has different attributes such as materials and patterns, a strategy of projecting these details into vectors representing the whole outfit should be decided. The strategy described here is a combination of rule-based and embedding-based approaches. The main steps in our methodology are defined as follows:

- Mapping of item details into clothing entities
- Projecting the entities into outfit vectors

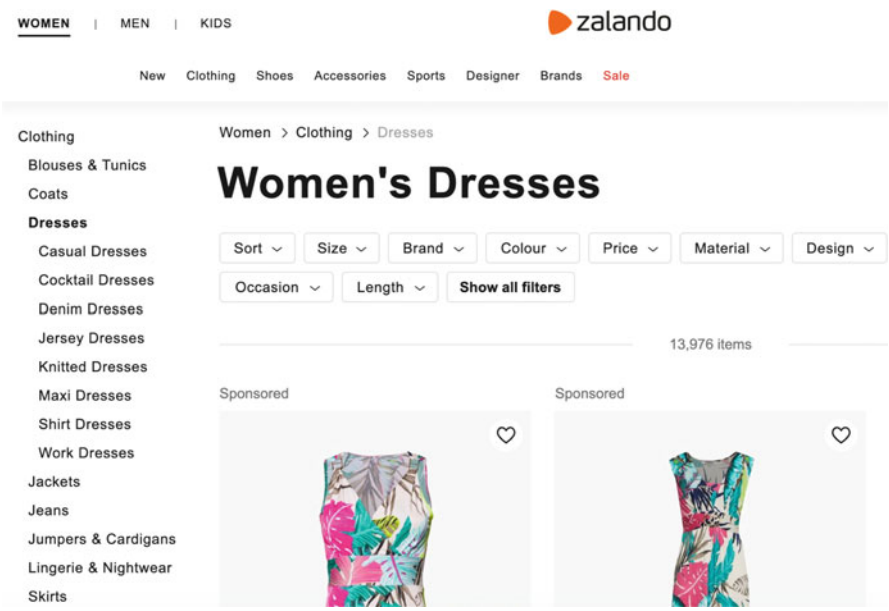


Fig. 2 An example of the fashion taxonomy that is used in Zalando online shop and that we followed in our work. A category like Dresses has multiple subcategories such as: Casual Dresses

3.1.1 Mapping of Item Details into Clothing Entities

As each outfit has a number of clothing categories, subcategories, materials, and patterns, the first step is to map the clothing items' details to instances describing the whole item hierarchically based on the fashion taxonomy that we follow. For example: a Skinny Jeans in an outfit (Jeans is the clothing category and Skinny is the subcategory) with its material's label (Denim) and pattern's label (Plain) should be mapped to a single clothing entity. A semantic description of the clothing item within an outfit would not just include the clothing category but rather the attributes of materials and patterns. This in turn makes the item unique and differentiated from other Jeans instances. These entity instances are then converted to the vector space using embedding-based approach. We experimented with different candidate vectors describing clothing items within an outfit. Candidate vectors vary based on the details they contain such as: clothing categories, subcategories, materials, patterns, styles, brands, and hashtags. They also vary based on the arrangement of clothing details. For example, we experimented with sequences of outfits with clothing categories, followed by the subcategories, then followed by the attributes. Another candidate vector describes the outfit by having the structure: pattern-material-subcategory-category for each item. In other candidate vectors we tried adding the brands and hashtags information in addition to the clothing details in the same sequence. The representative vector that we chose for each clothing entity

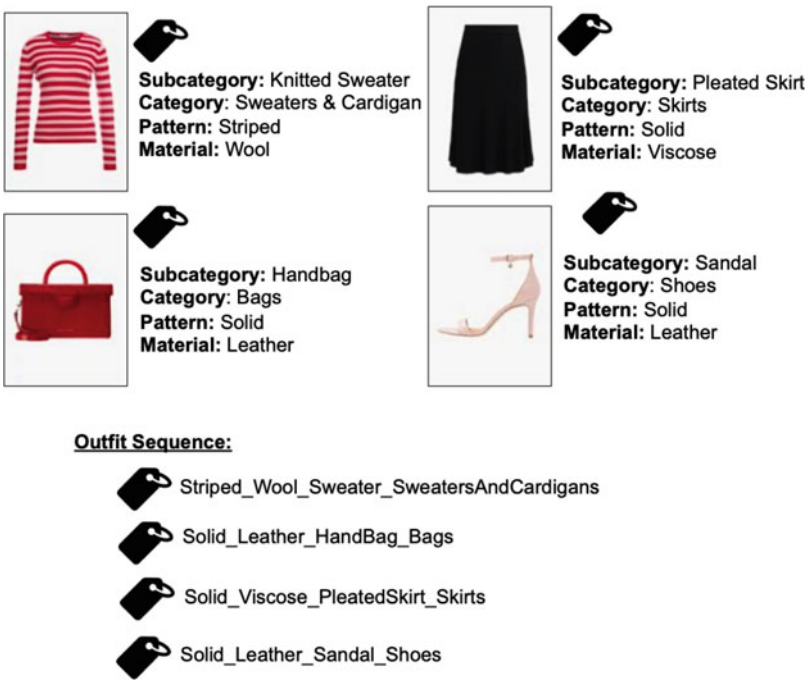


Fig. 3 An illustration of combining clothing metadata into clothing entity descriptions. Each of the items in the figure is part of an outfit, where each item has metadata attributes. In the bottom, we illustrate the sequence built with all the item's metadata, representing an outfit

was mapped from the following structure: *pattern-material-subcategory-category*. We found that combining the clothing details into one word which later results into the model projecting the combined words into a single word vector had the best results in whole outfits prediction. We have also experimented building entities from the structure *pattern-material-subcategory-category* by considering each word individually, and changing the prediction to be for a group of words from the outfit's sequence rather than single word in the partial outfits prediction. The **objective from the arrangement** (pattern-material-subcategory-category) is to facilitate having partial recommendations as we split sequences of outfits and generate recommendations based on some parts of the clothing entities. So, for the predictions they result as four words each from different type (pattern-material-subcategory-category) and describing a single clothing entity. Figure 3 shows an illustrative example of the procedure of combining the metadata of each clothing item to form an entity. In the experiments section, we show the results of experimenting with different candidate vectors while deciding the structure of the representative vector.

3.1.2 Projecting Clothing Entities into Outfit Vectors

As mentioned in the previous subsection, multiple candidate vectors were calculated for each clothing entity to decide the structure of the representative vectors. Then a unified rule-based approach was followed to decide some order of the entities to be projected as outfits' sequences. The order was followed to provide a consistent way of describing outfits. Depending on the assumption that we usually describe an outfit starting from the Jacket and dress and then the shoes and the bags, we have defined some rules which could be stated as the following:

- *Add Jacket or Coat Entity if Exists*
- *If Upper Body and Lower Body Exists:*
  - *Add Upper Body Entity*
  - *Add Lower Body Entity*
- *If Upper Body doesn't Exist and a Dress Exists: Add Dress Entity*
- *Add Tights and Socks Entity if Exists*
- *Add Shoes Entity if Exists*
- *Add Bags Entity if Exists*
- *Add Accessories Entity if Exists*.

Upper body entities consist of the following categories: (1) *Blouses and Tunics*, (2)*Tops and T-Shirts*, (3) *Jumpers and Cardigans*. Lower body categories include: (1) *Skirts*, (2) *Jeans*, (3) *Trousers and Shorts*. The procedure can be generalised as we illustrate in Fig. 4 to other hierarchically-composed structures. As shown in the figure, the first step is to map the raw text to a defined taxonomy from which instances can be defined. A structure of the instance's description should be decided as in our case it was: pattern-material-subcategory-category. Entities are generated by combining components of instances to be projected as single words to the model. A unified order of the sequences can provide a consistent way of describing the predictions. Finally, the generated sequences are used to train the model.

3.2 Outfit2Vec and PartialOutfit2Vec Models

To address the challenges of dealing with the hierarchical levels in each outfit's item description and motivated by the success of distributed language models such as Paragraph Vector (Paragraph2Vec) [19], we present Outfit2Vec and PartialOutfit2Vec. Paragraph2Vec which is an unsupervised model for constructing representations of input sequences of variable length such as paragraphs has two models. In the distributed memory model of Paragraph Vector (PV-DM), every word is mapped to a unique vector, and every paragraph is also mapped to a unique vector to act as a memory that remembers what is missing from the context of the paragraph. The paragraph vector is then concatenated with several word vectors from a paragraph to predict the following word in the given context. This model

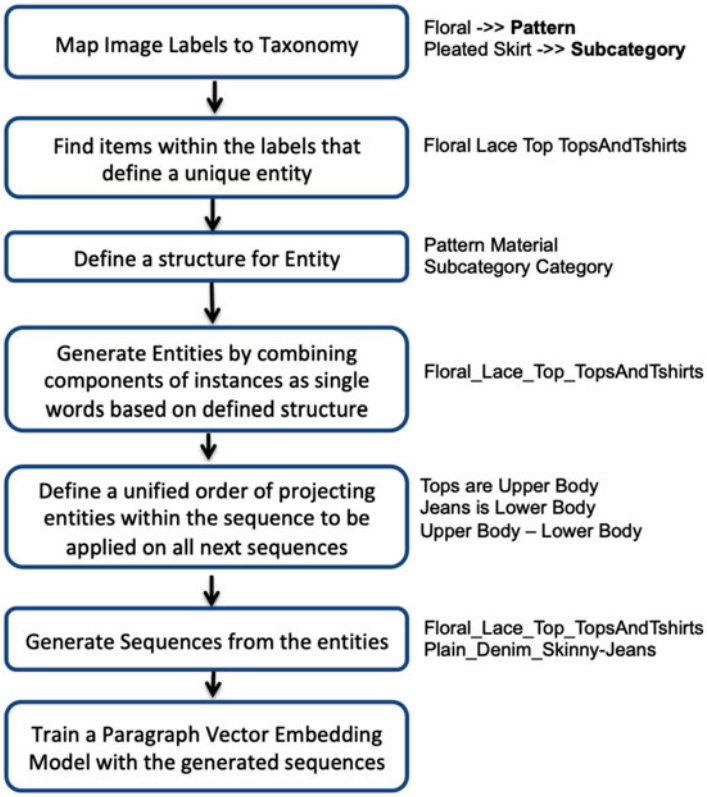


Fig. 4 General process to follow to generate embeddings from Hierarchically-Composed Clothing Instances

addresses the weaknesses of bag-of-words models as it considers the order of words which is important in some tasks. We propose an outfit embedding model that learns user's outfits representations using the PV-DM model but rather than providing the outfit details as separate words to the model, we apply the described methodology in the previous section to present clothing entities as words to the model, and the sequence of clothing entities as paragraphs. In this case, a clothing entity denoted as **i** with the structure [category-subcategory-material-pattern] is mapped to a unique word vector rather than providing the separate parts of the entity as words. The objective of our model is to evaluate the effect of projecting each clothing entity as a single word vector on the ability of the model to provide related fine-grained recommendations at the whole and partial outfit levels. We train our model for two purposes: (a) performing a prediction of a clothing entity within a sequence of an outfit which we refer to as **(PartialOutfit2Vec)**, and (b) performing a prediction of a whole new outfit to the user which we refer to as **(Outfit2Vec)**. For the first purpose, the task is to find the most similar clothing entity to complete a partial sequence of

an outfit. The second task is to find the most similar outfits in the training set from the inferred outfit's vector A formal description is as follows: given a sequence of outfits o_1, o_2, o_3, ... o_N, (N is the total number of outfits), where each outfit o_n is composed of a sequence of clothing entities i_1, i_2, i_3, ... i_T, (T is the total number of entities), the objective of the **PartialOutfit2Vec** model is to maximize the average log probability of predicting the entity i_t given its context of entities within the outfit and the outfit document d and is defined as follows:

$$\frac{1}{T}\sum_{t=k}^{T-k} logp(i_t|i_{t-k}, ...i_{t+k}, d) \tag{1}$$

The prediction task can be obtained through a softmax function over vocabulary I which is extracted from the clothing sequences:

$$P(i_t|i_{t-k}, ...i_{t+k}, d) = \frac{e^{y_{it}}}{\sum_i e^{y_n}} \tag{2}$$

Where y is computed as follows: $y = b + Uh(i_{t-k}, ..., i_{t+k}, d; I)$ Where U,b are the softmax parameters, h is constructed by the concatenation of both clothing entities and outfit paragraph vectors. For larger scale classifications, hierarchical softmax is commonly used. For **Outfit2Vec** the task is finding similar whole vectors from the embedding space (paragraph documents) to the existing sequence o_t.

The model can be tweaked by providing tags to each paragraph, where the tag vector can be concatenated along with the paragraph and word vectors. In our model, we provide the outfit details as tags while training the sequences. Figure 5 illustrates learning outfit and clothing entities vectors framework.
In PV-DM model, the total number of parameters is computed as: N × p + M × q where N is the total number of outfit sequences in a given corpus, M is the number of clothing entities in the vocabulary, each outfit is mapped to p dimensions and each clothing entity is mapped to q dimensions. As the number of parameters can get larger in a larger scale coprus, we have also experimented using Paragaraph Vectors Distributed Bag of Words model (PV-DBOW) where the model is supposed to store less data as it doesn't store the word (clothing entities) vectors. So, the context of words in the input is ignored, and the prediction is performed from samples of the paragraphs. A detailed comparison of the behavior of our models in both ways is shown in the experiments section.

4 Experimental Pipeline

In this section, we present our evaluation experiments on top-k recommendations for whole and partial outfits. Then, we present our evaluation on multi-class style classifications. A detailed description of our dataset is also provided.

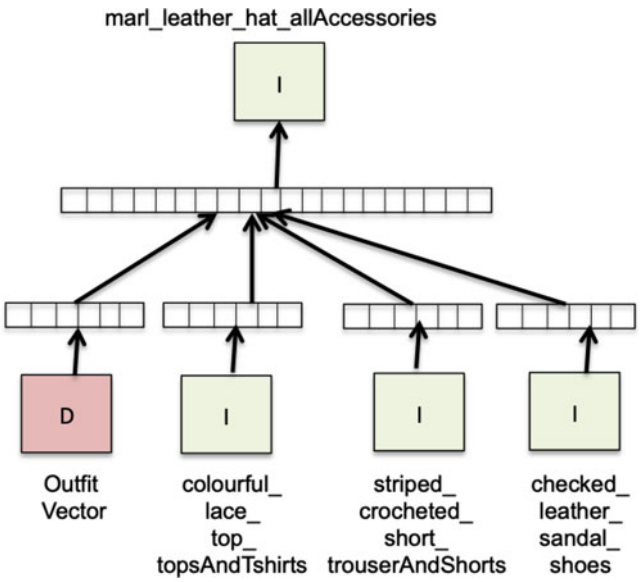


Fig. 5 An illustration of learning outfit vector Outfit2Vec(PV-DM). The clothing entities are presented as words and the sequence of clothing entities form the outfit's vector

4.1 Datasets

The experiments were performed on an extensively-annotated Instagram dataset that we gathered in our work [20] from a community of fashion models (RewardStyle).[2] The data is in the form of images, image captions, user comments, and hashtags associated with each post. We applied the text analysis procedure which was described in [18]. This resulted in annotating each image with a list of categories, sub-categories, materials, patterns and styles with percentages according to the importance. We employed a subset consisting of around **50,000** posts from the collected Instagram dataset, each with the described annotations. Some statistics about the dataset are as follows: the number of unique outfits is 14,573, the number of clothing entities is 72,479, and the number of unique clothing entities is 4790. The number of outfits vary per user as it is based on the amount of posts collected from each user.

[2] https://www.rewardstyle.com/

4.2 Whole Outfits Recommendation (Outfit2Vec)

In our experiments, the models are trained with sequences of whole outfits with 80% split for training and 20% for validation. An example of an outfit sequence is "colourful lace top topsAndTshirts – striped crocheted short trouserAndShorts -checked leather sandal shoes". The difference between whole and partial outfits' evaluation is in the input provided to the model during validation. In whole outfits experiments, a complete sequence is provided to the model for inference of its representation. We consider the next sequence in this case to the be the ground truth as our objective is to evaluate the model's ability to predict the next outfit based on the history of outfits in the corpus. The most similar top k paragraph vectors to the inferred one are then retrieved. To evaluate the top k results, we use the following measures: Normalized Discounted Cumulative Gain (NDCG), Mean Reciporcal Rank (MRR) and Mean Average Precision (MAP). We compare Outfit2Vec (PV-DM) in two cases: by modeling the clothing entities as single words *Structured-Entities* and by providing the clothing details as separate words in the outfit's sequence *Structured-Words*. An example of a clothing entity within a structured entity sequence is: floral-lace-top-topsAndTshirts (one word), and an example of a clothing entity within a structured word sequence is: floral lace top topsAndTshirts. So, the difference between Structured Entities and Structured Words models are in projecting the details into single or separate words but both of them reflect the same structure. The same approach is applied for Outfit2Vec (PV-DBOW) models where we donate the models under evaluation as *Structured-Entities-DBOW* and *Structured-Words-DBOW*. We also run our experiments on randomly arranged sequences which we denote here as PV-DBOW-Random and PV-DM-Random. We experimented different values for the vector size used for ParagraphVector training, and we selected 200. We train the models under evaluation for 30 epochs.

A ranking measure that we chose in our evaluation is Normalized Discounted Cumulative Gain (NDCG) which is one of the mostly used measures of effectiveness in information retrieval algorithms. The usefulness of a retrieved document is decided using a graded relevance scale based on the document's position in the result list. The gain of the result list is accumulated from the top to the bottom of the list with the gain of each result discounted at lower ranks. With that, higher relevant documents are more useful when appearing earlier in a search engine result list (having higher ranks). This is a very relevant measure for the outfits' prediction evaluation as the users expect to find the most relevant suggestions at the top of the recommendations list. For a prediction, the normalised discounted cumulative gain nDCG is computed as: $nDCG_p = \frac{DCG_p}{IDCG_p}$ where DCG_p is the discounted cumulative gain and $IDCG_p$ is the ideal DCG which reflects the maximum possible DCG through the position p. They are calculated using the following formulas respectively: $DCG_p = \sum_{i=1}^{p} \frac{rel_i}{log_2(i+1)}$, $IDCG_p = \sum_{i=1}^{|Rel_p|} \frac{2^{rel_i}-1}{log_2(i+1)}$ where rel_i is the graded relevance of the result at position i. In our context i is the relevance of the prediction result. In Table 1 we show the average NDCG for all retrieved results for

Table 1 Evaluation Results of Avg. NDCG, MAP and Avg. MRR for k = 30 and 40 for all the models under comparison in whole outfits prediction task

Model	NDCG30	NDCG40	MAP30	MAP40	MRR30	MRR40
Structured-Entities	**0.22**	**0.33**	0.37	0.41	0.06	0.06
Structured-Words	0.08	0.09	0.39	0.44	0.06	0.05
Structured-Entities-DBOW	**0.30**	**0.38**	0.37	0.41	0.07	0.07
Structured-Words-DBOW	0.08	0.10	0.21	0.23	0.04	0.04
PV-DBOW-Random	0.08	0.09	0.13	0.14	0.03	0.03
PV-DM-Random	0.07	0.07	0.23	0.23	0.04	0.03

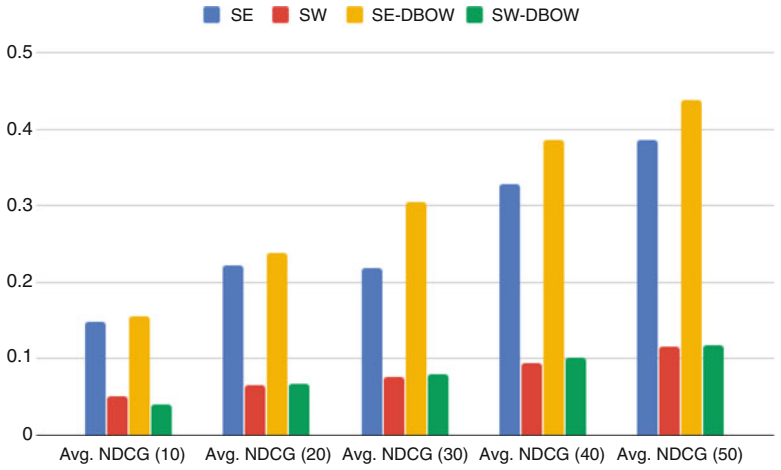


Fig. 6 Illustration of the improved NDCG results achieved using the representative vectors methodology when applied to the models under study

k = 30 and 40 where k is the number of retrieved predictions. We chose the values of k based on the assumption that it is a suitable number of retrieved recommendations for an online shopper to browse from. We have also computed the results for k = 10, 20 and k = 50 which is shown in Fig. 6.

As can be seen from the statistics in Table 1, *Structured-Entities* outperformed *Structured-Words* with an average of increase of 19% for the measured k values. For Outfit2Vec (PV-DBOW) models, it can be also seen that using the Structured Entities methodology resulted in an average increase of 25%. The average increase of *Structured-Entities-DBOW* when compared to *Structured-Words-DBOW* was higher than the average increase of *Structured-Entities (PV-DM)* compared to *Structured-Words (PV-DM)*. This can be explained by the ignorance of order in the PV-DBOW models. That's why the introduced structured approach achieves more significant improvement in PV-DBOW model when the details are projected as single words. Both models have outperformed *PV-DBOW-Random* and *PV-DM-Random* which are composed by random sequences of outfits without applying our methodology. Figure 6 shows a graphical comparison of the increase in NDCG

values for the models under evaluation. Abbreviations such as SE (*Structured Entities*) and SW (*Structured Words*) were used to simplify the visualisations.

Another statistical measure that we compute is the Mean Reciprocal Rank (MRR). The mean reciprocal rank is the average of the reciprocal ranks of outfits retrievals for a sample of predictions Q and is computed as follows: $MRR = \frac{1}{Q}\sum_{i=1}^{|Q|}\frac{1}{rank_i}$ Where $rank_i$ refers to the rank position of the first relevant document for the i_{th} query. The sequences that we handle are composed of clothing categories, subcategories, materials and patterns. To calculate the matching between the ground truth and the predictions, we calculate if all the items within the goal are available in the prediction sequence to be counted as a correct result. As can be seen in Table 1, for *Structured-Entities* and *Structured-Words*, the models performed similarly in MRR. However, a positive increase was noticed for *Structured-Words-DBOW* when compared to *Structured-Entities-DBOW*.

Mean Average Precision (MAP) is a classical evaluation metric in binary classification algorithms and information retrieval tasks. We are interested in calculating the precision at top k results. But since precision is a binary metric used to evaluate the models with binary output and in our case we have a multiclass output. Thus, we need to translate our output to a binary output (relevant and not relevant outfits). To do the translation, we assume that a relevant outfit intersects with a number of items above a certain threshold from the number of items in the ground truth. The threshold we have used is 0.7. A relevant outfit for a specific user means that it contains relevant materials and patterns and categories that are similar to what the user prefers in the ground truth. So, we compute precision and recall at K where k is a user defined integer that reflects the number of recommended outfits to the user. Based on the assumption that a recommended item should be a relevant item, Precision at k is the proportion of recommended items in the top-k set that are relevant and is calculated as follows: $P(k) = \frac{RR@k}{R@k}$ Where RR is the number of recommended items that are relevant @K, and R is the number of recommended items @k. $AP(k) = \frac{1}{m}\sum_{k=1}^{N} P(k)$ if kth item was relevant) where m is the number of outfits. Then we calculate the MAP which is the average of the average precision metric over all the results' list. What is noticed from the results is that the MAP values for *Structured-Entities (DM)* and *Structured-Words-DBOW* were similar. However, *Structured-Entities-DBOW* had increased values when compared to *Structured-Words-DBOW*. It is also expected that they both outperform the models that were trained with random sequences. Figure 7 illustrates the MAP performance for all the models. A clear decrease is noticed towards the models that were not trained with the structured methodology. As described in the methodology section, we have compared different candidate vectors before choosing the structure we use in our experiments. Figure 8 illustrates the results of evaluating the candidate vectors with the metrics NDCG, MRR, precision and recall @k = 1. The chosen representative vector had the highest values for the evaluated measures.

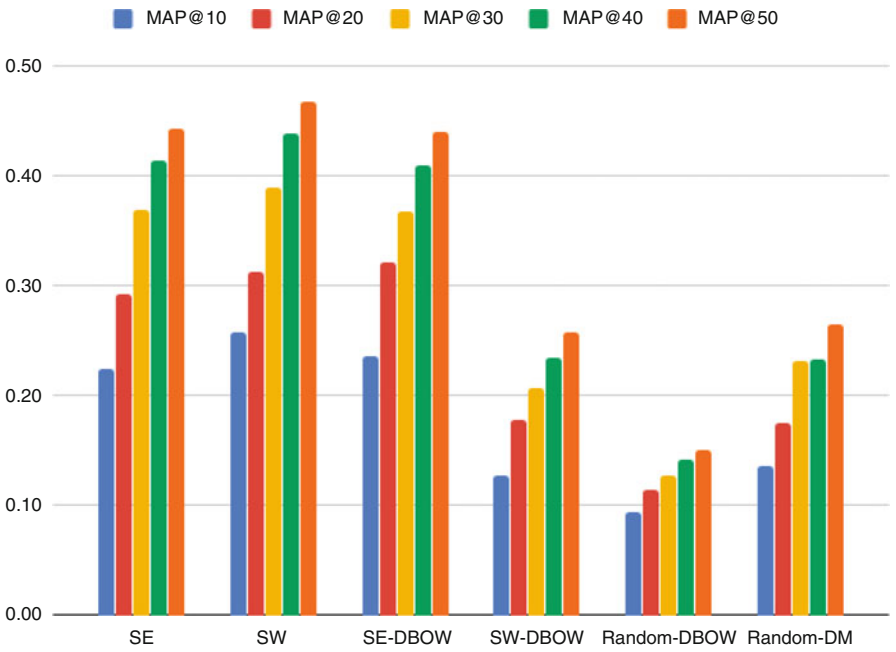


Fig. 7 Illustration of the MAP performance for all the models

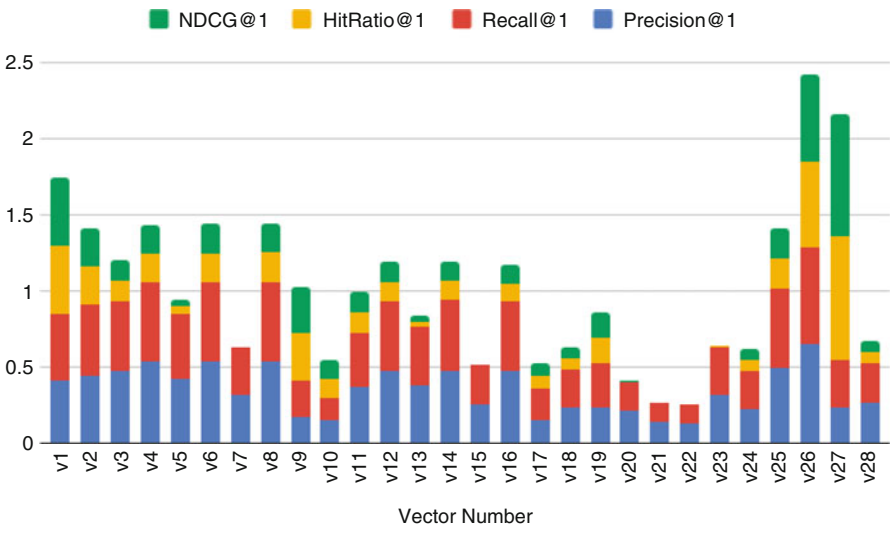


Fig. 8 Illustration of representative vectors choice strategy

4.3 Partial Outfits Recommendation

As previously described, Outfit2Vec represents the whole-outfits recommendation task, and PartialOutfit2Vec represents the recommendation of a clothing entity within an outfit's sequence. In partial outfits experiments, we split the outfit sequences in a way that includes a group of clothing entities and infers the remaining entity. For example: in the sequence "colourful lace top topsAndTshirts – striped crocheted short trouserAndShorts -checked leather sandal shoes", the first two entities are provided to the model for inference of its representation where the top k results of the most similar completing part of sequence are retrieved and compared to the ground truth which is the last entity in the sequence. The same approach of structured entities and structured words was applied in this experiment, so the models under evaluation are: Structured-Entities (DM), Structured-Words (DM), Structured-Entities (DBOW) and Structured-Words (DBOW). PartialOutfit2Vec was compared with Word2Vec (Skip-gram) for the task of predicting a clothing entity from a sequence of an outfit. We have also compared against the Word2Vec (CBOW) model but the results were significantly lower so we focused on skip-gram model. The hyperparameters used for Word2Vec are: window size: 10 and embedding size 200. The values were chosen based on the average length of the outfit's sequence details. As can be noticed from Table 2, both PartialOutfit2Vec (DM) and PartialOutfit2Vec (DBOW) have outperformed Word2Vec. This can be explained by the ability of ParagraphVector-based models to capture the relationship between components of a paragraph more than word2vec models. Interestingly, the **Structured Words** trained models have performed significantly better than the **Structured Entities** models for the partial outfit prediction task. We explain this improved performance by the length of predictions as it is shorter than the whole outfits predictions. While at the whole outfits prediction, the whole sequence is to be predicted and the structured entities have shown more improvements in that scenario. For MRR evaluations, it was noticed that both PartialOutfit2Vec models have outperformed Word2Vec. We conclude that the structured approach has achieved improvements in prediction generally, but the structured entities showed additional value at the whole-outfits prediction task.

Table 2 Evaluation Results of Avg. NDCG, MAP and Avg. MRR for k = 30 and 40 for all the models under comparison in **partial** outfits prediction task

Model	NDCG30	NDCG40	MAP30	MAP40	MRR30	MRR40
Structured-Entities	0.43	0.60	0.26	0.34	0.19	0.26
Structured-Words	**0.77**	**0.86**	**0.65**	**0.67**	**0.59**	**0.58**
Structured-Entities-DBOW	0.54	0.67	0.34	0.38	0.28	0.31
Structured-Words-DBOW	**0.77**	**0.79**	**0.82**	**0.81**	**0.74**	**0.75**
Word2Vec-SkipGram	0.07	0.08	0.19	0.29	0.05	0.05

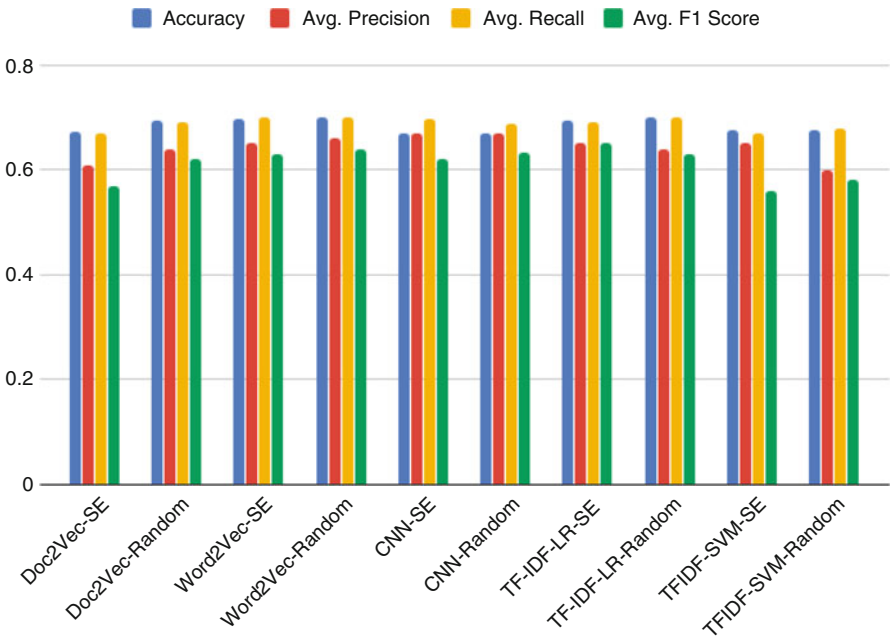


Fig. 9 Comparison of Multiclass Style Classification results for different baselines using the described methodology

4.4 MultiClass Classification Evaluation

As one of our objectives is to classify outfits into their relevant styles, we have compared the effect of the suggested methodology on the performance of the following baselines in a multiclass style classification experiment: (a) Convolutional Neural Network (CNN), (b) PV-DBOW, (c) Word2Vec, (d) TF-IDF for transforming text and Logistic Regression classification, (e) TF-IDF for transforming text and Support Vector Machines (SVM) classification. Figure 9 shows that the models performed similarly when provided with input after applying our methodology. Average accuracy, precision, recall and F1 score were calculated. As the figure shows, the performance of the classification was similar. We conclude from this experiment that our methodology affects the recommendation evaluation task more than the classification. We explain this by the amount of details that affect the recommendation task, whereas in the classification, only the style value is predicted, which shows the value of our approach in a fine-grained recommendation evaluation tasks.

4.5 Discussion

As presented in the experiments section, projecting the clothing details as separate entities have improved the accuracy of retrievals in a significant way at the whole-outfits recommendation evaluation. The improvement was noticed on the rank evaluation measures which is very important for the outfits recommendation task. Both whole and partial outfits prediction tasks have been improved using the described **Structured Entities** and **Structured Words** approaches. We have also noticed that our methodology can be more important to fine-grained recommendation than to simple classification tasks. As shown in the classification task's evaluation, the models performed similarly with no noticed significant changes. This can be explained by the level of difficulty we have in the recommendation task, as the model is expected to find predictions similar to the ground truth in terms of multiple clothing details, while for the classification, the number of labels which are the style labels in this case are limited. No pre-trained models were used in our experiments for two major reasons: (a) in our methodology we create new words in the model's vocabulary by combining the clothing entity details, so having a pre-trained words model will not add a value or enhance the training in this case, (b) PV-DBOW models use the paragraph vectors for training and then for inference, so having a pre-trained words model will not add a value as well. Same applies for PV-DM based models where the training happens by concatenating the words and the paragraph vector.

5 Conclusions and Future Work

We present Outfit2Vec and PartialOutfit2Vec models for learning clothing embeddings. In our models, we deal with a complex scenario of hierarchically-composed clothing items where we aim to recommend whole- and partial- outfits that consist of multiple clothing entities. Our objective from this work is to present a general strategy of dealing with learning representations of hierarchically-composed complex structures to learn their embeddings as unique instances within a taxonomy. We showed in our experiments that we run on an extensively-annotated Instagram dataset on recommendation and multi-class classification tasks the improvements achieved with our approaches. For our future work, we plan to experiment on larger-scale samples of data to evaluate the performance of our model in different settings.

References

1. Orendorff A (2019) The state of the ecommerce fashion industry: statistics, trends and strategy. https://www.shopify.com/enterprise/ecommerce-fashion-industry. Accessed 13 Jan 2020
2. Martineau P (2018) Inside the pricy war to influence your Instagram Feed. https://www.wired.com/story/pricey-war-influence-your-instagram-feed/. Accessed 13 Jan 2020
3. Brown B (2018) 19 Micro-influencer statistics you must know in 2018. https://www.grin.co/blog/19-micro-influencer-statistics-you-must-know-in-2018. Accessed 13 Jan 2020
4. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781
5. Grbovic M, Radosavljevic V, Djuric N, Bhamidipati N, Savla J, Bhagwan V, Sharp D (2015) E-commerce in your inbox: product recommendations at scale. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 1809–1818
6. Barkan O, Koenigstein N (2016) Item2vec: neural item embedding for collaborative filtering. In: 2016 IEEE 26th international workshop on machine learning for signal processing (MLSP). IEEE, pp 1–6
7. McCormick C (2018) Applying word2vec to recommenders and advertising. http://mccormickml.com/2018/06/15/applying-word2vec-to-recommenders-and-advertising/. Accessed 13 Jan 2020
8. He X, Liao L, Zhang H, Nie L, Hu X, Chua TS (2017) Neural collaborative filtering. In: Proceedings of the 26th international conference on world wide web, pp 173–182. International World Wide Web Conferences Steering Committee
9. Wu X, Shi B, Dong Y, Huang C, Chawla N (2018) Neural tensor factorization. arXiv preprint arXiv:1802.04416
10. Nedelec T, Smirnova E, Vasile F (2017) Specializing joint representations for the task of product recommendation. In: Proceedings of the 2nd workshop on deep learning for recommender systems. ACM, pp 10–18
11. Vasile F, Smirnova E, Conneau A (2016) Meta-prod2vec: product embeddings using side-information for recommendation. In: Proceedings of the 10th ACM conference on recommender systems. ACM, pp 225–232
12. Grbovic M, Djuric N, Radosavljevic V, Silvestri F, Baeza-Yates R, Feng A, Ordentlich E, Yang L, Owens G (2016) Scalable semantic matching of queries to ads in sponsored search advertising. In: Proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval. ACM, pp 375–384
13. Han X, Wu Z, Jiang YG, Davis LS (2017) Learning fashion compatibility with bidirectional lstms. In: Proceedings of the 25th ACM international conference on multimedia. ACM, pp 1078–1086
14. Bettaney EM, Hardwick SR, Zisimopoulos O, Chamberlain BP (2019) Fashion outfit generation for E-commerce. arXiv preprint arXiv:1904.00741
15. Veit A, Kovacs B, Bell S, McAuley J, Bala K, Belongie S (2015) Learning visual clothing style with heterogeneous dyadic co-occurrences. In: Proceedings of the IEEE international conference on computer vision, pp 4642–4650
16. Kang WC, Kim E, Leskovec J, Rosenberg C, McAuley J (2019) Complete the look: scene-based complementary product recommendation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 10532–10541
17. Chen W, Huang P, Xu J, Guo X, Guo C, Sun F, Li C, Pfadler A, Zhao H, Zhao B (2019) POG: personalized outfit generation for fashion recommendation at Alibaba iFashion. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 25 Jul 2019, pp 2662–2670
18. Hammar K, Jaradat S, Dokoohaki N, Matskin M (2018) Deep text mining of instagram data without strong supervision. In: 2018 IEEE/WIC/ACM international conference on web intelligence (WI). IEEE, pp 158–165

19. Le Q, Mikolov T (2014) Distributed representations of sentences and documents. In: International conference on machine learning, pp 1188–1196
20. Jaradat S, Dokoohaki N, Wara U, Goswami M, Hammar K, Matskin M (2019) TALS: a framework for text analysis, fine-grained annotation, localisation and semantic segmentation. In: 2019 IEEE 43rd annual computer software and applications conference (COMPSAC), vol 2. IEEE, pp 201–206

Part IV
Sizing and Fit Recommendations

Learning Size and Fit from Fashion Images

Nour Karessli, Romain Guigourès, and Reza Shirvany

Abstract Finding clothes that fit has increasingly become a hot topic in the e-commerce fashion industry. Mainly due to causing frustration on the customers side, and the large ecological and economical footprint on the companies side. Most approaches addressing this problem are based on statistical methods relying on historical data of articles purchased and returned to the store. Such approaches suffer from the cold start problem for the thousands of articles appearing on the shopping platforms every day, for which no prior purchase history is available. We propose to employ visual data to infer size and fit characteristics of fashion articles. We introduce SizeNet, a weakly-supervised teacher-student training framework that leverages the power of statistical models combined with the rich visual information from article images to learn visual cues for size and fit characteristics, capable of tackling the challenging cold start problem. We demonstrate the strong advantage of our approach through extensive experiments performed on thousands of textile garments, including dresses, trousers, knitwear, tops, etc. from hundreds of different brands.

Keywords Size and fit · Fashion e-commerce · Cold-start recommendation · Computer vision · Weakly supervised learning · Teacher student learning

1 Introduction

Fashion industry has been a major contributor to the economy in many countries. Fashion e-commerce, in particular, has largely evolved over the past few years becoming a major player for delivering competitive and customer-obsessed products and services. Recent studies have shown that finding the right size and fit is among the most important factors impacting customers' purchase decision making process

N. Karessli (✉) · R. Guigourès · R. Shirvany
Zalando SE, Berlin, Germany
e-mail: Nour.Karessli@zalando.de; Romain.Guigoures@zalando.de; Reza.Shirvany@zalando.de

N. Dokoohaki (ed.), *Fashion Recommender Systems*, Lecture Notes in Social Networks, https://doi.org/10.1007/978-3-030-55218-3_6

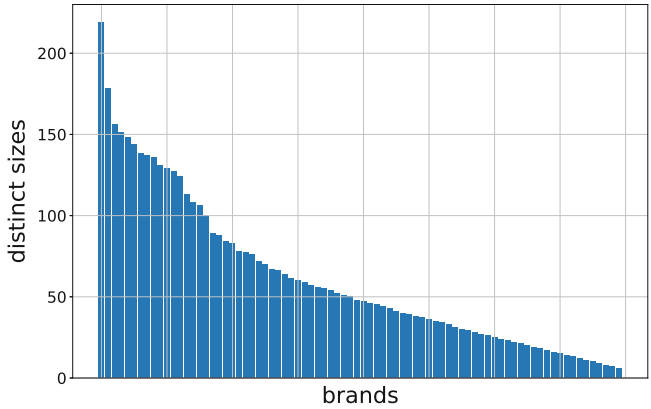


Fig. 1 Number of distinct sizes offered by 80 major brands in women textile categories

and their satisfaction from e-commerce fashion platforms [1]. In the context of online shopping, customers need to purchase clothes without trying them on. Thus, the sensory feedback phase about how the article fits via touch and visual cues is naturally delayed, leading to uncertainties in the buying process. As a result, a lot of consumers remain reluctant to engage in the purchase process in particular for new articles and brands they are not familiar with.

To make matters worse, fashion articles including shoes and apparel have important sizing variations primarily due to: 1. a coarse definition of size systems for many categories (e.g. small, medium, large for garments); 2. different specifications for the same size according to the brand; 3. different ways of converting a local size system to another, as an example in Europe garment sizes are not standardized and brands don't always use the same conversion logic from one country to another. With the aim of understanding the problem space complexity from the industry side, we present in Fig. 1 the number of distinct sizes per brand from 80 major brands in women textile categories. Here, each vertical bar represents the number of unique different sizes offered by the brand. As seen, most of the major brands offer more than 20 different distinct sizes.

A way to circumvent the confusion created by these variations is to provide customers with size conversion tables which map aggregated physical body measurements to the article size system. However, this requires customers to collect measurements of their bodies. Interestingly, even if the customer gets accurate body measurements with the aid of tailor-like tutorials and expert explanations, the size tables themselves almost always suffer from high variance that can go up to one inch in a single size. These differences stem from either different aggregated datasets used for the creation of the size tables (e.g. German vs. UK population) or are due to vanity sizing. The latter happens when a brand deliberately creates size inconsistencies to satisfy a specific focus group of customers based on age, sportiness, etc. which represent major influences on the body measurements

presented in the size tables [2–4]. The combination of the mentioned factors leaves the customers alone to face a highly challenging problem of determining the right size and fit during their purchase journey.

In recent years, there has been a lot of interest in building recommendation systems in fashion e-commerce with major focus on modeling customer preferences using their past interactions, taste, and affinities [5–7]. Other work involve image classification [8, 9], tagging and discovery of fashion products [10, 11], algorithmic outfit generation and style extraction [12], and visual search that focuses on the problem of mapping studio and street images to e-commerce articles [13, 14]. In this context, only very few research work have been conducted to understand how fashion articles behave from the size and fit perspective [15–20], with the main goal of providing a size advice for customers, mainly by exploiting similarities using article sales and returns data, as detailed in Sect. 2. Returns have various reasons such as "don't like the article, article damaged, size problems, etc." We propose a weakly-supervised [21] teacher-student approach [22–24] where we first use article sales and size related returns to statistically model whether an article suffers from sizing issues or conversely has a normal size and fit behaviour. In this context, we don't have access to size and fit expert-labeled data for articles, and thus, only rely on weakly-annotated data from the returns process. We then make use of a teacher-student approach with curriculum learning [25] where the statistical model acts as the teacher and a CNN-based model, called SizeNet, acts as the student that aims to learn size issue indicators from the fashion images without direct access to the privileged sales and returns data.

The contributions of our work are three-fold: 1. We demonstrate, for the first time to our best knowledge, the rich value of fashion images in inferring size characteristics of fashion apparel; 2. At the same time our approach is novel in using the image data to effectively tackle the cold start problem that is known to be a very challenging topic in the literature; 3. We propose a teacher statistical model that uses crowd's subjective and inaccurate feedback (highly influenced by personal perception of article size) to generate large scale confidence-weighted weak annotations. This enables us to control the extent to which the weak annotations influence the quality of the final model, and we demonstrate that not applying this approach, i.e. treating weak labels uniformly, highly degrades the quality of the learned model.

The outline of the paper is as follows. In Sect. 2 we present related work. In Sect. 3 we present the proposed approach; Sect. 3.1 presents the teacher-student framework, Sect. 3.2 presents the statistical model predicting size issues taking into account the article's category, sales period, number of sales, and number of returns due to size problems. In Sect. 3.3 we introduce the architecture of the SizeNet along with the curriculum learning approach using the statistical class labels and their confidence scores to train SizeNet on fashion images. In Sect. 4 we present two baselines, experimental results, and discussion to assess the quality of the SizeNet results over different categories of garments including dresses, trousers, knitwear, and tops/blouses. Furthermore, we analyze different cases going from warm to cold start. Finally in Sect. 5, we draw conclusions and discuss future work directions.

2 Related Work

The topic of understanding article size issues, and more generally predicting how e-commerce fashion articles may fit customers is challenging. Recent work has been done for supporting customers by providing size recommendations in [15] and [16]. Both approaches propose a personalized size advice to the customer using different formulations. The first one uses a skip gram based word2vec model [26] trained on the purchase history data to learn a latent representation for articles and customers in a common size and fit space. Customer vector representation is obtained by aggregating over purchased articles, and a gradient boosted classifier predicts the fit of an article to a specific customer. The second publication proposes a hierarchical Bayesian approach to model what size a customer is willing to buy along with the resulting return status (article is kept, returned because it's too big, or returned because it's too small).

Following a different approach, the authors of [17] propose a solution, using the purchase history for each customer, to determine if an article of a certain size would be fit or would suffer a size issue (defined as large, or small). This is achieved by iteratively deducing the true sizes for customers and products, fitting a linear function based on the difference in sizes, and performing ordinal regression on the output of the function to get the loss. Extra features are simply included using addition to the linear function. To handle multiple personas using a single account, hierarchical clustering is performed on each customer account before doing the above. An extension of that work proposes a Bayesian approach on a similar model [18]. Instead of learning the parameters in an iterative process, the updates are done with mean-field variational inference with Polya-Gamma augmentation. This method therefore naturally benefits from the nice advantages of Bayesian modeling – the uncertainty outputs, and the use of priors. It however does not tackle the cold start problem – where zero or very few article sales and returns are available.

In the fashion e-commerce context, everyday thousands of articles are introduced to the catalog. The life cycle of most articles in fashion e-commerce is usually short – after a few weeks, the article is out of stock and removed from the assortment. The "hassle-free" return policy of e-commerce platforms allows customers to return items with no additional cost, whenever they desire up to multiple-weeks from the purchase date. When customers return an item, they can disclose a return reason, for example "did not like the item", "item did not fit" or "item was faulty". In this work, we are interested in estimating whether an item has sizing issues, therefore, we make use of the weakly-annotated size related return data where a customer mentions that an article is not fitting. It is important to note that article returns can reach the warehouses only after multiple days (if not weeks) from the date of the article activation resulting in a cold start period.

Indeed, all the aforementioned publications state that data sparsity and cold start problem are the two major challenges in such approaches. They propose to tackle those challenges with limited success by exploiting article meta data or taxonomies in the proposed models. In this paper, we leverage the potential of learning visual cues from fashion images to better understand the complex problem of article size

and fit issues, and at the same time, provide insights on the value of the image-based approach in tackling the cold start problem. A major advantage of images over meta data or taxonomies lies in the richness of the imagery data, in addition to a lower subjectivity of the information when compared to the large list of ambiguous fashion taxonomies- for example, a slim jeans from Levi's does not follow the same physical and visual characteristics as a slim jeans from Cheap Monday, as both brands target different customer segments.

3 Proposed Approach

In this section, we explain the details of the proposed approach to infer article sizing problems from images. We first start by introducing our weakly-supervised teacher-student framework. Then we introduce our statistical model, and finally we discuss the SizeNet – our CNN model capable of predicting size issues from fashion images thanks to the insights from the statistical model. Figure 2 illustrates the overall architecture of the proposed approach.

3.1 Teacher-Student Learning

The concept of training machine learning models following a teacher-student approach is a well-known concept where its mention in the community dates back

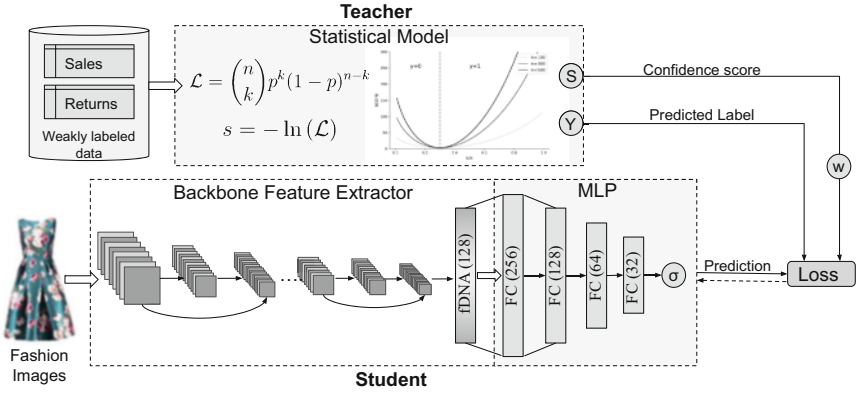


Fig. 2 Architecture of the proposed teacher-student approach. On the top, the statistical model acts as the teacher with direct access to the privileged sales and returns data. On the bottom, SizeNet is shown as the student, composed of a CNN backbone feature extractor followed by a multi-layer perceptron. The student is trained using fashion images and weak annotations and their confidence scores acquired from the teacher

to the 1990s. In recent years, however, there has been an extensive interest in further developing the teacher-student and related learning frameworks such as the curriculum learning approaches [23–25]. Interestingly, to motivate the teacher-student training approach, [23] illustrates a cold start problem using the example of an outcome of a surgery three weeks after it has occurred. The classifier is trained on historical data where the historical data contains privileged information about the procedure and its complications during the three weeks following the surgery. The model trained using the privileged data is then considered as the teacher. A second model is trained on the same samples but without using the privileged information. Therefore, this second model – the student – tries to learn from the insights given by the teacher to replicate the outcome of the teacher without directly having access to the privileged data. Vapnik and Izmailov [23] uses the teacher-student approach along with a support vector machine (SVM) [22]. In the non-separable case – i.e. when there exists no hyperplane separating classes – SVM needs to relax the stiff condition of linear separability and allow misclassified observations. As shown in [27], the teacher also helps refine the decision boundary and can be assimilated to a Bayesian prior.

Following a similar concept, curriculum learning [25] suggests to train classifiers first on easier (or more confident) samples and gradually extend to more complex (or less confident) samples. It is shown that this strategy leads to better models with increased generalization capacities. Most approaches using a teacher-student learning strategy [23–25] derive the importance of the samples from the teacher classifier. In this paper, we build a statistical model that has privileged information on the article sales and returns data, as the teacher, and train the SizeNet model, as the student, on fashion images using a confidence score to weigh the samples from the teacher. In other words, the approach is transferring knowledge from privileged information space to the decision space. Though the teacher model in our case does not use the article images as input, it leverages the privileged historical data of sold articles (privileged information space), and the student uses this knowledge to learn from images in the decision space. The confidence-weighted annotations generated by the teacher enables us to control the extent to which these weak annotations (built from the crowd's subjective and inaccurate feedback) influence the quality of the final model, and thus, delivering better learned model.

3.2 Statistical Modeling

In this work, we opt for a simplifying approach and formulate the sizing problem as a binary classification problem. Thus, we arrange articles based on their sizing behavior into two categories. Class 1 groups articles that are annotated as having a size issue, e.g. too small, shoulder too tight, sleeves too long, etc. Class 0 groups other articles with no size issue. To allocate articles to the appropriate class, we need to consider two factors:

- *The category*: Article categories are diverse; some examples are shoes, t-shirts, dresses, trousers, etc. Generally, for each category we expect a different return rate and sizing issues. As an example, high heels have a higher size related return rate than sneaker shoes, since customers are more demanding in terms of size and fit with the former than the latter. Therefore, we should consider for each category the amount of size related returns in the category compared to that of its average.
- *The sales period*: The usual life cycle of an article starts with its activation on the e-commerce platform, after which customers start purchasing the article and potentially return the article if it does not meet their expectations. This process naturally results in a time dependency in the purchase and return data. Therefore, for each category, we should consider the amount of the size related returns of an article compared to the amount of the returns in its category over the same time period.

Therefore considering the above points, if an article has higher size related returns than the average of its category over the same period of time, then the article is considered to demonstrate a sizing problem (labeled as class 1); otherwise, it is considered to have a normal sizing characteristics and thus belongs to the no-sizing-issue class (labeled as class 0).

For each article and category our confidence in labeling the article as a size issue or not greatly depends on how large the number of sales and returns are. Therefore, we propose to use a binomial likelihood $\mathcal{L}$ to assess the confidence in the class assertion. Let's denote p the expected size related return rate of the item, i.e. the size related return rate of its category, k the number of size related returns of the item, and n the number of purchases. We can define the binomial likelihood as following:

$$\mathcal{L} = \binom{n}{k} p^k (1-p)^{n-k} \tag{1}$$

We note that the value of the likelihood is maximized when the ratio of k over n is equal to p. In other words, k is the expected number of size related returns sampled from the distribution p, when drawing n times. The more observations are sampled, the more the estimator is confident. That way, for a large value of n, if the ratio k over n diverges from p, the likelihood is low. Conversely, if only few observations have been sampled, the estimator is really uncertain and tends to distribute the density over all possible values of k. Let's define a score s based on the negative logarithm of the binomial estimator:

$$s = -\ln(\mathcal{L}) \tag{2}$$

In that way, the score s is very high when k is unlikely to have been sampled from p, meaning that the size related return rate is either very high (sizing problem, class $y = 1$) or very low (no size issue, class $y = 0$). Figure 3 shows the behaviour of s with respect to n and k. In this Figure, as an example, let's assume that the

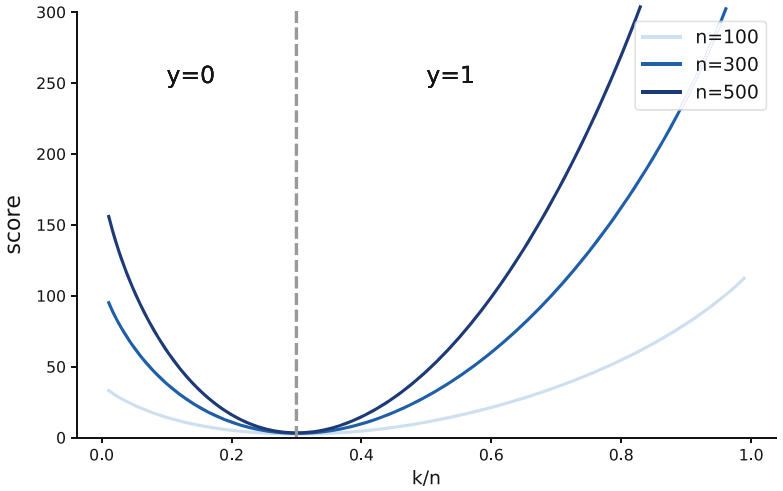


Fig. 3 Y-axis is the value of the score s. X-axis is the ratio of the number of size related returns k over the number of sales n. Curves are plotted for different amounts of sales n. In this example the expected size return rate p is arbitrarily set at 0.3 for illustration (vertical dashed gray line)

expected size return rate p for a defined category and a fixed sales period is 0.3 (vertical dashed gray line). Therefore, articles in this category and in the same sales period, for which the size related returns is larger than 0.3 (right side of the line) are considered to demonstrate a sizing problem (labeled as $y = 1$). On the other hand, for the same ratio of k over n, we see how an increase in number of purchases n (different U shape blue curves) results in an increase in the score s, and thus, demonstrating a better confidence in the class assertion.

To get a better understanding of the score function s, we can look at the asymptotic interpretation of the Equation 2. By applying the Stirling approximation, we can easily derive the following property:

$$s \rightarrow nKL(\bar{P}||P) \text{ when } n \rightarrow \infty \tag{3}$$

where $P = \{p, 1-p\}$ and $\bar{P} = \{\frac{k}{n}, 1-\frac{k}{n}\}$, and KL denotes the Kullback-Leibler divergence [28]. This property provides a better understanding of the behaviour of the confidence score: if $\bar{P}$ is very different from P, i.e. if the size related return rate is either way lower or way higher than the one of its category, then the Kullback-Leibler divergence is high and s is high too. However, if the number n of purchases of the article is low, the score is penalized.

The negative log likelihood – as well as the Kullback-Leibler divergence – are defined on $\mathbb{R}^+$, and consequently can in theory tend to the infinity. In practice, we can however define upper bounds for the score s. Upper bounds are reached when p is very different from the ratio k over n, i.e. in the following two cases:

- when $p \to 0^+$ and $k = n$, then $s = -n \ln(p)$
- when $p \to 1^-$ and $k = 0$, then $s = (k - n) \ln(1 - p)$

Note that the cases $p = 0$ and $p = 1$; that is when the size related return rate of the category is zero, or in contrast, when all items are returned, define very interesting edge cases. The first usually happens in the few weeks that follow the activation of the articles on the e-commerce platform, where no returns are recorded yet. Therefore, in this case $p = 0$ implies $k = 0$; since as soon as we record a return for an article, we also record a return for its category. As a consequence, for this case the binomial likelihood is equal to 1, meaning that the confidence score is zero. Therefore, the statistical model is not capable of providing any size issue prediction. It is important to note that this case actually corresponds to the challenging cold start problem in e-commerce fashion for which we propose a solution in this paper thanks to our SizeNet approach. The latter case where $p = 1$ implies $k = n$, and the confidence score is also zero. However, this scenario, in which all articles are returned due to size issues, is practically non-existent in the e-commerce context.

Now that we have established our statistical model as the teacher, capable of providing sizing class labels with a confidence score, we discuss the student for learning of visual cues for size issues following a curriculum learning framework, keeping in mind the generalization to the cold start articles.

3.3 SizeNet: Learning Visual Size and Fit Cues

In this section, we propose the SizeNet architecture to investigate the article size and fit characteristics in a weakly-supervised teacher-student manner using fashion images. We make use of the labels and their confidence scores acquired from the statistical model described in the previous section, to teach the image-based SizeNet model size issue classification. In particular, we adopt a curriculum learning approach that gradually makes use of feeding the articles with high size issue confidence scores for learning confident visual representations for sizing issues in the images followed by less confident samples to improve generalization. Figure 2 illustrates the architecture of our approach including the statistical model, and the proposed SizeNet composed of a CNN backbone feature extractor followed by a multi-layer perceptron.

3.3.1 Backbone Feature Extractor

We employ the Fashion DNA (fDNA) [7] network as a backbone features extractor for SizeNet. The adopted flavour of fDNA is similar to the ResNet [29] architecture except for the prediction layer. fDNA models are trained with the aim of predicting limited fashion article metadata such as categorical attributes, gender, age group, commodity group, and main article color. We use pre-trained network weights on

1.33 million fashion articles (sold from 2011 to 2016). With the fDNA backbone, we are able to extract for each image a bottleneck feature vector of dimension 128.

3.3.2 Multi-layer Perceptron

On top of the backbone network, we attach a multi-layer perceptron (MLP) that consists of four fully-connected layers. We opt for a bottleneck MLP approach [30] going up from the 128 extracted feature vector, to 256 units and down again to 128. Therefore, the numbers of units of the four fully connected layers are respectively 256, 128, 64 and 32. Each of these layers is followed by a non-linear ReLU activation function. To avoid over-fitting on the training data, we use standard dropout layers for each fully connected layer. The output layer has a sigmoid activation with a unit indicating the sizing issue. We use a binary cross-entropy loss function and optimize the network weights through stochastic gradient descent (SGD).

We adopt a curriculum that gradually trains the SizeNet starting from more confident samples, coming from the statistical model, down to less confident samples. To prepare the loss function for samples where the label confidence from the statistical model is low, we propose to use a weighting function in the loss. Let's define a sample weight w_i using logarithmic transformation of the sample confidence score s_i as follow:

$$w_i = \ln(1 + s_i) \tag{4}$$

The logarithmic transformation allows us to reduce the skewness in the confidence score distribution and provides numerically well-behaving weights compared to the scores. Putting it all together, given image and class pairs $x_i \in X$ and $y_i \in Y$ from a training set $S = \{(x_i, y_i), i = 1..n\}$ with y_i being the class label produced by the teacher model and a backbone network $\phi : X \to \tilde{X}$ as image features extractor. The goal of the student is to learn a function $f : \tilde{X} \to Y$ using the following objective function:

$$\min_{\theta} \sum_{i=1}^{n} w_i L(y_i, f(\phi(x_i); \theta)) \tag{5}$$

where L is binary cross-entropy loss of predicting $f(\phi(x_i))$ given the teacher label y_i and w_i is the sample confidence score from Equation 4.

Once the network has been fully trained, we evaluate the performance using unseen test data in the next section. We analyze on cases that extend from a. articles where the statistical model provides quality predictions of sizing issue, to b. articles where the statistical model fails to provide quality predictions. The aim of this approach is twofold: first to see to what extent SizeNet is capable of producing quality results comparable to the statistical model using purely images, and second

to see to what extent SizeNet can generalize its predictions thanks to the learned visual representations, to those unknown, cold start, and low confidence articles.

4 Experimental Results and Discussion

In this section, we conduct multiple experiments to evaluate and understand the performance of the proposed SizeNet model over multiple garment categories from around 500 different brands.

4.1 Dataset

We use a rich dataset of women textile garments purchased between 2017 and 2018 restricted to one European country, from a major fashion e-commerce platform. This dataset is anonymized and is not public due to various privacy challenges and proprietary reasons which lie outside the scope of this work. However, to gain a better understanding of the data, we highlight in the following the important aspects of it related to the sizing problem.

We define observations at the stock keeping unit (SKU) level. This means that two pieces of garments belonging to the same model, but with different colors, are considered as two distinct observations. We justify this choice by two main reasons derived from expert knowledge: 1. manufacturers use different fabrics depending on the dying technique, 2. the customer's perception of size and fit varies depending on garment color. Those two points lead to very different size related return reason distributions for the same article model but with different colors.

Teacher dataset Our teacher model has access to privileged information on historical articles, specifically, purchase and return data. We only use returns that are due to size issues within the considered period of the time. To ensure the quality of the return data and considering the fact that returned articles require a variable amount of time to be received by the stores, we exclude the articles which have been activated only in the last 6 months of this time period. An overall description of the dataset is presented in Table 1.

Studying this dataset, we observe different return rates for different categories depending on how challenging the category is in terms of finding the right size. For example, trousers and dress categories both show return rates that are almost two times higher than the return rate of the knitwear category. This supports our strategy laid out in Sect. 3.2 where we compare each article return rate to the mean return rate of its own category.

Student dataset The student model has no access to purchase and return data and depends on article images only. We opt to use packshot images with white

Table 1 Description of different features of the teacher dataset used to learn the statistical classifier

	Purchases	Articles	Brands	Categories
#unique values	30,677,930	127,044	535	12

Table 2 Number of articles per class according to the week labels produced by the teacher model in the student dataset

Class	Articles
Size issue	68,892
No size issue	58,152

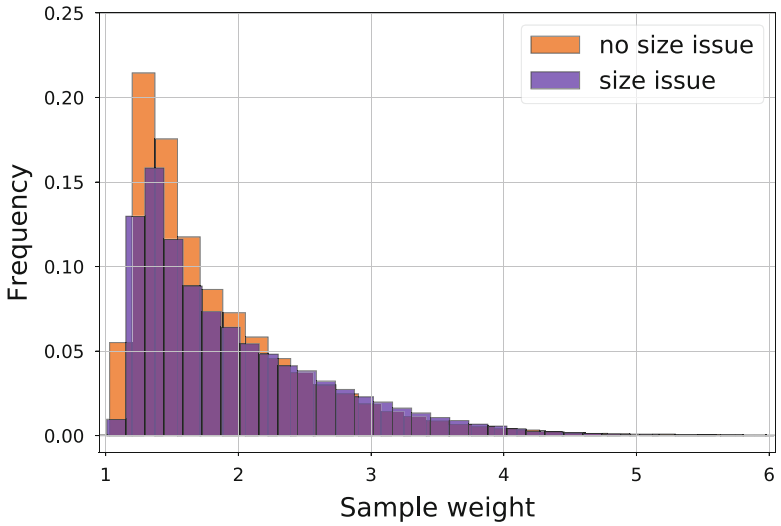


Fig. 4 Histogram representation of sample weight distribution based on teacher confidence score per class in the student dataset

background and without a human model wearing the garment. We do not perform any task-specific image pre-processing, the input images are simply re-sized to (177×256). As described in Sect. 3.1, the dataset is weekly annotated by the teacher classifier. Table 2 reports number of articles in the resulting week labels and Fig. 4 plots a histogram representation of the sample weight distribution acquired from the teacher confidence score, showing relatively balanced dataset between articles with size issue and articles with no size-issue.

For the following experiments, the dataset was split by maintaining a ratio of 60/20/20 for training, validation, and test sets respectively. We cross-validate hyper-parameters of the network, such as start learning rate, batch size, number of epochs, and stopping criteria using the validation set.

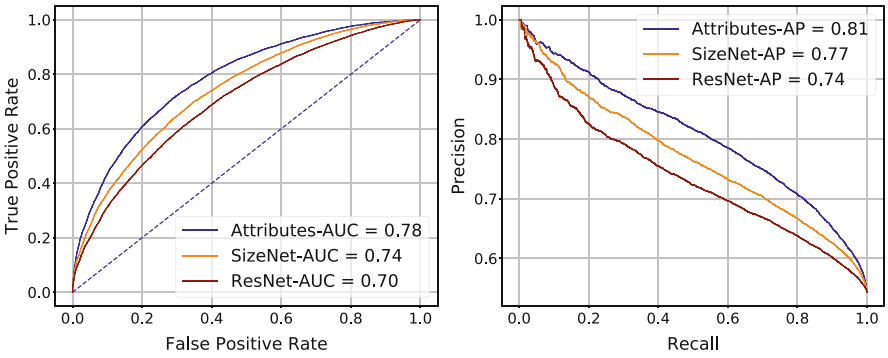


Fig. 5 Evaluation of size issue prediction for the overall dataset (12 categories) comparing SizeNet to two baselines. Left: Receiver Operating Characteristic (ROC) curves with area under curve (AUC); Right: Precision-Recall curves with average precision (AP)

4.2 Evaluation

In order to assess the performance of our student model, we first study the classification metrics including the receiver operating characteristic (ROC) and precision-recall curves. Then, we evaluate the importance of using the suggested sample weighting acquired from the teacher confidence score on the student performance.

4.2.1 Baselines

Here we introduce two baselines: first baseline is a model denoted as `Attributes` that instead of article image uses sparse k-hot encoding vector of binary fashion attributes (e.g. fabric material, fit type, length, etc.) of size 13,422. These attributes are created following a laborious and costly process by human expert annotators. As a second baseline, we use a standard `ResNet` pre-trained on ImageNet as the backbone CNN instead of fDNA. We report the results for the overall size issue predictions (12 categories combined). Figure 5 demonstrates that `SizeNet` outperforms `ResNet` baseline, and achieves promising results compared to that of `Attributes` model which requires tremendous annotation effort. This benchmark establishes the value of the `SizeNet` purely using image data. Figure 6 presents SizeNet performance per category curves for the four major garment categories: dresses, trousers, knitwear, and tops/blouses, where for each category more than 2000 articles are present in the test set. From these curves we can observe good results for SizeNet predictions; in particular prediction of size issues in dress and trouser categories outperforms other categories.

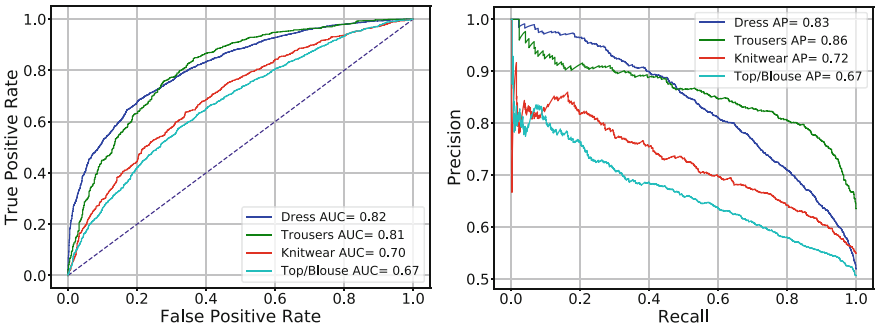


Fig. 6 Evaluation of size issue prediction for the four major categories. Left: Receiver Operating Characteristic (ROC) curves with area under curve (AUC); Right: Precision-Recall curves with average precision (AP)

4.2.2 Weight Importance

Here, we investigate how the teacher and the student interact with each other. As mentioned in Sect. 3, the neural network (the student) learns the image based size issue predictions from the output of a statistical model (the teacher) that has access to privileged sales and returns data. Samples are weighted to favor regions in the parameter space where the certainty of the teacher is maximal. As a result, we expect to observe good predictions from the student for samples where the teacher is confident. To verify this hypothesis, we plot in Fig. 7 the accuracy of the SizeNet model with respect to different values of a threshold τ applied on the weights w_i which correspond to a monotonous transformation of class confidences from the statistical model. Figure 7 (left) shows the overall accuracy (12 categories) on the test set, obtained both with and without sample weighting during the training phase. Figure 7 (right) shows per category accuracy for the four major categories using sample weighting in the training phase. Low values of τ correspond to all articles particularly including those that suffer from the cold start problem. With higher values of τ, only those articles which are not suffering from the cold start problem are considered (higher confidence in the class). As expected, the curve shows a high correlation between the SizeNet model performance and the confidence level based on the binomial estimator.

With regards to the added value of the weighting during the training phase, from Fig. 7 (left) we observe that the performance of both cases follow the same trend, though for lower values of τ, using the weights in the training phase improves the performance on the test set in particular for the cold start articles. For high values of τ, results do not provide much insights since the variance is too high (caused by too few samples). The algorithm exploiting weights is relaxed around the decision boundary in agreement with the study from [23], leading the model to provide a better generalization capacity.

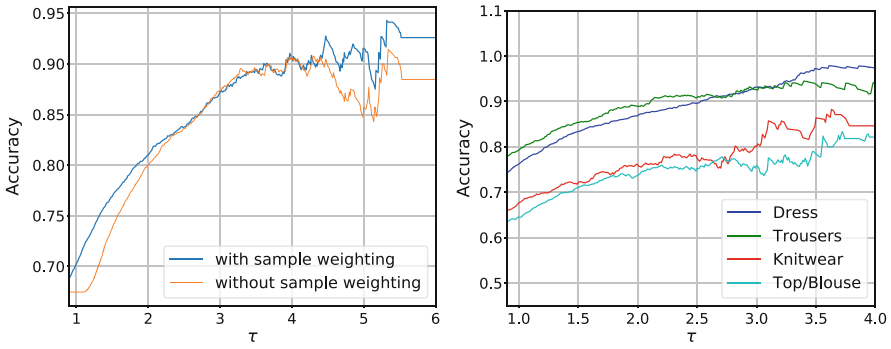


Fig. 7 Accuracy of SizeNet model for different thresholds τ on the test set. Lower values of τ correspond to including cold start articles, where an increase in τ corresponds to only considering articles with larger sales and returns (Left) Overall accuracy with and without using the sample weights w_i in the training phase. (Right) Per category accuracy for major categories using sample weighting in the training phase

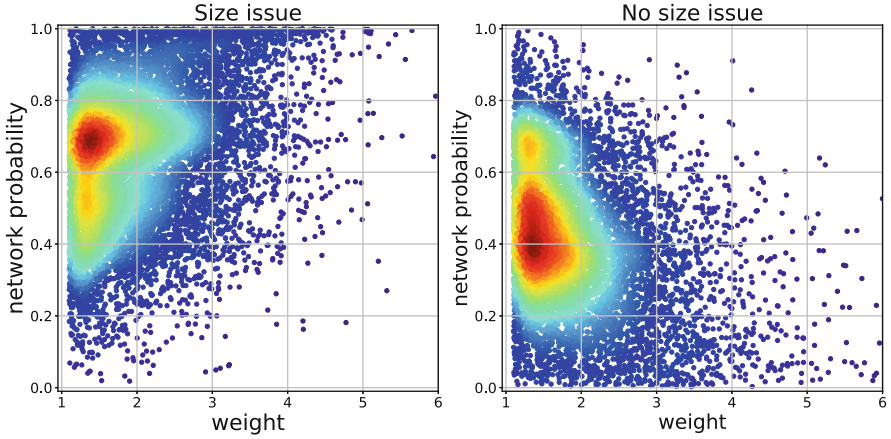


Fig. 8 SizeNet output probability vs. statistical weights w_i: Y axis is the output prediction of SizeNet. X axis is the weights w_i from the statistical model corresponding to a monotonous transformation of class confidences. Left plot is for class 1 (size issue) and right plot is for class 0 (no size issue)

As referred earlier, one of the added values of SizeNet is its capability to tackle the cold start problem using only images, while ensuring good performance for cases where return data is enough to accurately predict size related issues. To get a better understanding of the relation between the sample weights and the outcome of the neural network, we plot the output of the network as a function of the weights w_i (a monotonous transformation of the confidence score) in Fig. 8 showing the density as a heatmap. In both plots, dots are distributed like triangles. Let us focus on the four corner regions of the left plot (class size issue) in Fig. 8:

- Upper right corner: the network outputs a value close to 1 (sizing problem) and the statistical weight is high, meaning that the teacher is very certain of the size issue. The dots in that area confirm that the student has learned accurately from the teacher.
- Upper left corner: the network outputs a value close to 1 (sizing problem) but the weight is low, meaning that the teacher is unsure of the class. This is the interesting case where the student correctly predicts the class, thanks to the learned visual cues, whereas the teacher fails due to lack of historic data – this region mainly corresponds to the cold start problem.
- Lower left corner: the network misclassifies samples for which the teacher is not certain of the class. Though we would prefer avoiding misclassification, those samples are next to the decision boundary where we expect disagreements between the teacher and the student.
- Lower right corner: the network misclassifies samples for which the teacher is very certain of the class. No points are observed in this region that would indicate a strong disagreement between the teacher and the student.

Similar observations can be made from the right plot in Fig. 8, corresponding to the class 0 (no size issue). Following this analysis we observe that SizeNet is capable of learning and replicating the knowledge of the teacher without direct access to the privileged data. In cold start cases, the learned cues can even help the student to make a more informed decision compared to that of the teacher.

4.3 Brand Size Issue Scoring

In this section, we briefly investigate large-scale brand analysis using SizeNet to determine how articles from different brands behave from size and fit issue perspective. We perform this study on the test dataset composed of around 500 different brands that are fully anonymized in this publication due to privacy and business reasons. In Fig. 9 we present SizeNet ranking of top 40 brands with most sizing issues for each category separately. We notice that the ranking matches our intuition that some categories are more challenging than others. For example, trousers and dresses tend to be more difficult in terms of finding the right size when compared to knitwear and tops. This is reflected in Fig. 9 by the model ranking, where brand size issue scores are relatively higher for dress and trousers categories in comparison to the other categories. This experiment highlights the advantage of SizeNet in this context, where we can perform large-scale analysis of brands products for size and fit issues even at the prototyping stage before mass production. In a longer term, such insights may be of great value for delivering improved products and customer satisfactions for each brand.

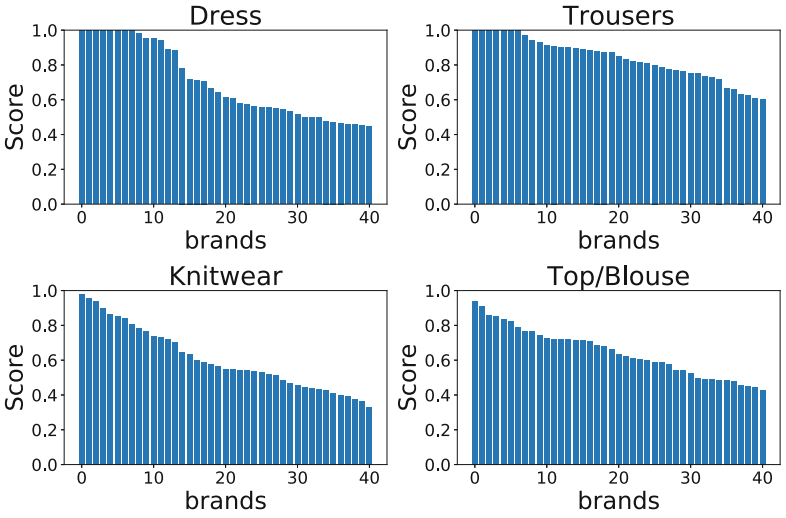


Fig. 9 Brand size issue scores: we calculate a brand size issue score according to the share of its size issue articles among all its articles within the category

4.4 Visualization of Size Issue Cues

In the spirit of explainable AI, and to better understand the SizeNet predictions from fashion images, in this section we follow the recent methodology proposed by [31] called randomized input sampling for explanation of black-box models (RISE). We randomly generate masked versions of the input image and obtain the corresponding outputs of the SizeNet model to assess the saliency of different pixels to the prediction. Therefore, estimated importance maps of image regions can be generated for size issue predictions in different garment categories.

In Fig. 10 we show the highest ranked true positives (top) and false positives (bottom) for size issues from different categories. It should be recalled, SizeNet was trained without any size and fit related image segment-annotations or human expert-labels. Overall from Fig. 10 we observe, for true positives, localized heatmaps attached to specific regions of the clothes, whereas for false positives we observe more expanded heatmaps covering large portions of the images. When looking closer, we can speculate that SizeNet predicts the following size issues for the highest ranked true positive articles; chest area for the evening dress, sleeves for the leather jacket, the length of the wedding dress, and areas of the trousers that may indicate too tight fit. On the other hand, when considering the top ranked false positives, we can observe that SizeNet misclassifies the pink top and the loose trousers based on regions of the article that are not related to size issues.

Now that we have generated explanations for different samples, we would like to build an intuition on the model size issue prediction for different categories. Thus, we choose dress and trousers as the two most dominant categories in our dataset.

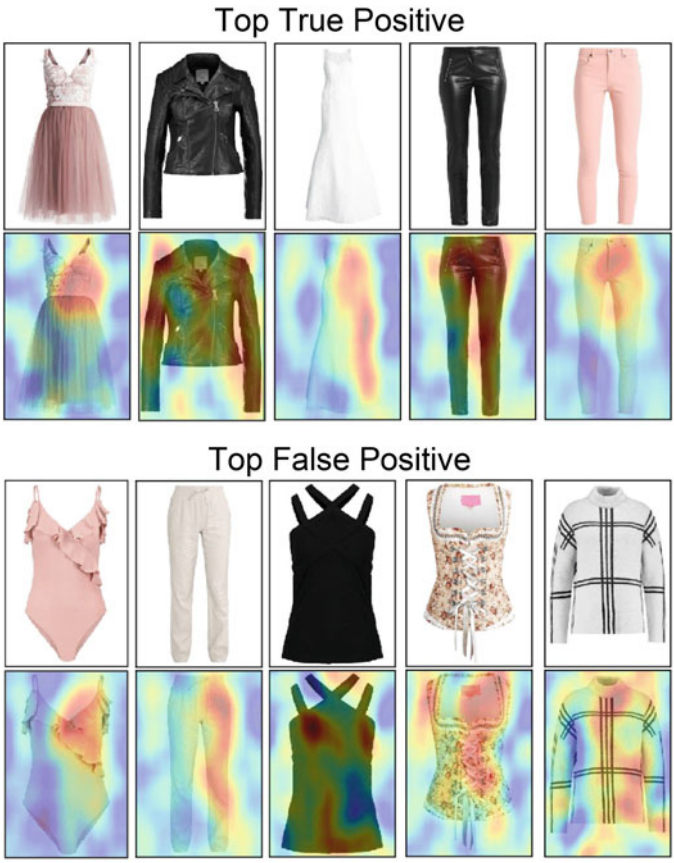


Fig. 10 Explanations for SizeNet model predictions: importance maps showing the effect of different image regions on the model predictions for the top five true positives (top rows), and the top five false positive predictions (bottom rows)

In Fig. 11, we localize the important regions that activate the size issue for each category by averaging the heatmaps of true positive examples. Our first observation is that generally we get heatmaps focused around the same spot where the garments are usually centered in the image. We notice a slight shift towards the right side, this can be explained by the symmetric design of most garments and so the model has learned to focus only on the right side. Interestingly, for dress category, the concentration suggests that the upper part of the garment is the most important region. Whereas for trousers, it looks like the model focuses along the garment leg length.

The false positive examples and the categories explanation can provide qualitative insights into the complexity of size and fit in fashion and show limitations of our approach which in its current implementation does not take into account any information on the style of fashion articles. In future work we aim to validate

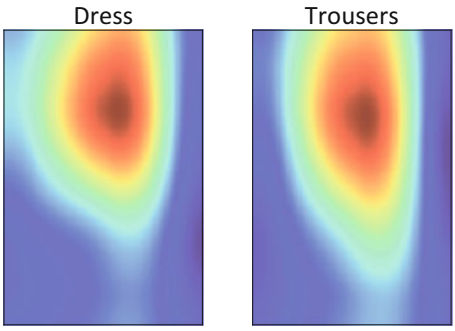


Fig. 11 Averaged heatmaps of model explanations to size issue prediction that highlights important regions for dress and trousers categories

or reject these observations either by analyzing customer reviews or by including region based expert size issue annotations.

5 Conclusion

The potential of fashion images for discovering size and fit issues was investigated. A weakly-supervised teacher-student approach was introduced where a CNN-based architecture called SizeNet, acts as a student, learns visual sizing cues from fashion images thanks to a statistical model, acting as a teacher, with privileged access to articles sales and returns data. Quantitative and qualitative evaluation was performed over different categories of garments including dresses, knitwear, tops/blouses, and trousers for both warm and cold start scenarios. It was demonstrated that fashion images in fact contain information about article size and fit issues and can be considered valuable assets in tackling the challenging cold start problem. A large-scale size and fit quality metrics was calculated for brands using SizeNet, potentially already at the prototyping stage before mass production, which can in turn result in improved products and customer satisfaction. Future work consists of including expert-labeled data, evaluating the generalization capacities of SizeNet to fashion images in the wild, and multi-task learning for SizeNet using fit style taxonomies. Also, further evaluation of size issue explanations derived from SizeNet is necessary to understand, on one hand, to what extent these weakly-learned localized explanations (i.e. tight shoulders, long sleeves) correspond to the actual customer experience, and on the other hand, how these explanations may be used in the future to visually support retail customers in their purchase decision making.

References

1. Pisut G, Connell LJ (2007) Fit preferences of female consumers in the USA. J Fash Market Manag Int J 11(3):366–379

2. Ujević D, Szirovicza L, Karabegović I (2005) Anthropometry and the comparison of garment size systems in some European countries. Coll Antropol 29(1):71–78
3. Shin S-JH, Istook CL (2007) The importance of understanding the shape of diverse ethnic female consumers for developing jeans sizing systems. Int J Consum Stud 31(2):135–143
4. Faust M-E, Carrier S (2014) Designing apparel for consumers: the impact of body shape and size. Woodhead Publishing
5. Hu Y, Yi X, Davis LS (2015) Collaborative fashion recommendation: a functional tensor factorization approach. In: Proceedings of the 23rd ACM international conference on multimedia. ACM, pp 129–138
6. Arora S, Warrier D (2016) Decoding fashion contexts using word embeddings. In: Workshop on machine learning meets fashion, KDD
7. Bracher C, Heinz S, Vollgraf R (2016) Fashion DNA: merging content and sales data for recommendation and article mapping. In: Workshop machine learning meets fashion, KDD
8. Ferreira BQ, Baía L, Faria J, Sousa RG (2018) A unified model with structured output for fashion images classification. In: Workshop on machine learning meets fashion, KDD
9. Liu Z, Luoa P, Qiu S, Wang X, Tang X (2016) Deepfashion: powering robust clothes recognition and retrieval with rich annotations. In: Conference on computer vision and pattern recognition (CVPR)
10. Gutierrez P, Sondag P-A, Butkovic P, Lacy M, Berges J, Bertrand F, Knudsong A (2018) Deep learning for automated tagging of fashion images. In: Computer vision for fashion, art and design workshop in European conference on computer vision (ECCV)
11. Di W, Wah C, Bhardwaj A, Piramuthu R, Sundaresan N (2013) Style finder: fine-grained clothing style detection and retrieval. In: Workshop in conference on computer vision and pattern recognition (CVPR)
12. Nakamura T, Goto R (2018) Outfit generation and style extraction via bidirectional LSTM and autoencoder. In: Workshop machine learning meets fashion, KDD
13. Hadi Kiapour M, Han X, Lazebnik S, Berg AC, Berg TL (2015) Where to buy it: matching street clothing photos to online shops. In: International conference on computer vision (ICCV)
14. Lasserre J, Rasch K, Vollgraf R (2018) Studio2shop: from studio photo shoots to fashion articles. arXiv preprint :180700556
15. Mohammed Abdulla G, Borar S (2017) Size recommendation system for fashion e-commerce. In: Workshop on machine learning meets fashion, KDD
16. Guigourès R, Ho YK, Koriagin E, Sheikh A-S, Bergmann U, Shirvany R (2018) A hierarchical Bayesian model for size recommendation in fashion. In: Proceedings of the 12th ACM conference on recommender systems. ACM, pp 392–396
17. Sembium V, Rastogi R, Saroop A, Merugu S (2017) Recommending product sizes to customers. In: Proceedings of the Eleventh ACM conference on recommender systems. ACM, pp 243–250
18. Sembium V, Rastogi R, Tekumalla L, Saroop A (2018) Bayesian models for product size recommendations. In: Proceedings of the 2018 world wide web conference, WWW '18, pp 679–687
19. Lasserre J, Sheikh A-S, Koriagin E, Bergman U, Vollgraf R, Shirvany R (2020) Meta-learning for size and fit recommendation in fashion, pp 55–63
20. Sheikh A-S, Guigourès R, Koriagin E, Ho YK, Shirvany R, Vollgraf R, Bergmann U (2019) A deep learning system for predicting size and fit in fashion e-commerce. In: Proceedings of the 13th ACM conference on recommender systems, pp 110–118
21. Zhou Z-H (2017) A brief introduction to weakly supervised learning. Natl Sci Rev 5(1):44–53
22. Vapnik V (2013) The nature of statistical learning theory. Springer Science & Business Media
23. Vapnik V, Izmailov R (2015) Learning using privileged information: similarity control and knowledge transfer. J Mach Learn Res 16:2023–2049
24. Wong JHM, Gales MJ (2016) Sequence student-teacher training of deep neural networks
25. Bengio Y, Louradour J, Collobert R, Weston J (2009) Curriculum learning. In: Proceedings of the 26th annual international conference on machine learning. ACM, pp 41–48

26. Mikolov T, Chen K, Corrado G, Dean J (2018) Efficient estimation of word representations in vector space. In: Workshop in international conference on learning representations (ICLR)
27. Lopez-Paz D, Bottou L, Schölkopf B, Vapnik V (2015) Unifying distillation and privileged information. arXiv preprint :1511.03643
28. Kullback S, Leibler RA (1951) On information and sufficiency. Ann Math Stat 22(1):79–86
29. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), pp 770–778
30. Vu NT, Metze F, Schultz T (2012) Multilingual bottle-neck features and its application for under-resourced languages. In: Spoken language technologies for under-resourced languages
31. Petsiuk V, Das A, Rise SK (2018) Randomized input sampling for explanation of black-box models. arXiv preprint :1806.07421

Part V
Generative Outfit Recommendation

Generating High-Resolution Fashion Model Images Wearing Custom Outfits

Gökhan Yildirim, Nikolay Jetchev, Roland Vollgraf, and Urs Bergmann

Abstract Visualizing an outfit is an essential part of shopping for clothes. On fashion e-commerce platforms, only a limited number of outfits are visually represented, as it is impractical to photograph every possible outfit combination, even with a small assortment of garments. In this paper, we broaden the set of articles that can be combined into visualizations by training two Generative Adversarial Network (GAN) architectures on a dataset of outfits, poses, and fashion model images. Our first approach employs vanilla StyleGAN that is trained only on fashion model images. We show that this method can be used to transfer the style and the pose of one randomly generated outfit to another. In order to control the generated outfit, our second approach modifies StyleGAN by adding outfit/pose embedding networks. This enables us to generate realistic, high-resolution images of fashion models wearing a custom outfit under an input body pose.

Keywords Generative adversarial networks · Fashion · Outfit visualization · Style transfer

1 Introduction

In recent years, advances in Generative Adversarial Networks (GANs) [2] enabled sampling realistic images via implicit generative modeling. This development established new avenues in visual design and content creation, especially in fashion, where visualization is a key component.

One explicit application of GANs in fashion is to design garments. In [14], a GAN is trained to generate stylized articles with an input shape mask and a latent vector that governs the texture. This idea is extended in [17], where diverse images of dresses are created through disentangled control of their color, shape, and texture.

G. Yildirim (✉) · N. Jetchev · R. Vollgraf · U. Bergmann
Zalando Research, Berlin, Germany
e-mail: Gokhan.Yildirim@zalando.de; Nikolay.Jetchev@zalando.de; Roland.Vollgraf@zalando.de; Urs.Bergmann@zalando.de

N. Dokoohaki (ed.), *Fashion Recommender Systems*, Lecture Notes in Social Networks, https://doi.org/10.1007/978-3-030-55218-3_7

GANs can be used to create personalized visual content, such as rendering an outfit on a human body [18], which can enrich shopping experience on e-commerce platforms. Recent approaches tackle different aspects of the outfit visualization. In [11] and [13], an input fashion model image and a target pose are used to render the fashion model from a different perspective. One can redress [19] the fashion model by using a text input describing the new garment. In another approach, an outfit can automatically be modified according to fashionability criteria [5]. In [10] they focus on generating low-resolution fashion model images by using pose and color as input conditions. These methods implicitly visualize an outfit and do not render a real input article.

Previous studies that generate the visualizations of a real article focus on replacing a single garment of a fashion model. In [7], this is performed by training a conditional GAN architecture on matching and unmatching model/article pairs. Whereas in [3], an input article image is warped to align with a body pose so that a fashion model image can be redressed. Current approaches do not address the general problem of visualizing a set of real and diverse articles (i.e. an outfit that is composed of jackets, shirts, trousers, shoes, accessories etc.) on a human body.

In this paper, we propose a complete solution that concentrates on generating high-resolution images of fashion models wearing desired outfits and under given poses. We employ and modify StyleGAN [9], which builds on the idea of generating high-resolution images using Progressive GAN [8] by modifying it with Adaptive Instance Normalization (AdaIN) [6]. We use a dataset of fashion model-outfit-pose images under two settings: We first train the vanilla StyleGAN on a set of fashion model images and show that we can transfer the outfit color and body pose of one generated fashion model to another. Second, we modify StyleGAN to condition the generation process on an outfit and a human pose. This enables us to rapidly generate high-fidelity fashion model images wearing custom outfits under different body poses and types. Preliminary results from this work were presented at [18]. However, in this extended paper, we have included a more thorough analysis and an in depth discussion.

2 Outfit Dataset

We use a proprietary image dataset with around 380K entries. Each entry in our dataset consists of a fashion model wearing an outfit with a certain body pose. An outfit is composed of a set of 6 articles (a jacket, two upper body garments, one pair of trousers, one pair of shoes, and an accessory). If an outfit is missing a certain garment type, it is represented with an empty gray image. In order to obtain the body pose, we extract 16 keypoints using a deep pose estimator [15]. In Fig. 1, we visualize two samples from our dataset. The red markers on the fashion models represent the extracted keypoints. Both model and articles images have a resolution of 1024×768 pixels.

Fig. 1 Samples from our image dataset. The red markers on the fashion model images represent the extracted keypoints. An empty gray image is used for missing garments

3 Methods

Our goal is not only to generate realistic high-resolution images of fashion models, but also to control the rendered outfit and body pose. Therefore, we train both vanilla StyleGAN (unconditional) and its modified version (conditional) on our outfit dataset and compare their capabilities.

3.1 Unconditional

The flowchart for the unconditional version is illustrated in Fig. 2. This version is the vanilla StyleGAN [9], which we train only on the fashion model images of our outfit dataset. With this method, we can sample a high-resolution image of a fashion model, but without the ability to control the generated outfit or the pose, both of which are implicitly governed by the input latent vector and consequently by the style vector. The generator has 18 layers that receive a linearly transformed (via a dense layer) version of the style vector for adaptive instance normalization (AdaIN). The generator and discriminator architectures are identical to the original StyleGAN, with the exception of generating/processing images with 4:3 aspect ratio, instead of square images.

3.2 Conditional

In the conditional version, we augment StyleGAN architecture with embedding networks as shown in Fig. 3. Inputs to these networks are the concatenated article

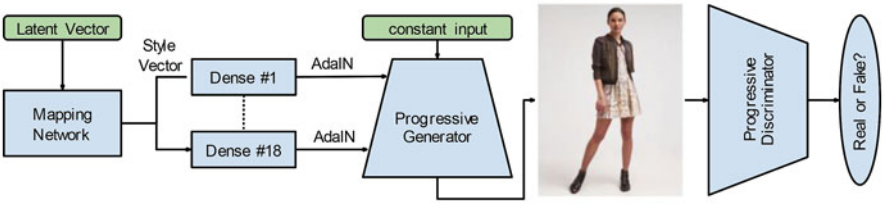


Fig. 2 The flowchart of the unconditional GAN. The style vector is transformed by separate dense (i.e. fully-connected) layers that compute the AdaIN parameters of each generator layer

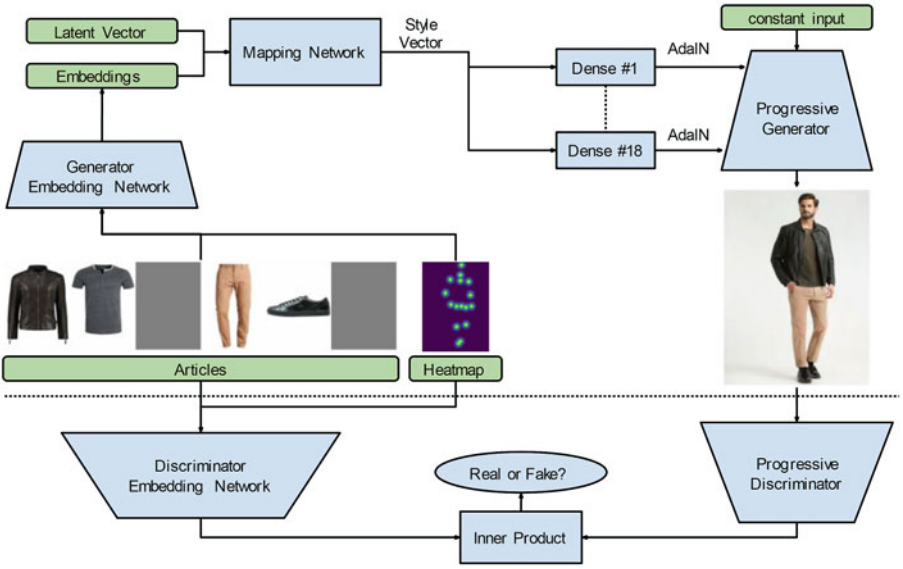


Fig. 3 The flowchart of our conditional GAN. In this model, StyleGAN is augmented with two embedding networks, which create 512-dimensional representations for concatenated input outfits and poses

images (six article images resulting in 18 channels in total) and a 16-channel heatmap image that is computed from 16 keypoints. The article images are concatenated with a fixed ordering for semantic consistency across outfits. We can see this ordering in Fig. 1. The generator embedding network creates a 512-dimensional representation, which is concatenated with the latent variables in order to produce the style vector. The discriminator in the conditional model uses a separate network to compute the embedding for the input articles and heatmaps. Inner product of this embedding and the discriminator output is considered as the final score of an input image [12]. For training stability and GPU memory efficiency, our embedding networks have a fixed input size (i.e. not progressively grown) of 256×192 pixels. These networks have the same architecture as our discriminator at

256 × 192 resolution, except the last layer, which outputs a 512-dimensional vector, instead of a single score.

4 Experiments

We train both of our GANs for around four weeks on four NVIDIA V100 GPUs, resulting in 160 and 115 epochs on unconditional and conditional cases, respectively. In the unconditional case, the discriminator has seen around 61 million real fashion model images, with a batch size of 256 at the beginning of the training and then gradually reduced to 16, to accommodate for progressive growing of neural networks. On the other hand, the conditional case has seen around 44 million fashion model images, where the batch size varies from 128 to 8. The batch sizes are smaller than that of the unconditional case, as we have the embedding networks, which occupy additional GPU memory.

4.1 Unconditional

In Fig. 4, we illustrate images that are generated by the unconditional model. As we can see, not only the articles and accessories (e.g. handbag in the fourth image), but also the human body parts are realistically generated at the maximum resolution of 1024 × 768 pixels. In our unconditional training, lack of semantics can sometimes lead to particular errors, such as inconsistent shoe colors and incorrectly rendered occluded body parts in the fifth and sixth image in Fig. 4, respectively.

During the training, one can regularize the generator by switching the style vectors for certain layers. This has the effect of transferring information from one generated image to another. In Fig. 5, we illustrate two examples of information transfer. First, we broadcast the same source style vector to layers 13 to 18 (before the dense layers in Fig. 2) of the generator, which transfers the color of the source outfit to the target generated image, as shown in Fig. 5. If we copy the source style vector to earlier layers, this transfers the source pose.

In Table 1, we show which layers we broadcast the source and the target style vectors to achieve the desired transfer effect. In order to analyze our generator, we calculate the AdaIN vectors (i.e. style vector that is transformed by a dense layer) at each layer for 20 K generated images. We then visualize the nearest neighbors of a generated query image at different network depths as shown in Fig. 6. We can see that at layer #1, the neighboring generated images have similar body poses. On the other hand, at layer #14, color/style of the outfit is more prevalent. This is consistent with the layers we use to create transfer effects in Table 1.

Fig. 4 Model images that are generated by the unconditional StyleGAN

Fig. 5 Transferring the colors or a body pose from one generated fashion model to another by using the unconditional GAN

Table 1 The layer numbers indicate how to broadcast the style vector to achieve either color or pose transfer

	Color transfer	Pose transfer
Source	13–18	1–3
Target	1–12	4–18

Fig. 6 A query image (first column) and its five nearest neighbors, which are obtained via computing the Euclidean distance between the AdaIN vectors at different layers

4.2 Conditional

After training our conditional model, we can input a desired set of articles and a pose to visualize an outfit on a human body as shown in Fig. 7. We use two different outfits in Fig. 7a, b, and four randomly picked body poses to generate model images in Fig. 7c, d, respectively. We can observe that the articles are correctly rendered on the generated bodies and the pose is consistent across different outfits. In Fig. 7e, we visualize the generated images using a custom outfit by adding the jacket from the first outfit to the second one. We can see that the texture and the size of the denim jacket are correctly rendered on the fashion model. In certain body poses, where a hand can be rendered in or out of pockets, our generator can get confused and produce uncanny human body parts (see Fig. 7e, third image). Note that, due to the spurious correlations within our dataset, the face of a generated model might vary depending on the outfit and the pose.

In our dataset, we have fashion models with various body types that depend on their gender, build, and weight. This variation is implicitly represented through the relative distances between the body keypoints. Our conditional model is able to capture and reproduce fashion models with different body types as shown in the fourth generated images in Fig. 7. This result is encouraging, and our method might be extended in the future to a wider range of customers through virtual try-on applications.

Fig. 7 Two different outfits (**a**) and (**b**) are used to generate model images in (**c**) and (**d**). (**e**) The jacket from outfit #1 is added to outfit #2 to customize the visualization

Fig. 8 For each row, a set of input articles and a body pose (represented as a heatmap) are used to generate the image of a fashion model in the last column

In Fig. 8, we illustrate additional renderings of fashion models. Overall, our conditional method captures and represents the details of the input articles on a human body. Articles with complex textures, such as the pair of trousers on the fourth outfit, may not be accurately rendered due to their unique visual look. The articles in each outfit is semantically ordered for consistent training. However, some garment types, such as dresses, are ambiguous (i.e., they are both top and bottom garments) and therefore can sometimes be incorrectly rendered, as shown in the fifth row of Fig. 8.

Table 2 FID score for the models

	FID score	Training epochs
Unconditional	5.15	115
Conditional	9.63	115

4.3 Quantitative Results

We measure the quality of the generated images by computing the Fréchet Inception Distance (FID) score [4] of the unconditional and conditional GANs. The FID score is calculated by using InceptionV3 [16] feature vectors of both real and generated images, which has seen different statistics (i.e. ImageNet [1]) as the one targeted here. However, we believe it provides a good proxy on the similarity between the distributions of real and generated images. As we can see from Table 2, the unconditional GAN produces higher quality images, which can be observed by comparing Figs. 4 and 7. The conditional discriminator has the additional task of checking whether the input outfit and pose are correctly generated. This might cause a trade-off between image quality (or 'realness') and the ability to directly control the generated outfit and pose.

5 Conclusion

In this paper, we proposed two ways to generate high-resolution images of fashion models. First, we showed that the unconditional StyleGAN can be used to transfer the style/color and the pose between generated images via swapping the style vectors at specific layers. Second, we modified StyleGAN with two embedding networks, so that we can generate images of fashion models wearing a custom outfit with a give pose. As future work, we plan to improve the image quality and consistency of the conditional model on more challenging cases, such as generating articles with complicated textures and text.

References

1. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) ImageNet: a large-scale hierarchical image database. In: CVPR
2. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: NIPS
3. Han X, Wu Z, Wu Z, Yu R, Davis LS (2017) VITON: an image-based virtual try-on network. In: CVPR
4. Heusel M, Ramsauer H, Unterthiner T, Nessler B, Hochreiter S (2017) Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: NIPS
5. Hsiao W, Katsman I, Wu C, Parikh D, Grauman K (2019) Fashion++: minimal edits for outfit improvement. In: ICCV

6. Huang X, Belongie SJ (2017) Arbitrary style transfer in real-time with adaptive instance normalization. In: ICCV
7. Jetchev N, Bergmann U (2017) The conditional analogy gan: swapping fashion articles on people images. In: ICCV Workshops
8. Karras T, Aila T, Laine S, Lehtinen J (2017) Progressive growing of gans for improved quality, stability, and variation. In: ICLR
9. Karras T, Laine S, Aila T (2019) A style-based generator architecture for generative adversarial networks. In: CVPR
10. Lassner C, Pons-Moll G, Gehler PV (2017) A generative model of people in clothing. In: ICCV
11. Ma L, Jia X, Sun Q, Schiele B, Tuytelaars T, Gool LV (2017) Pose guided person image generation. In: NIPS
12. Mescheder L, Geiger A, Nowozin S (2018) Which training methods for gans do actually converge? In: ICML
13. Neverova N, Guler RA, Kokkinos I (2018) Dense pose transfer. In: ECCV
14. Sbai O, Elhoseiny M, Bordes A, LeCun Y, Couprie C (2018) Design: design inspiration from generative networks. In: ICCV
15. Sun K, Xiao B, Liu D, Wang J (2019) Deep high-resolution representation learning for human pose estimation. In: CVPR
16. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: CVPR
17. Yildirim G, Seward C, Bergmann U (2018) Disentangling multiple conditional inputs in gans. In: KDD Workshop on AI for Fashion
18. Yildirim G, Jetchev N, Vollgraf R, Bergmann U (2019) Generating high-resolution fashion model images wearing custom outfits. In: ICCV Workshop on Computer Vision for Fashion, Art and Design
19. Zhu S, Fidler S, Urtasun R, Lin D, Loy CC (2017) Be your own prada: fashion synthesis with structural coherence. In: ICCV