Complementary-Similarity Learning using Quadruplet Network

Mansi Ranjit Mane
mansimane5@gmail.com
Walmart Labs
Sunnyvale, CA, USA

Stephen Guo
sguo@walmartlabs.com
Walmart Labs
Sunnyvale, CA, USA

Kannan Achan
kachan@walmartlabs.com
Walmart Labs
Sunnyvale, CA, USA

ABSTRACT

We propose a novel learning framework to answer questions such as "if a user is purchasing a shirt, what other items will (s)he need with the shirt?" Our framework learns distributed representations for items from available textual data, with the learned representations representing items in a latent space expressing functional complementarity as well similarity. In particular, our framework places functionally similar items close together in the latent space, while also placing complementary items closer than non-complementary items, but farther away than similar items. In this study, we introduce a new dataset of similar, complementary, and negative items derived from the Amazon co-purchase dataset. For evaluation purposes, we focus our approach on clothing and fashion verticals. As per our knowledge, this is the first attempt to learn similar and complementary relationships simultaneously through just textual title metadata. Our framework is applicable across a broad set of items in the product catalog and can generate quality complementary item recommendations at scale.

CCS CONCEPTS

• **Information systems → Recommender systems**; Novelty in information retrieval.

KEYWORDS

neural networks, complementary items, similar items, recommender systems

ACM Reference Format:
Mansi Ranjit Mane, Stephen Guo, and Kannan Achan. 2019. Complementary-Similarity Learning using Quadruplet Network. In *Proceedings of Workshop on Recommender Systems in Fashion, 13th ACM Conference on Recommender Systems (recsysXfashion'19).* ACM, New York, NY, USA, 6 pages.

1 INTRODUCTION

With growing numbers of online purchases and increases in the variety of items available in e-commerce item catalogs, it has become increasingly necessary for e-commerce companies to offer product recommendations on their websites and online channels. One very important area to item recommendations is the suggestion of complementary items. Whether online or offline, when a customer wishes to purchase a top, (s)he often would be interested in first exploring options for tops, before deciding upon which top to purchase. Once (s)he has purchased the top, additional product suggestions for jeans or jackets can be helpful. Complementary item recommendations serve a variety of purposes, such as reminding customers about other relevant complementary items to purchase, enabling catalog product discovery, and encouraging additional purchases and basket expansion.

How can we identify if a pair of items are complementary? In practice, functionally complementary items are often purchased together by users. Historical co-purchase data can be used to learn such item-complementary relations; however, there are some challenges [11]. Co-purchase data is usually available for only a small percentage of items in an item catalog, e.g., Pareto principle [10]. Therefore, modeling approaches primarily relying upon customer interaction data (co-views, co-purchases), such as collaborative filtering, would not properly handle cold-start or low-engagement items.

While learning functional complementary relations, one also needs to learn similarity, to be able to differentiate between complementary items and similar items. If a model is not able to differentiate between the two relations, spurious complementary recommendations can be surfaced. For example, given shoes as an anchor item, one can easily end up recommending different types of other shoes as complementary items, which is not ideal from a customer experience perspective. To address this issue, a common approach is to utilize taxonomy metadata, such as category labels, to differentiate between similar items and complementary items. However, with the increasing number of items and categories available online (more than 600 million on Amazon [15]), it's difficult to label each and every item with fine-grained category information. This process typically requires a large amount of human labeling from domain experts, along with crowdsourcing budget and support.

Recommendations should be also diverse. e.g. if a customer is shopping for a top, reminding customers about a set of jeans, belt, and scarf is more useful to the user than just recommending a set of complementary jeans.

We propose a novel Siamese Quadruplet network to address these issues. Our approach learns latent representations where for a given anchor item, similar items are clustered together. Complementary items are clustered together, but are farther than similar items, while non-complementary and non-similar items (we call them negative items) are placed further apart. We use item title information to generate initial representations.

For this study, we create a dataset of item quadruplets (anchor, similar, complementary, and negative) from the Amazon co-purchase data released by Veit et al. [20]. Using this labeled dataset, we train our proposed Siamese Quadruplet network. Our results indicate that our network is able to differentiate well between similar products and complementary products.

recsysXfashion'19, September 20, 2019, Copenhagen, Denmark

In summary, our contributions are as follows:

- We propose a new Siamese Quadruplet framework which is capable of differentiating between similar, complementary, and negative items.
- We create a dataset of similar, complementary and negative items to evaluate our approach and compare our proposed approach with various baselines.

2 LITERATURE REVIEW

Compatibility Learning: Compatibility Learning methods focus on learning compatibility or style; i.e., given a t-shirt and jeans predict if they go well together, while using mostly visual features to learn the style. Veit et al. [20] utilize Siamese loss to learn compatibility relationships. He et al. [7], Vasileva et al. [18] extend this approach by learning multiple embedding spaces to capture different types of styles. McAuley et al. [12] model user preferences as well, while utilizing low rank Mahalanobis transform to learn style space. Shih et al. [16] describe a generative approach to learn compatibility between items. Han et al. [6] propose a sequential LSTM based approach to learn compatibility by providing sequences from top to bottom. Kang et al. [8] utilize images features from detected bounding boxes of respective items in an image, and then uses attention and siamese-triplet based loss to calculate compatibility between items. For all these approaches, it's assumed that the model has to rank compatibility between items belonging to different categories.

Complementary Learning: The objective of these methods is to learn functional complementary relations. Zhang et al. [23] utilize a multimodal approach (images, text, user ratings) to improve the quality of recommendations. Wan et al. [21] learn content unaware embeddings for users and items to learn both complementary and compatibility relations.

Similarity Learning: These methods focus on learning similarity between items. Grbovic et al. [5] learn similarity space by using a skip-gram model on item sequences from email receipts. Linden et al. [9], Sarwar et al. [14], Wei et al. [22] utilize collaborative filtering with different similarity functions like cosine, Pearson similarity etc. In the computer vision domain, Bell and Bala [1] propose triplet loss for similarity learning. Chen et al. [4], Son et al. [17] describe using quadruplet loss for learning visual similarity. Note that our formulation of quadruplet loss is very different than these approaches.

Complementary-Similarity Learning: McAuley et al. [11] learn similarity and complementary relations by using latent dirichlet allocation (LDA) [2] distributions from review and product description data. Rakesh et al. [13] improve upon this approach by using linked variational autoencoders, while still relying on reviews data. It is important to note that on many e-commerce websites such rich textual data may not be available, even for training models. Title information is easily available, as compared to product reviews and detailed product descriptions. Our quadruplet network utilizes only widely available title information to learn a shared feature space for both complementary and similarity relationships.

3 DATASET GENERATION

To train a Siamese quadruplet network, we require quadruplets of anchor (a), similar (s), complementary (c), and negative (n) items. We utilize the Amazon Clothing, Shoes, and Jewelry dataset released by Veit et al. [20] for anchor, complementary, and negative items. The dataset contains item category information, along with titles and images. For an anchor item and complementary item pair to be functionally complementary, the items need to be from different categories. Therefore, we remove the anchor and complementary pairs, which are from same category. We then generate similar items to each anchor item by sampling items from the same category as the anchor. [1] We have a total of 376,999 unique items and 3,693,416 quadruplets in our dataset. Train and test datasets are generated by splitting quadruplets based on anchor items. The train data has 90% of the anchor items, while the test data has the remaining 10% of the anchor items.

The dataset generation proposed here is different than the one used in McAuley et al. [11], where the train-test split is done based on links. Splitting data based on links results in anchor items during training to appear in the test data as well. We make sure that no anchor item in the test data appears in the train data, making the task more realistic and difficult.

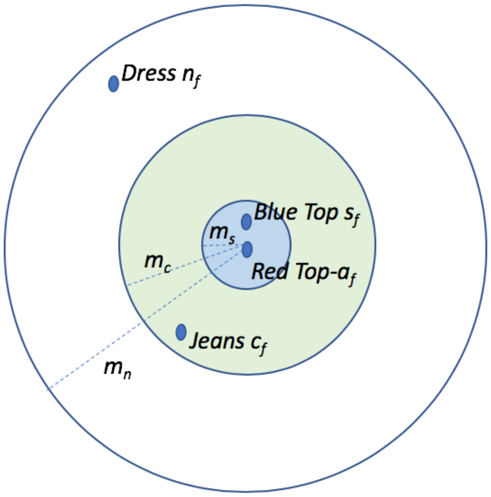

Figure 1: Expected representation space

4 METHODOLOGY

We utilize item title information for given $\{a, c, s, n\}$ quadruplets, as text is more helpful to capture functional complementary relations than images and also has good coverage as compared to other attributes as shown in Table 1. The text feature vectors $\{a_t, c_t, s_t, n_t\}$ are generated from title information using the Universal Sentence

[1] The code and data are available at https://github.com/mansimane/quadnet-comp-sim

Table 1: Coverage - Percentage of items for which attribute information is present

Attribute	Coverage
Item id (asin)	100
Image	99.99
Description	5.68
Title	99.95
Price	38.29
Brand	6.25

Table 2: Notations

Notation	Description
a	Anchor item
c	Complementary item to anchor item
s	Similar item to anchor item
n	Negative item to anchor item
a_t, c_t, s_t, n_t	Text feature vectors for a, c, s, n
a_f, c_f, s_f, n_f	Learnt feature representation for a, c, s, n
a'_f, c'_f, s'_f, n'_f	Normalized learnt feature representation for a, c, s, n

Encoder [3]. We then learn a mapping function f which projects these text feature vectors into representations which model functional complementariness and similarity between items. Our goal here is to learn representations in a space where we can differentiate between similar, complementary and negative items. Referring to Figure 1, the blue top and red top should be close together as both are functionally similar items. Jeans complement tops, so jeans should also be mapped close to the blue top, but not as close as the red top. On the other hand, dresses are not functionally complementary to tops, so should be placed far away from tops. In this study, we denote the representations learnt through the mapping functions as $\{a_f, c_f, s_f, n_f\}$.

4.1 Siamese Networks

Siamese networks have been utilized in the computer vision community to learn similarity between two images. Veit et al. [20] use Siamese triplet networks to learn compatibility between items using visual features. We discuss same framework for learning complementary relation. The loss function of a Siamese triplet network minimizes the distance between the anchor (given) item and complementary item, as compared to the distance between the anchor item and negative item. We denote a_f as the embedding projection for the anchor item, c_f as the embedding for the complementary item, and n_f as the embedding for the negative item. The siamese-triplet loss is given as:

$$L = max(d(a_f, c_f) - d(a_f, n_f) + margin, 0) \quad (1)$$

The margin defines how far away negative items should be from complementary items. This loss is good for similarity learning, however for learning functional complementariness, it has the following challenges:

- When $d(a_f, c_f) = 0$, and $d(a_f, n_f) = margin$ or $d(a_f, n_f) > margin$, the loss is zero. This causes complementary items to project onto the anchor item. For example, if the anchor item is blue jeans and complementary item is belt, we do not want their distance in embedding space to be zero, as this will hinder the ability to differentiate between the current anchor jeans and other jeans; i.e., we do not want $d(a_f, c_f) = 0$ as both are different items.
- Consider the following two cases: case A - when $d(a_f, c_f) = 1.8$ and $d(a_f, n_f) = 2$ and $margin = 0.2$ and case B- when $d(a_f, c_f) = 0.2$ and $d(a_f, n_f) = 0.4$ and $margin = 0.2$. Even though the distance between complementary items is more in case A as compared to case B, the loss would be zero in both cases. The network is not learning anything in case A.

We overcome these drawbacks of Siamese triplet loss by incorporating similarity learning, and proposing a novel multitask quadruplet loss for our task.

4.2 Quadruplet Loss

Given an item anchor a and it's mapping a_f, our goal is to have its similar item s_f not farther than a specific distance, called the margin m_s in the mapped space. This can be achieved by minimizing distance between the anchor item and similar item. We first normalize the embeddings a_f, c_f, s_f and n_f to unit norm, denoted by a'_f, c'_f, s'_f, n'_f and then calculate Euclidean distances between those embeddings. Thus, we write similarity based loss as follows:

$$L_{sim} = max(d(a'_f, s'_f) - m_s, 0) \quad (2)$$

Here we denote Euclidean distance between normalized learnt feature vectors by $d(a_f, s_f)$.

For complementary items, we would like the complementary items mapping c_f closer to anchor item a_f, but farther than similar item s_f. Thus, we develop the loss between anchor and complementary items as follows.

$$L_{comp} = max(d(a'_f, c'_f) - m_c, 0) + max(margin_s - d(a'_f, c'_f), 0) \quad (3)$$

Here, the first term in the loss will be positive only when the complementary item is away from the anchor item by more than margin m_c distance, thus making sure the distance between anchor item and complementary item is smaller then m_c. The second term will be positive when the distance between anchor item and complementary item is less than margin m_s, this prevents complementary items projecting onto similar items.

Finally, we want negative items farther from the anchor item than similar and complementary items. We do not want to penalize the model if negative items are farther than a specific distance, let's say margin m_n, otherwise this may hamper learning of similar and complementary items.

The loss for negative items is modeled as follows

$$L_{neg} = max(m_n - d(a'_f, n'_f), 0) \quad (4)$$

One of the constraints here in the proposed set up would be $m_s < m_c < m_n$. We also add $l2$ regularization on the weights in the projection matrix. Our projection matrix is a two layer neural network.

The final loss, L is given by:

$$L = L_{sim} + L_{comp} + L_{neg} + \lambda L_{l2} \tag{5}$$

5 EXPERIMENTS

Metrics: We evaluate our approach using prediction accuracy on complementary and similar items. We also use metric "ranking accuracy". We predict model has ranked items correctly if Euclidean distances between anchor and similar and complementary items are ranked in proper sequence. i.e.

$$y = \begin{cases} 1 \text{ if } d(a'_f, s'_f) < d(a'_f, c'_f) < d(a'_f, n'_f) \\ 0 \text{ otherwise} \end{cases} \tag{6}$$

Implementation: We use the Universal Sentence Encoder [3] to generate 512 dimensional embeddings using item title. The mapping function used is two fully connected layers with ReLU activation. The first layer has 256 hidden units and the second layer has 128 hidden units. We set a learning rate of 0.001 with batch size 512. m_s, m_c, m_n are set to $0.1, 04, 0.8$ respectively. We train the model for 30 epochs; training time is approximately 5 hours with a TensorFlow implementation on Tesla K80 GPUs.

Baselines: We compare our results against text embeddings generated by the Universal Sentence Encoder [3] and the Siamese triplet network utilized by Veit et al. [20].

- Universal Sentence encoder: These text representations are generated by training a transformer [19] based encoder on multiple tasks like semantic text classification, fine grained question classification, sentiment analysis, etc. We only report ranking accuracy on this as we do not have specific thresholds on distances to decide similarity-complementary relationships.
- Siamese triplet network: We train the Siamese triplet network from [20] to learn only complementary relationships using text (as compared to compatibility using images). For a fair comparison, we use the same layers, learning rate etc. while comparing with our approach.

Figure 2 shows the probability density function (pdf) for the Euclidean distance between anchor items and similar items, distance between anchor items and complementary items, and distance between anchor items and negative items for pretrained Universal Sentence Encoder embeddings. Table 3 shows mean and standard deviation for the distance distributions, before and after training. In figure 2 we observe that both similar and complementary items have similar distances, hence it is difficult to differentiate between two. In Figure 3, we show the distance plot for learnt embeddings using quadruplet network. We see that similar and complementary distances are fairly separated after training. The distance distribution between anchor and negative items has more variance. One of the reasons could be that negative items are randomly sampled, which means negative items can also contain some similar and complementary items. We plan to investigate strategic sampling for negative items in our future work.

In Table 4, we compare ranking, complementary, and negative accuracy between different approaches. We can see that, when Siamese networks are optimized for the complementary task only, similarity accuracy is affected, and vice versa. However with our Quadruplet network, we have a good balance between similar and complementary accuracy and the best ranking accuracy. In Figure 4 we show few example predictions from our model, for different clothing items.

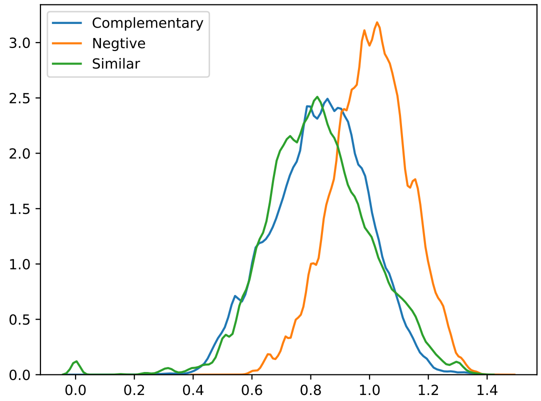

Figure 2: Before training - Euclidean distance distribution between similar, complementary, and negative items

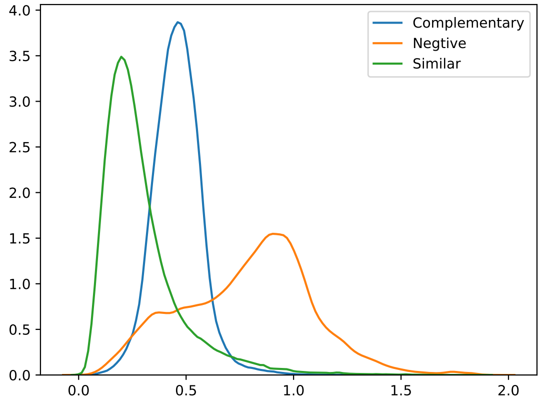

Figure 3: After training - Euclidean distance distribution between similar, complementary and negative items

6 CONCLUSION AND FUTURE WORK

In this study, we propose a novel Siamese Quadruplet network to learn complementary and similarity relations. We utilize item title text, which is widely available item attribute on most e-commerce websites for the task. Our qualitative and quantitative results show that learning complementary and similarity relations together enable better learning of functional complementary relations. Our approach enables searching for similar items and complementary

Table 3: Euclidean distance statistics - between anchor items and their similar, complementary & negative items

	Similar		Complementary		Negative		
Data	Mean	Std Dev	Mean	Std Dev	Mean	Std Dev	cnt
Train Data Before training	0.78089	0.18809	0.78089	0.18809	0.78089	0.18809	3359252
Test Data Before training	0.78748	0.18817	0.78089	0.18809	0.78089	0.18809	334164
Train Data After training	0.24069	0.11226	0.45845	0.11485	0.86774	0.27724	3359252
Test Data After training	0.24772	0.11485	0.45181	0.09963	0.86023	0.27182	334164

Table 4: Evaluation

Method	Ranking Acc	Complementary Acc	Similarity Acc
Universal Sentence Encoder	37.68	-	-
Veit et al. [20]	14.92	91.05	56.45
Our Approach	**67.15**	86.92	**68**

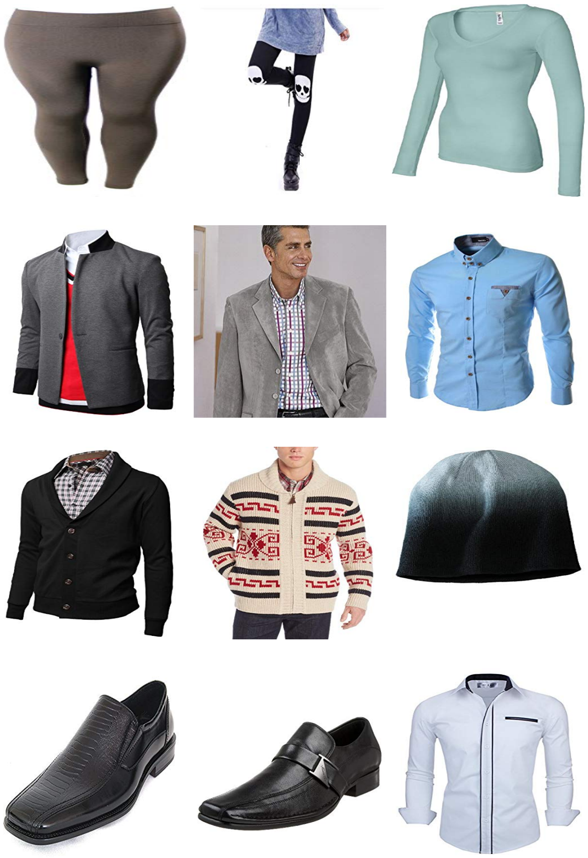

Figure 4: Example model predictions - anchor item (left column), similar item (middle column), and complementary item (right column)

items for a given anchor item. Our approach also is applicable for cold-start items (items for which no purchase data is available).

One area of improvement which we do not address is asymmetry in item relationships. For example, if a customer is shopping for a television stand, (s)he likely already has purchased a television. Although televisions are the most co-purchased items for television stands, televisions are not suitable for recommendations in this scenario. As future work, we plan to expand our approach to model asymmetry and also perform larger scale evaluation on various datasets, including the full Amazon dataset and Polyvore outfits dataset.

REFERENCES

[1] Sean Bell and Kavita Bala. 2015. Learning visual similarity for product design with convolutional neural networks. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 98.

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[3] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175* (2018).

[4] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. 2017. Beyond triplet loss: a deep quadruplet network for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 403–412.

[5] Mihajlo Grbovic, Vladan Radosavljevic, Nemanja Djuric, Narayan Bhamidipati, Jaikit Savla, Varun Bhagwan, and Doug Sharp. 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, 1809–1818.

[6] Xintong Han, Zuxuan Wu, Yu-Gang Jiang, and Larry S Davis. 2017. Learning fashion compatibility with bidirectional lstms. In *Proceedings of the 25th ACM international conference on Multimedia.* ACM, 1078–1086.

[7] Ruining He, Charles Packer, and Julian McAuley. 2016. Learning compatibility across categories for heterogeneous item recommendation. In *2016 IEEE 16th International Conference on Data Mining (ICDM).* IEEE, 937–942.

[8] Wang-Cheng Kang, Eric Kim, Jure Leskovec, Charles Rosenberg, and Julian McAuley. 2018. Complete the Look: Scene-based Complementary Product Recommendation. *arXiv preprint arXiv:1812.01748* (2018).

[9] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* 1 (2003), 76–80.

[10] Perry Marshall. 2013. The 80/20 rule of sales: how to find your best customers. *Entrepreneur October* 229294 (2013).

[11] Julian McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining.* ACM, 785–794.

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

[13] Vineeth Rakesh, Suhang Wang, Kai Shu, and Huan Liu. 2019. Linked variational autoencoders for inferring substitutable and supplementary items. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. ACM, 438–446.

[14] Badrul Munir Sarwar, George Karypis, Joseph A Konstan, John Riedl, et al. 2001. Item-based collaborative filtering recommendation algorithms. *Www* 1 (2001), 285–295.

[15] ScrapeHero. 2017. How Many Products Does Amazon Sell Worldwide âĂŞ October 2017. Retrieved October 27, 2017 from https://www.scrapehero.com/how-many-products-does-amazon-sell-worldwide-october-2017/

[16] Yong-Siang Shih, Kai-Yueh Chang, Hsuan-Tien Lin, and Min Sun. 2018. Compatibility family learning for item recommendation and generation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[17] Jeany Son, Mooyeol Baek, Minsu Cho, and Bohyung Han. 2017. Multi-object tracking with quadruplet convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5620–5629.

[18] Mariya I Vasileva, Bryan A Plummer, Krishna Dusad, Shreya Rajpal, Ranjitha Kumar, and David Forsyth. 2018. Learning type-aware embeddings for fashion compatibility. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 390–405.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

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

[21] Mengting Wan, Di Wang, Jie Liu, Paul Bennett, and Julian McAuley. 2018. Representing and Recommending Shopping Baskets with Complementarity, Compatibility and Loyalty. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 1133–1142.

[22] Jian Wei, Jianhua He, Kai Chen, Yi Zhou, and Zuoyin Tang. 2017. Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Systems with Applications* 69 (2017), 29–39.

[23] Yin Zhang, Haokai Lu, Wei Niu, and James Caverlee. 2018. Quality-aware neural complementary item recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 77–85.