Exploring Heterogeneous Metadata for Video Recommendation with Two-tower Model

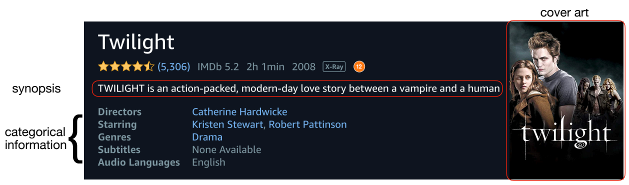
JIANLING WANG, Texas A&M University
AINUR YESSENALINA, Amazon Inc.
ALIREZA ROSHAN-GHIAS, Amazon Inc.

Online video services acquire new content on a daily basis to increase engagement, and improve the user experience. Traditional recommender systems solely rely on watch history, delaying the recommendation of newly added titles to the right customer. However, one can use the metadata information of a cold-start title to bootstrap the personalization. In this work, we propose to adopt a two-tower model, in which one tower is to learn the user representation based on their watch history, and the other tower is to learn the effective representations for titles using metadata. The contribution of this work can be summarized as: (1) we show the feasibility of using two-tower model for recommendations and conduct a series of offline experiments to show its performance for cold-start titles; (2) we explore different types of metadata (categorical features, text description, cover-art image) and an attention layer to fuse them; (3) with our Amazon proprietary data, we show that the attention layer can assign weights adaptively to different metadata with improved recommendation for warm- and cold-start items.

1 INTRODUCTION

Online video services like Amazon Prime Video add cold-start contents, i.e. titles that do not have any watch history, on a regular basis to their catalog. Newly added titles are typically the most interesting for users due to the novelty factor. In fact, many users scroll the page endlessly to find new titles that pique their interest. Thus, it is important to match these highly desired titles to the right users as soon as possible.

Recommender systems based on watch history such as neural network-based architectures [2, 15] or matrix factorization techniques [8, 10] can only generate recommendations for a set of items seen during the model training. This hinders the recommendation to work effectively typically for many days until enough watches have happened. Cold-start titles typically come with different types of metadata, including categorical features (i.e., genre, cast, release time, etc), synopses and cover art images. As in Figure 1, metadata always shows up on the content page to characterize the title and it helps users decide whether they are interested in a title or not. In this work, we postulate that we can utilize the metadata of the cold-start titles and match the right title to the right user, based on their previous watch history. Meanwhile, the metadata can work as supplementary information and help to infer users' preferences on the warm-start titles.



Twilight
★★★★☆ (5,306) IMDb 5.2 2h 1min 2008 X-Ray 12
TWILIGHT is an action-packed, modern-day love story between a vampire and a human
Directors Catherine Hardwicke
Starring Kristen Stewart, Robert Pattinson
Genres Drama
Subtitles None Available
Audio Languages English

Fig. 1. The detail page for a video in Amazon Prime Video. There are different types of metadata for each video, including categorical features, synopsis and cover art.

In this work, we adopt the two-tower architecture for recommendation of videos with heterogeneous metadata. The item and user representations are learned from two different neural towers, which can be used to infer the user-item preference with dot product operation. To support heterogeneous metadata, we develop an attention fusion layer to

Authors' addresses: Jianling Wang, Texas A&M University, jlwang@tamu.edu; Ainur Yessenalina, Amazon Inc. yessenal@amazon.com; Alireza Roshan-Ghias, Amazon Inc. ghiasali@amazon.com.

combine metadata from different modalities. In this work, we explore the use of three different metadata in Amazon Prime Video: categorical features (i.e., genres, actors and directors and release time), synopsis features, and cover art features. With the experiments, we show how different metadata contribute to the title representation learning and also contribute to both warm- and cold-start item recommendation.

2 RELATED WORK

Context-aware recommender systems improve recommendations by incorporating contextual information of the user's decision into the recommendation process [1, 6]. Previous research in video recommendation benefits a lot from different types of contextual information. In an early work, [3] tried to obtain the candidate videos based on co-visitation counts and ranks them utilizing the rule-based signals. To obtain informative user representations for candidate generation, the authors of [2] concatenated heterogeneous user features and input them into a deep neural network. To enable the recommendation for video with limited historic feedback, different content-based video recommendation framework is proposed recently. For example, [12] embedded all the videos relying on their raw video and audio content, and recommended the similar videos located closely on an embedding space. They concatenated heterogeneous user features and generate user embedding with a deep neural network. In our work, we are exploring the impacts of heterogeneous metadata with the two-tower model to conduct efficient user modeling and video metadata modeling at the same time.

Besides modeling user preferences on warm-start videos, we also aim to explore the feasibility of recommending the newly-available cold-start videos. In fact, cold-start item recommendation has been explored via mapping between metadata and the well-trained embedding [5] or training strategies like Dropout [17] and meta-learning [11]. These methods usually work on one particular type of metadata. In this work, we are focusing on exploring the feasibility of utilizing heterogeneous metadata in characterizing videos for cold-start recommendation.

The two-tower structure we have adopted in this work has been applied in the industry to predict the relevance between a query and the candidate items (e.g., news, apps and documents) [4, 9, 15, 19], in which one tower is used to model the query and the other one is for the candidate item. These works show that the two-tower model can be easily modified and extended for multi-view learning, for cross-domain transfer learning or for supporting different negative sampling strategies. There are also previous works utilizing the two-tower model for user and item representation learning in recommendation systems [20]. However, those works usually rely on only one specific type of metadata or directly concatenate various types of metadata. We adopt the two-tower model for user-video preference prediction task and explore the design of the item tower with different item metadata. Ultimately, the model can generate predictions on both warm and cold-start items.

3 TWO-TOWER MODEL

3.1 Architecture

As the name implies, the model has two components: user and item towers, each producing the corresponding embeddings, culminating in a dot product between the two and passing through a sigmoid activation function.

User Tower. Users are represented by their watch histories in the training time period and some additional user-level features such as their country. We are adopting a user encoder architecture previously tested in production. Due to the focus on item encoder in this work, we did not experiment with different types of user encoder and have its architecture fixed. With the user input, the watch history is firstly encoded with a position-aware attention layer to extract the sequential patterns. The user-level features are directly embedded with an MLP layer. Then the encoder concatenates

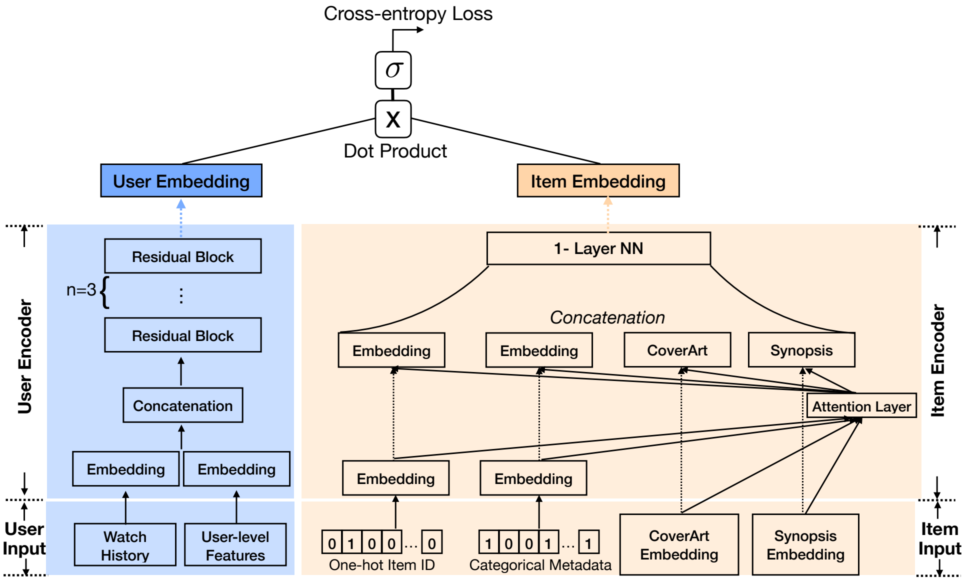


Fig. 2. The proposed two-tower model with heterogeneous metadata.

watch history embeddings and user feature embeddings, passing them through the three residual blocks [7], which is introduced to assist in training a deeper architecture and thus help to achieve better performance.

Item Tower. For each item appearing in the training period, we can use so-called **ID feature** that is in essence the one-hot encoding for all known items. The embedding layer of the ID feature is uniquely linked to the ID of the video and is updated during the training process. Meanwhile, each item also has other metadata available (genres, actors, directors, etc; synopsis, cover art image), which can be used to generate a dense item representation. More details are provided in the following subsections.

Preference Prediction. Given a user-item pair (u, i), we can generate the user embedding $\mathbf{u}$ and item embedding $\mathbf{i}$ with the corresponding towers. Then following the idea of matrix factorization [10], we can use the dot product $\mathbf{u} \cdot \mathbf{i}$ to approximate u's preference on item i. Note that we also apply the Sigmoid function $\sigma(\cdot)$ on the dot product $\mathbf{u} \cdot \mathbf{i}$ as activation. In the training process, we use cross-entropy to calculate the loss. In the prediction process, we will predict users' preference scores on the set of videos and recommend top scoring titles.

3.2 Metadata

In the following section, we will elaborate on three types of metadata used in the item tower: categorical features, synopsis and cover art.

Categorical features is a subset of metadata listed on the product page to characterize the videos. They include categorical features represented with a high-dimensional binary vector.

- **Genre** is used to indicate the topic of a title as displayed on the detail page, which covers 27 tags and 227 subgenre tags indicating more fine-grained topics. We use a binary vector with 227+ 27 dimensions to represent this feature.

- **Actors/Directors** are important factors indicating the relevance of the title to a user. To avoid the long-tail issue, only the top-2000 actors and directors are taken into consideration.
- **Maturity Rating** containing 17 different levels, indicating the maturity level of a title.
- **Country of Origin** containing 159 different values, indicating the country/region title is created in.
- **Release Year** indicating the time the title was released.
- **Acquisition Date** indicating when the title became available in Prime Video. We set the granularity to be 1-month to convert the release dates into categorical values.
- **Popularity** is based on the number of views in a certain period. We use two different values, one is based on the 2-year watching history and the other one is based on the watching history of the most recent 60 days. We use the total number of views to normalize the values and use the log function to smooth the values. We apply uniform discretization transformation on the resulting values to covert them into categorical type of input.

By concatenating the vectors generated from each of the features above, we obtain a high dimensional vector for each of the title.

Synopsis is the short text (usually a few sentences) displayed on the detail page which summarizes a title. It can help users to make a decision on whether to watch a title or not. We encode the synopsis using TF-IDF-weighted sum of the word2vec [13, 14] embeddings of each word.

Cover Art is the image displayed on the detail page to give a visual for the movie, which can also hint on the content and style of the title. In order to obtain an embedding for the cover art, we use a pre-trained 34-layer ResNet trained on ImageNet data [7], and extract the activations before the last layer.

ID is used to represent titles present during the training. Note that we do not have IDs for cold-start users during training. During the evaluation process, we set the ID embedding to zero for cold-start titles.

3.3 Attention Layer for Fusion

After obtaining the ID, categorical, synopsis and cover art embeddings, we concatenate them all together. However, different components can have different influences on the title representations. For example, some titles have informative cover arts but the others may not. Thus we need to fuse the metadata from different modalities considering their importance. We adopt the attention mechanism [16, 18] to calculate the weights for each component. Let α_t^m denote the attention weight for metadata type m of video t and M includes all the types of metadata that the model considering as input for the item tower. Then we will have the pre-score O_t^m:

$$\alpha_t^m = \frac{\exp O_t^m}{\sum_{k \in M} \exp O_t^k}, \quad O_t^m = \mathbf{z}^T \cdot \tanh(\mathbf{P}\mathbf{h}_t^m + \mathbf{b})$$

where $\mathbf{h}_t^m$ represents the embedding for metadata type m of title t, $\mathbf{P}$ is the weight matrix, $\mathbf{z}$ is a transform vector and $\mathbf{b}$ is the bias vector. By applying softmax operation on the pre-scores, we can obtain the attention weights of different components which can sum up to be one. After we obtaining the weights, they will be multiplied with the metadata before the concatenation. In the end, a 1-layer perceptron is used to convert the fused embedding to have the same size of the user representation. Given that cold-start items haven't shown up in the training, the ID embedding will be a zero vector. Thus the item representation is purely based on its metadata.

4 EXPERIMENT

In this section, we will first elaborate the experiment setup and then explore the feasibility of utilizing heterogeous metadata for both warm- and cold-start items in video recommendation with experiments on Amazon Prime Video streaming history.

4.1 Evaluation methodology and metrics

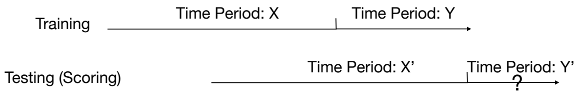


Fig. 3. Illustration of training and scoring data splitting.

To evaluate the proposed model, we conduct a series of offline experiments on real-world Amazon Prime Video data. As shown in Figure 3, during the training process, we use the watch history of users in a two-year time period (X) to predict their watch behaviors in the following two-week period (Y). Then, in the scoring (testing) mode, given the watch history in the two-year period X', it can calculate the preference scores and predict users' watch behavior for the following one-week time period Y'. To avoid data leakage, there is no overlap between Y' and $(X \cup Y)$.

For each user, we calculate preference scores for a list of candidate movies or TV shows. We rank these titles based on the predicted scores and select the Top-K titles for different categories (i.e.,movies and series). We pick $K = 6$ to simulate the use cases in Amazon Prime Video. With the top-k titles and the ground-truth, we adopt 4 different metrics to evaluate the recommendation performance. (1) **Precision@K** represents the ratio of the actual watched items among the Top-K Recommendation; (2) **Recall@K** is the ratio of the actual watched items which were uncovered by the recommendations; (3) **Coverage@K** shows the number of unique items that were recommended for all users, and (4) **Converted Coverage@K** to indicate the number of unique items recommended to all users that were actually watched.

Data. We collect and split the data according to the scheme explained in Figure 3. Specifically, the training data with density[1] of 0.00672% has the length of 2-year in total (i.e., $X + Y = 2$ years), in which the data for the last 14-day is treated as labels. For the test set, we collect user streaming history in 2-year as input features to predict what user is going to watch in the next 7-day (Time Period Y'). To evaluate how the proposed model works for cold-start title recommendation we did a series of offline experiments. Here we have a set of items that have no watch history in time period X, Y or X' but have at least one watch in time period Y'. There are 360 movies and 75 TV series in total in this category. In the following experiments, during the scoring/evaluation, we only calculate preference scores for this set of cold-start titles and rank among them.

Parameter Setup. We use positive samples from a user's watch history, and negative samples from randomly sampling a set of items the user hasn't watched before. We tried different negative sampling rates ranging from 1 to 30. We found that more negative samples can lead to better performance. Empirically, the recommendation performance becomes stable when the negative sampling rate is larger than 20. The embedding size before the dot product is also set to be 512. We used Adam optimizer with the learning rate of 0.001.

[1] density = $\frac{\sharp watches}{\sharp users * \sharp items}$

	P@6 (%)		R@6 (%)		Cov@6		ConCov@6	
Model	Movie	TV	Movie	TV	Movie	TV	Movie	TV
Synopsis	(B)	(B)	(B)	(B)	(B)	(B)	(B)	(B)
Cover art	0.61	0.77	2.07	1.41	-30.87	-26.65	-25.60	-23.59
Categorical	11.38	-25.38	10.27	-11.76	19.24	-55.50	30.13	-46.38
NCF-extention	15.38	7.69	12.75	8.24	44.05	4.52	70.72	15.82
Con (w/o ID)	13.53	1.54	12.76	1.88	33.83	0.31	36.78	4.29
Att (w/o ID)	17.23	15.38	14.94	16.24	65.06	9..03	79.49	18.49
Con (w/ ID)	17.85	18.46	14.95	19.05	79.92	25.14	99.15	39.14
Att (w/ ID)	**26.76**	**25.38**	**21.88**	**24.71**	**167.57**	**50.63**	**176.23**	**81.76**

Table 1. Comparison on warm-start video recommendation. For fair comparison, *NCF-extention* extends NCF [8] by concatenating both the watch history embedding and user-level features embedding with the user ID embedding. (B) indicates the basline for percentage calculation. All numbers are reported in percentage (%) lift w.r.t. the baseline. *Synopsis*, *Cover art* and *Categorical* denote the models use only categorical features, synopsis features or cover art features; *Con* concatenate all types of metadata directly, and feeds the concatenation into the 1-NN; *Att* means to fuse different types of metadata using attention.

4.2 Results

4.2.1 Warm-start Video Recommendation Task. To fully understand how different metadata and the proposed attention layer work for video recommendation, we evaluate the recommendations for the *warm-start videos*. The results are summarized in Table 1. First, we compare the models using only one type of metadata. We can see that the categorical features can result in the best offline performance under different evaluation metrics, indicating that the categorical features are more informative in characterizing the warm-start videos compared with other metadata. When we fuse the aforementioned metadata, the proposed attention layer can bring in significant improvement compared to the models using one of the metadata, while the simple concatenation only works slightly better than those models. This observation illustrates the effectiveness of the attention layer. Furthermore, if we take the ID embedding into consideration, under the warm-start setup, the models can perform better than all the models without ID embedding. The reason is that for items with abundant watch history, the ID embedding layer can learn an informative representation, which hints at the necessity of ID embedding in the warm-start setup.

4.2.2 Cold-start Item Recommendation Task. We compare the proposed two-tower model with the random baseline in which we just randomly select 6 cold-start titles to each user. For cold-start titles, during evaluation/scoring phase, we set their ID embedding to zero. To examine how the ID embedding will influence the cold-start recommendation, we also compare the two-tower model with all the metadata and the version with metadata and ID embedding. The

	P@6 (%)		R@6 (%)		Cov@6		ConCov@6	
Model	Movie	TV	Movie	TV	Movie	TV	Movie	TV
Random	(B)	(B)	(B)	(B)	**(B)**	**(B)**	(B)	(B)
Att (w/o ID)	**2556.52**	**385.42**	**2976.51**	**400.11**	-26.46	-9.33	354.54	3.85
Att (w/ ID)	1221.73	368.75	1277.35	329.92	-2.78	**0.00**	**486.36**	**57.69**

Table 2. Comparison on cold-start video recommendation. (B) indicates the baseline for percentage calculation. All numbers are reported in percentage (%) lift w.r.t. the baseline. *Random* is the random recommendation strategy; *Att w/o ID* is the two-tower model with Attention layer fusing all types of metadata; *Att with ID* is the two-tower model considering all types of metadata and ID embeddings, using Attention layer to fuse those metadata and ID embeddings.

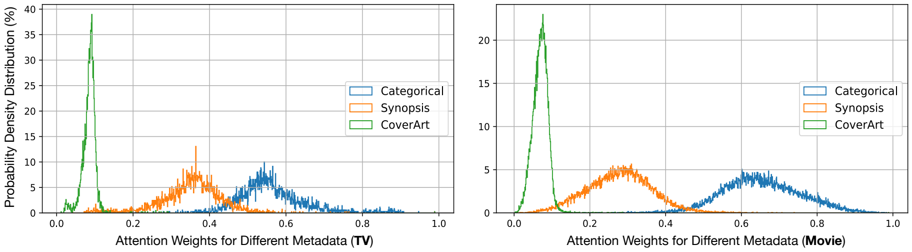


Fig. 4. The distribution of attention weights.

results are summarized in Table 2. We can see that, both the versions with or without ID embedding can beat the random method significantly, explaining the effectiveness of the two-tower model with metadata. Further, the model with ID embedding falls behind the model without ID embedding in terms of precision@6 and recall@6. The reason is that higher weights are usually assigned to the ID embedding by the attention layer in the training phases, and leading to a weak representation for the metadata components. Thus if we remove the ID embedding, we can see the improvements for the cold-start item prediction task. Note, the convertedCoverage@6 and Coverage@6 are slightly higher for the two-tower model with ID embedding. We need to carefully consider this tradeoff scenario while designing an appropriate model for cold-start video recommendation.

4.3 Visualization

To examine how attention layer works in controlling the fusion process, in Figure 4, we summarize the resulted weights for different types of metadata in TV shows and movies. The results for movie and TV shows are shown individually. We find that for both title categories, cover art feature gets the lowest weights, and categorical features get the highest weights. This is intuitive, since categorical data is rich with important features, whereas the cover art is only represented by a pre-trained embedding, and not fine-tuned for this task. When comparing TV shows and movies, we find that synopsis features are more important for TV shows compared to movies.

5 CONCLUSION

We proposed a two-tower model for cold- and warm-start item recommendation with heterogeneous metadata. We also explored different types of metadata, including categorical features, cover-art images and synopsis, With offline experiments, we show that the proposed framework can produce best recommendations by fusing different types of metadata using attention. By introducing the metadata with the well-designed attention layer, the two-tower model enables the recommendation for cold-start videos and also improve the recommendation for warm-start videos.

REFERENCES

[1] Gediminas Adomavicius, Ramesh Sankaranarayanan, Shahana Sen, and Alexander Tuzhilin. 2005. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information systems (TOIS)* 23, 1 (2005), 103–145.

[2] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.

[3] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. 2010. The YouTube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems*. 293–296.

[4] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th international conference on world wide web*. 278–288.

[5] Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, Steffen Rendle, and Lars Schmidt-Thieme. 2010. Learning attribute-to-feature mappings for cold-start recommendations. In *2010 IEEE International Conference on Data Mining*. IEEE, 176–185.

[6] Boning Gong, Mesut Kaya, and Nava Tintarev. 2020. Contextual Personalized Re-Ranking of Music Recommendations through Audio Features. In *CARS Workshop at ACM RecSys*.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[8] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[9] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. 2333–2338.

[10] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[11] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. 2019. Melu: Meta-learned user preference estimator for cold-start recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1073–1082.

[12] Joonseok Lee and Sami Abu-El-Haija. 2017. Large-scale content-only video recommendation. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 987–995.

[13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[14] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[15] O. Rybakov, V. Mohan, A. Misra, S. LeGrand, R. Joseph, Kiuk Chung, Siddharth Singh, Q. You, Eric T. Nalisnick, Leo Dirac, and Runfei Luo. 2018. The Effectiveness of a two-Layer Neural Network for Recommendations. In *ICLR Workshop*.

[16] 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.

[17] Maksims Volkovs, Guang Wei Yu, and Tomi Poutanen. 2017. DropoutNet: Addressing Cold Start in Recommender Systems.. In *NIPS*. 4957–4966.

[18] Jianling Wang, Ziwei Zhu, and James Caverlee. 2020. User Recommendation in Content Curation Platforms. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 627–635.

[19] Ruoxi Wang, Zhe Zhao, Xinyang Yi, Ji Yang, Derek Zhiyuan Cheng, Lichan Hong, Steve Tjoa, Jieqi Kang, Evan Ettinger, and H Chi. 2019. Improving Relevance Prediction with Transfer Learning in Large-scale Retrieval Systems. In *Proceedings of the 1st Adaptive & Multitask Learning Workshop*.

[20] Ji Yang, Xinyang Yi, Derek Zhiyuan Cheng, Lichan Hong, Yang Li, Simon Xiaoming Wang, Taibai Xu, and Ed H Chi. 2020. Mixed negative sampling for learning two-tower neural networks in recommendations. In *Companion Proceedings of the Web Conference 2020*. 441–447.