A Stacking and Transfer Learning with Diverse Similarity For Building Multilingual Session-based Recommendation Systems

Jiangwei Luo*
luojw2007@126.com
MGTV REC
Changsha, Hunan, China

Zhouzhou He†
hzhzh007@gmail.com
MGTV REC
Changsha, Hunan, China

Ye Tang
tangye908@gmail.com
MGTV REC
Changsha, Hunan, China

Wentao Tang
hakula.t@gmail.com
MGTV REC
Changsha, Hunan, China

Cheng Li
kkxcam@gmail.com
MGTV REC
Changsha, Hunan, China

ABSTRACT

It is crucial for e-commerce stores to simulate customers' shopping intentions, as it directly affects user experience and user engagement. Session-based recommendations, which utilizes customer session data to predict their next purchase, has become increasingly popular with advances in data mining and machine learning techniques.However, few studies have explored session-based recommendation under real-world multilingual and imbalanced scenarios. To imporve this research, Amazon published a large-scale shopping queries dataset [1] and hosted KDD Cup 2023 Challenge for building multilingual recommendation systems.

In this pager, the recommendation algorithm team of MGTV present an effective and industrial solution to this challenge,our recommendation pipeline is composed of six stages, which is focused on data preprocessing, candidate generation, construct training samples, build ranking models, feature engineering and blend. Finally, with our solution, our team MGTV_REC won 2nd place in task1[1] and task2[2] among 1990+ participants.

KEYWORDS

Session-based Recommendation, Transfer Learning, KDD Cup, Multilingual Recommendation Systems

ACM Reference Format:
Jiangwei Luo, Zhouzhou He, Ye Tang, Wentao Tang, and Cheng Li. 2023. A Stacking and Transfer Learning with Diverse Similarity For Building Multilingual Session-based Recommendation Systems. In *Proceedings of The 2nd place in Amazon KDD Cup 2023 task1 and task2. (KDDCup '23).* ACM, Long Beach, California, United States, 5 pages.

1 INTRODUCTION

1.1 Background

Amazon KDD Cup 2023 Challenge for building multilingual recommendation systems is aims to provide practical solutions that benefit customers worldwide by promoting diversity and innovation in data science.

In order to further achieve this goal, amazon presents the "Multilingual Shopping Session Dataset", a dataset consisting of millions of user sessions from six different locales, where the major languages of products are English, German, Japanese, French, Italian, and Spanish. the dataset is imbalanced, with fewer French, Italian, and Spanish products than English, German, and Japanese.

1.2 Dataset Description

The "Multilingual Shopping Session Dataset [1]" is a collection of anonymized customer sessions that includes six different regional products. It consists of two data tables: the User Session Table and the Product Attributes Table. The User Session Table captures the sequential interactions of users with the products over time. On the other hand, the Product Attributes Table encompasses various detailed information about the products, such as product titles, prices in local currency, brands, colors, and descriptions.

1.3 Task Description

The main objective of this competition is to build advanced session-based algorithms/models that directly predicts the next engaged product or generates its title text. The challenge contains three different tasks, We were primarily involved in Task 1 and Task 2, and the relevant tasks are as follows:

- Task1: **Next Product Recommendation**
  Task 1 aims to predict the next product that a customer is likely to engage with, given their session data and the attributes of each product. The test set for Task 1 comprises data from English, German, and Japanese locales. We are required to create a program that can predict the next product for each session in the test set. For each session, we should predict 100 product IDs (ASINs) that are most likely to be

*Equal contribution And Corresponding author
†Equal contribution
[1] https://www.aicrowd.com/challenges/amazon-kdd-cup-23-multilingual-recommendation-challenge/problems/task-1-next-product-recommendation/leaderboards
[2] https://www.aicrowd.com/challenges/amazon-kdd-cup-23-multilingual-recommendation-challenge/problems/task-2-next-product-recommendation-for-underrepresented-languages/leaderboards

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
KDDCup '23, Aug 09, 2023, Long Beach, CA, USA
© 2023 Copyright held by the owner/author(s).

engaged, based on historical engagements in the session.The evaluation metric for Task 1 is Mean Reciprocal Rank (MRR).

- Task2: **Next Product Recommendation for Underrepresented Languages/Locales**
  The goal of this task is similar to Task 1, while the test set is constructed from French, Italian, and Spanish. In task 2, we focus on the performance on these three underrepresented languages. It is encouraged to transfer the knowledge gained from the languages with sufficient data such as English, German, and Japanese to improve the quality of recommendations for French, Italian, and Spanish.The input/output and evaluation metrics are the same to Task 1.

2 RELATED WORK

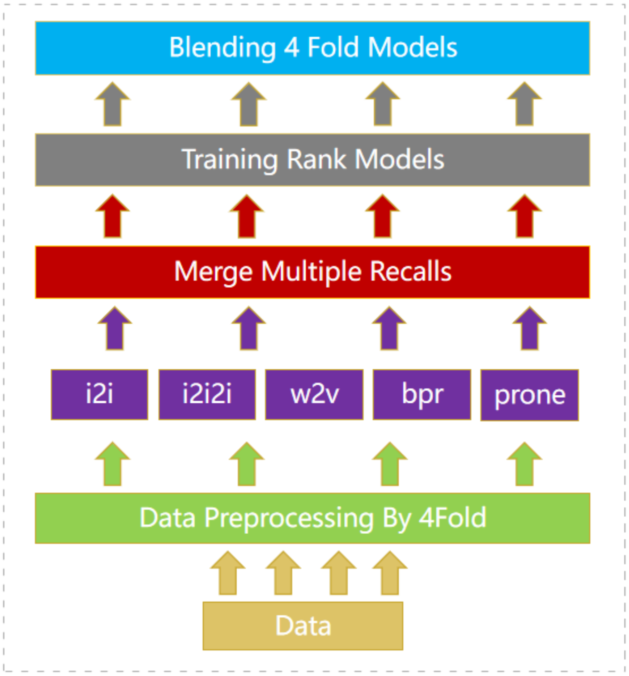


Figure 1: Solution Overall Architecture

In task1 and task2, we designed the 6 stage for building multilingual recommendation system, which is focused on Data Preprocessing, Candidate Generation, Construct Training Samples, Build Ranking Models, Feature Engineering and Blend.

2.1 Data Preprocessing

Given that datasets from the 3 tasks are exactly in the same format, due to the absence of a timestamp column in the given sessions data, we randomly split the train sessions data based on session id in task1 and task2. 75% of the data is used as historical data, while the remaining 25% is used as training data. In the training data, we use the last purchased item by session as the label. We combine the historical data, training data, phase_1 test data and phase_2 test data together to design the recall strategy and feature engineering. However, the label information needs to be removed from the training data.

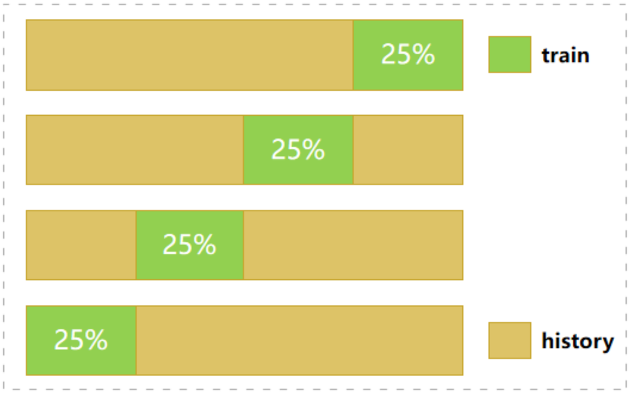


Figure 2: Data Preprocessing By KFold

Following this random data splitting approach, we further utilize K-fold cross-validation with K=4 for data partitioning. As shown in Figure 2, each fold's training set will have no cross-over. Through this process, we will obtain results from four different models. Finally, we will be blending the outputs of these four models.

2.2 Candidate Generation

We have designed five identical recall strategies with diverse similarity in task1 and task2 For generating the candidate items.

2.2.1 *Item to Item Similarity.*

First, we represent the user behavior sequence data from the competition dataset as an item-to-item co-occurrence directed graph. As shown in Figure 3. In this graph, nodes represent items, and edges represent co-occurrence relationships within a user session.

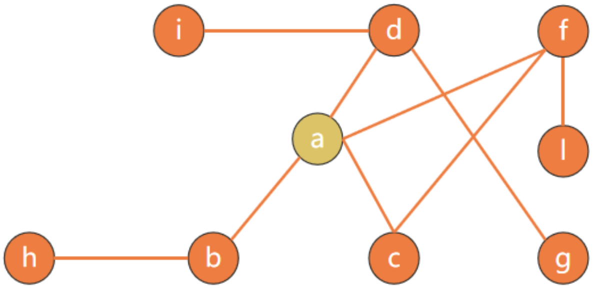


Figure 3: I2I Graph

The purpose of constructing co-occurrence relationships is to calculate the similarity between edges. The similarity between edges is computed using appropriate algorithms based on the co-occurrence relationships across multiple user sessions.

The specific algorithm is represented by the following formula: I2I_Sim(i, j), which indicates the similarity between items i and j. This can be obtained by considering users who interacted with both items simultaneously. The item-to-item[6, 7] similarity calculation takes into account various co-occurrence relationship weight factors.

$$I2I\_Sim(i,j) = \sum_{u \epsilon U_i \bigcap U_j} \frac{W_{pos}(i,j) * W_{sessions}(u) * W_{is\_last} * W_{is\_order}}{\sqrt{|i|} * \sqrt{|j|}}$$

$$W_{pos}(i,j) = \frac{1}{\log\left(|Pos_i - Pos_j| + 1\right)}$$

$$W_{sessions}(u) = \frac{1}{\log\left(|sessions| + 1\right)}$$

$$W_{is\_last} = \begin{cases} 1 & \text{if } is\_last \text{ is True} \\ 0.7 & \text{if } is\_last \text{ is False} \end{cases} \quad (1)$$

$$W_{is\_order} = \begin{cases} 1 & \text{if } is\_order \text{ is True} \\ 0.7 & \text{if } is\_order \text{ is False} \end{cases} \quad (2)$$

Firstly, there is a position weight factor that emphasizes the relevance between items based on their proximity in the user's interaction sequence. The closer their positions, the higher the weight.

Next, there is a purchase factor that assigns a weight of 1 if the co-occurrence relationship represents the last purchase, and 0.7 otherwise.

There is also a sequence factor that assigns a weight of 1 if the co-occurrence relationship follows the chronological order, and 0.7 if it is in reverse order.

Lastly, there is a weight decay for popular users and items. This is because highly popular users and items have a broad influence, and many items may be associated with them, which may not reflect personalized factors. Therefore, their weights are attenuated based on their popularity.

The item-to-item similarity calculation integrates these position, last purchase, sequence order, and popularity weight factors to derive a comprehensive similarity measure.

2.2.2 Item to Item to Item Similarity.

In the user behavior data, due to the short length of each user's interaction data, averaging only 3-4 interactions, we needed to introduce second-order i2i (item-to-item) relationships to enhance the correlation between items. We referred to this as i2i2i similarity. As shown in the following formula:

$$I2I2I\_Sim(i,j) = \sum_{K} I2I\_Sim(i,k) * I2I\_Sim(k,j)$$

As shown in Figure 4, when constructing the second-order relationship graph, we removed the first-order relationships between edges. Therefore, i2i2i serves as a complement to i2i relationships.

2.2.3 Word2vec Embedding Similarity.

Word2vec [4] is a technique for natural language processing published in 2013, The word2vec algorithm uses a neural network model to learn word associations in a large corpus of text. We combine a single user's views and purchases of the item id as a sentence to input into word2vec model. As mentioned before, most sessions are very short, so we just use the window size of 2. Considering there are only about 5 million sessions and 1 of millions of interacted items, which is not large enough to make embedding learned effective, so we use the 32 as the embedding dimension.

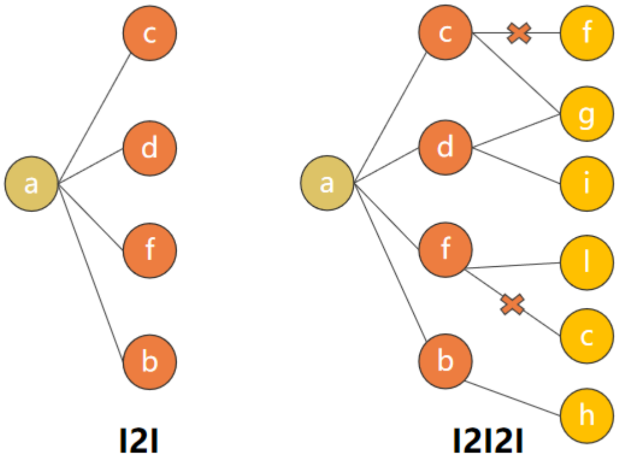


Figure 4: I2I2I similarity constructed by I2I similarity

We use word2vec embedding similarity with position weight, session weight, and item weight to re-rank the top 200 nearest item-to-item-to-item (i2i2i) similarities.

2.2.4 BPR Score.

BPR [5] (Bayesian Personalized Ranking) is a recommendation algorithm designed for collaborative filtering based on implicit feedback. It focuses on learning personalized ranking models from user-item interactions, where the feedback is often binary (e.g., whether a user has interacted with an item or not).

The algorithm utilizes a pairwise learning-to-rank approach, where it optimizes the ranking order of items based on observed user preferences. It formulates the problem as maximizing the posterior probability of item rankings given the observed user-item interactions.

BPR leverages a matrix factorization technique, typically using matrix factorization models like collaborative filtering with matrix factorization (CF-MF). It learns latent representations (i.e., embeddings) for users and items in a low-dimensional space, and then applies stochastic gradient descent (SGD) to optimize these embeddings by maximizing the posterior probability.

We generate item embedding by BPR with position weight, session weight, and item weight to re-rank the top 200 nearest item-to-item-to-item (i2i2i) similarities.

2.2.5 PRONE Embedding Similarity.

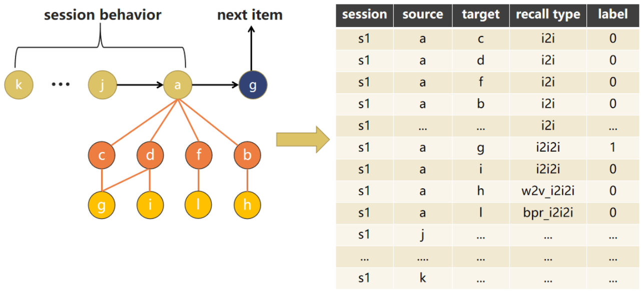
ProNE [8]: Fast and Scalable Network Representation Learning is a research paper that addresses the challenge of network representation learning on large-scale networks.

The main objective of ProNE is to provide a scalable solution for network embedding. It formulates network representation learning as an optimization problem, focusing on preserving proximity relationships between nodes in a low-dimensional embedding space.

ProNE generates high-quality network embeddings that capture structural and similarity information, enabling various downstream tasks such as node classification, link prediction, and community detection.

We generate item embedding by ProNE with position weight, session weight, and item weight to re-rank the top 200 nearest item-to-item-to-item (i2i2i) similarities.

2.3 Construct Training Samples



session	source	target	recall type	label
s1	a	c	i2i	0
s1	a	d	i2i	0
s1	a	f	i2i	0
s1	a	b	i2i	0
s1	...	...	i2i	...
s1	a	g	i2i2i	1
s1	a	i	i2i2i	0
s1	a	h	w2v_i2i2i	0
s1	a	l	bpr_i2i2i	0
s1	j	...	...	...
...	....	...	...	...
s1	k	...	...	...

Figure 5: Construct Training Samples by merge recalls

As shown in Figure 5, We construct a sample set of item-to-item relevance based on the u2i2i modeling approach. We split the user's engaged behavior sequence and take each engaged item as the source. By performing multi-hop random walks in the i2i graph, we extract target items to form relationships of i2i, i2i2i and i2i2i rerank by Word2vec, BPR, ProNE. This process generates the u2i2i sample set.

To introduce labels to the samples, we compare whether the item the user purchases next is consistent with the target item. If they are the same, we label it as a positive sample; otherwise, it is labeled as a negative sample.

2.4 Build Ranking Models

During the previous data preprocessing stage, we randomly split the data into four folds, with each fold being independent of the others. Therefore, we need to build four separate models for each of the four folds.

Due to the considerable size of the dataset after the feature engineering. It requires the model used for training and predictions to be scalable and fast. We use Lightgbm [2], a model based on decision tree algorithm and used for ranking, classification and other machine learning tasks. It has advantage on speed, multiple loss functions, parallel training, and has been proved to be SOTA solution's tool in many challenges.We test the objective of "binary" and "lambdarank [3]" to show the difference of classification and learning to rank, and finally found the learning to rank task can get higher score in the challenge.

2.5 Feature Engineering

2.5.1 *Recall Features.*

In Task1 and Task2, We use the recall strategy as feature inputs to the model, which include i2i similarity, i2i2i similarity, word2vec embedding similarity, BPR Score, ProNe embedding similarity, and the ranking scores of these similarities by grouped by session_id.

2.5.2 I2I/I2I2I Similarity Features With Seqno Gap Windows. In Task1 and Task2, We utilize i2i and i2i2i similarities by Weights with different sequence interval windows. The sequence interval window sizes used are 1, 2, 4, and 7, respectively.

2.5.3 *Text similarity Features.*

In Task1 and Task2, We calculated the Levenshtein distance, Hamming similarity, Jaccard similarity, and Jaro similarity between the titles of the items from the user's last 1, 2, and 3 interactions and the titles of the candidate items.

2.5.4 *Basic Features.*

We generate the basic features of session, item, session-item, session-price, session-brand and so on for task1 and task2 respectively.

2.5.5 *Split Features.*

We calculate the feature of items hot rank by split train and test for task1 and task2 respectively and the data from the task3 is not included in the calculation. When we calculate the feature for train, we kept the last item in historical data. This feature can boost 0.008+ in task1 and task2. The key feature to advance us to others.Here We assume the training dataset and test dataset are sample in diff dates. so the diff popularity may diffrenent, we need to statistics differently.

2.5.6 *Transfer Learning Features.*

In Task2, We are using the model built for Task 1 to predict Task 2. As shown in Figure 6, The predictions from Task 1 are treated as separate features for Task 2. This approach signifies that we are transferring the learned patterns from training Task 1 to Task 2. As we have divided the training data into 4 folds, we are using this method to predict the current data in Task 2 using different folds of data. Consequently, we will generate four columns representing the features transferred from Task 1 to Task 2. This features can boost 0.005+ in task2.

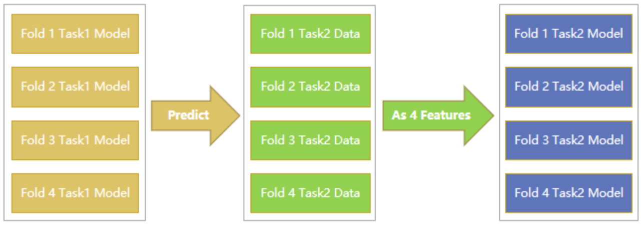


Figure 6: Task1 Patterns Transfer to Task2

2.5.7 *Stacking Features.*

In Task1 and Task2, We are using models trained on other folds to predict the current fold. As shown in Figure 7, For fold 1 Model, fold 2,3,4 models predict fold 1 Sample to generate three features by stacking. In this approach, we will generate three columns of stacking features for the current fold. This features can boost 0.001+ in task1 and task2.

2.6 Blend

Finally, We have indeed built four independent models for the four separate folds. Due to the absence of cross-interactions between the

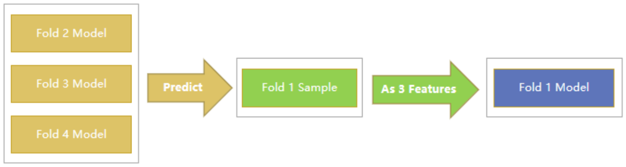


Figure 7: Fold 2,3,4 Models Predict Fold 1 Sample by Stacking

Rank	Team Name	MRR@100
1	NVIDIA-Merlin	0.41188
2	**MGTV-REC**	**0.41170**
3	unirec	0.40477
4	gpt_bot	0.40476
5	LeaderboardCar	0.40339
6	AIDA	0.40317
7	piggy-po	0.39754
8	iCanary	0.39651
9	wxd1995	0.39592
10	xuy	0.39566

Table 1: Top 10 score in task1. Our team "MGTV-REC" won 2nd in Task1 of Amazon KDD Cup 2023

folds, there is significant variation among the four models, with an average ranking correlation of around 0.75 in task1 and task2. We blend 4 fold results of final models in task1 and task2 respectively and this method can boost 0.004+ in task1 and task2, the final score is about 0.41170, 0.46578 in task1 and task2 respectively.

3 EXPERIMENT RESULTS

Table 1, Table 2 are the top 10 final leaderboard with MRR@100 score in task1 and task2. With our solution, our team MGTV-REC won 2nd place in task1 and task2.

4 CONCLUSION

In this paper, We have proposed a generic and popular pipeline for building recommendation systems, which includes data preprocessing, recall based on various diversity similarity, merging multiple recalls and constructing training samples, building ranking models, and blending. In addition, we have conducted extensive data analysis specifically for the competition, leading to the development of targeted feature engineering methods. It was also a key factor which won the 2nd place in Task1 and Task2 of Amazon KDD Cup 2023 Challenge.

Rank	Team Name	MRR@100
1	NVIDIA-Merlin	0.46845
2	**MGTV-REC**	**0.46578**
3	gpt_bot	0.46011
4	AIDA	0.45047
5	piggy-po	0.44914
6	chimuichimu	0.44798
7	iCanary	0.44747
8	QDU	0.44618
9	[Acroquest]YAMALEX	0.44380
10	DX2	0.44101

Table 2: Top 10 score in task2. Our team "MGTV-REC" won 2nd in Task2 of Amazon KDD Cup 2023

REFERENCES

[1] Wei Jin, Haitao Mao, Zheng Li, Haoming Jiang, Chen Luo, Hongzhi Wen, Haoyu Han, Hanqing Lu, Zhengyang Wang, Ruirui Li, Zhen Li, Monica Xiao Cheng, Rahul Goutam, Haiyang Zhang, Karthik Subbian, Suhang Wang, Yizhou Sun, Jiliang Tang, Bing Yin, and Xianfeng Tang. Amazon-m2: A multilingual multi-locale shopping session dataset for recommendation and text generation. 2023.

[2] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[3] Tie-Yan Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009.

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

[5] Gantner Z et al Rendle S, Freudenthaler C. Bpr: Bayesian personalized ranking from implicit feedback. *Conference on Uncertainty in Artificial Intelligence*, 2009.

[6] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295, 2001.

[7] Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009, 2009.

[8] Jie Zhang, Yuxiao Dong, Yan Wang, Jie Tang, and Ming Ding. Prone: Fast and scalable network representation learning. pages 4278–4284, 7 2019.