A Two-stage Ranking Framework for Multilingual Recommendation(Team:AIDA)

Shijie Liu
karrich128@gmail.com
East China Normal University
Shanghai, China

Chenliang Zhang
clzhang1997@gmail.com
East China Normal University
Shanghai, China

Xutao Han
luoyun7777777@gmail.com
Zhejiang University
Hangzhou, Zhejiang, China

Wen Wu
wwu@cc.ecnu.edu.cn
East China Normal University
Shanghai, China

Wei Zhang*
zhangwei.thu2011@gmail.com
East China Normal University
Shanghai, China

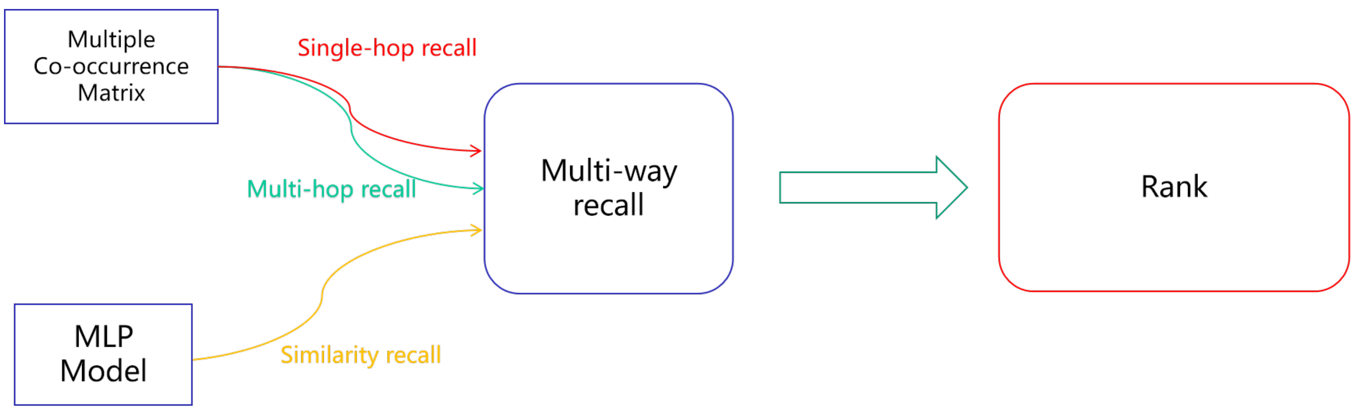

Figure 1: Overall Architecture

ABSTRACT

In the KDD Cup 2023 multilingual recommendation challenge, we proposed a ranking framework, which consists of two main stage: recall and ranking. In the recall stage, we use a carefully-designed co-occurrence matrix for single-hop and multi-hop recall of candidate items. Moreover, an MLP model initialized with "title" information is also used in this stage. In the ranking stage, we designed many features and additional transfer features for task 2's transfer learning.Then we use the XGBoost model for ranking. Our solutions rank 6th in task 1 and 4th in task 2. The code is available at https://github.com/karrich/KDD-CUP-2023-solution.

KEYWORDS

mutilingual, recommendation, transfer learning

*Corresponding author.

KDDCup '23,August 9,2023,Long Beach, CA, USA, August 06-10, 2023, Woodstock, NY

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

ACM Reference Format:
Shijie Liu, Chenliang Zhang, Xutao Han, Wen Wu, and Wei Zhang. 2023. A Two-stage Ranking Framework for Multilingual Recommendation(Team:AIDA). In *Proceedings of ACM (KDDCup '23,August 9,2023,Long Beach, CA, USA).* ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

1.1 Background

Session-based recommendation has always been an important component of e-commerce recommendation tasks. In recent years, with the booming development of cross-border e-commerce, there has been an increasing demand for multilingual session-based recommendation systems. However, most existing methods are designed for specific language-based product datasets, and they perform moderately when faced with diverse and imbalanced data from different locales or languages.

Based on the background mentioned above, Amazon has provided a multilingual shopping session dataset [4] and organized the *KDD Cup 2023 multilingual recommendation challenge*. The aim of this challenge is to develop novel and effective multilingual session-based recommendation systems that provide inspiration for system design and practical engineering solutions for researchers and engineers in the field of recommender systems.

1.2 Dataset and Problem

The dataset comprises two major categories of data tables: user sessions and product attributes. User sessions are lists of products with which users interact in chronological order, and product attributes include textual and numerical information such as title, price, brand, color, and description.

The dataset includes millions of user sessions from six different locales. The users are anonymous, and the main languages used for the products are English, German, Japanese, French, Italian, and Spanish. However, the dataset exhibits a strong imbalance, with fewer products available in French, Italian, and Spanish compared to English, German, and Japanese.

The competition introduces three tasks:

1) Predicting the next engaged product for sessions from English, German, and Japanese.
2) Predicting the next engaged product for sessions from French, Italian, and Spanish.
3) Predicting the title for the next engaged product.

Our work focuses on solving the first two tasks.

In this competition, all tasks share the same training dataset. The summaries of the training dataset are shown in Table 1. In addition, we find through practice that for each item sequence, its label item to be predicted will never appear in this item sequence. This means that for each item sequence to be predicted, the items that have already appeared in this sequence can be directly removed from the candidate set. This method is implemented in our method by default and is not mentioned later.

Table 1: Summary of the training dataset.

Language	Sessions	Products	Avg.Length
German	1,111,416	513,811	4.352
Japanese	979,119	389,888	4.482
English	1,182,181	494,409	4.122
Spanish	89,047	41,341	3.664
French	117,561	43,033	3.545
Italian	126,925	48,788	3.662
Overall	3,606,249	153,1270	4.244

1.3 Task Solution

As shown in Figure 1, our method consists of two stages: recall and ranking. The two-stage paradigm is commonly adopted by recommender systems and has gained tremendous success in related data competitions [2]. Section 2 introduces our data splitting method. Section 3 introduces our method of using co-occurrence matrices for single-hop and multi-hop recall, as well as using MLP to recall and using text information to improve the performance of the MLP model. Section 4 describes our method of constructing features and the ranking model. Section 5 gives the experimental results. Section 6 summarizes the paper.

2 DATASET SPLIT

For the validation set, in order to speed up the validation process, we randomly extracted 0.5% of the data from the training set as the validation set. It turns out that this amount of data is sufficient for us to observe the effect of the model. For training the ranking model, we took out 10% of the training set about Task1 as training data for ranking model, and we used all the training data about Task2 for Task2. We found that some items only appeared in the test dataset, so for training the MLP model and construction of features for the test dataset, we used the training set data and all the provided test dataset.In order to avoid the leakage of MLP model prediction, we deleted the last column of all data used in training ranking model.

3 RECALL

3.1 Co-occurrence Matrix Recall

In our proposed method, co-occurrence matrices are the most important part in both the recall and ranking stages. In the recall stage, our method constructs multiple co-occurrence matrices and performs single-hop and multi-hop recall. Usually, the construction method of co-occurrence matrices is to set the window length and calculate the co-occurrence times of goods in the window. A simple but effective co-occurrence matrix construction method: assuming $P(a, b)$ represents the co-occurrence count of a and b, whenever $[x_i, x_{i+1}]$ appears once in session, $P(x_i, x_{i+1})$ is incremented by 1, and $P(x_{i+1}, x_i)$ is incremented by 0.5.

After constructing the co-occurrence matrices, we use these matrices for single-hop and multi-hop recall. Our method for the single-hop recall is to use the position of goods in the session to weigh the co-occurrence times of goods with other goods in the co-occurrence matrix. The formula for single-hop recall is Equation 1, where X is a session of length n, $X = [x_1, x_2, ..., x_n]$, $P(a, b)$ is the co-occurrence times of item a and item b, W_i is the position weight, where $W_0 = 1$ and the larger i is, the closer W_i is to 0.

$$\arg\max_{y} \sum_{i=1}^{n} W_{n-i} \cdot P(x_i, y) \tag{1}$$

We can view the co-occurrence matrix as a directed graph with edge weights, and single-hop recall is to recall directly connected neighboring nodes with high edge weights. Therefore, our multi-hop recall is designed to recall the neighboring nodes of neighboring nodes with high edge weights to obtain more recalled items. In implementation, we construct a multi-hop co-occurrence matrix to simulate multi-hop recall operations from the directed graph. Specifically, we use the $(l-1)$-hop co-occurrence matrix to construct the l-hop co-occurrence matrix. The formula for calculating the multi-hop co-occurrence matrix is Equation 2, where $P(a, b)$ is the co-occurrence times of item a and item b, and I is the set of all items.

$$P_{l-hop}(i, j) = \sum_{k \in I} \frac{P_{(l-1)-hop}(i, k) \cdot P(k, j)}{\sum_{t \in I} P(k, t)} \tag{2}$$

After constructing the multi-hop co-occurrence matrix, we also use Equation 1 for recall.

3.2 MLP Model

We designed an MLP model for recall to ensure that the number of recalled items is not less than a fixed value. In the ranking stage, our model can also provide high-quality features. Our proposed method consists of two modules, session encoder and target encoder. The

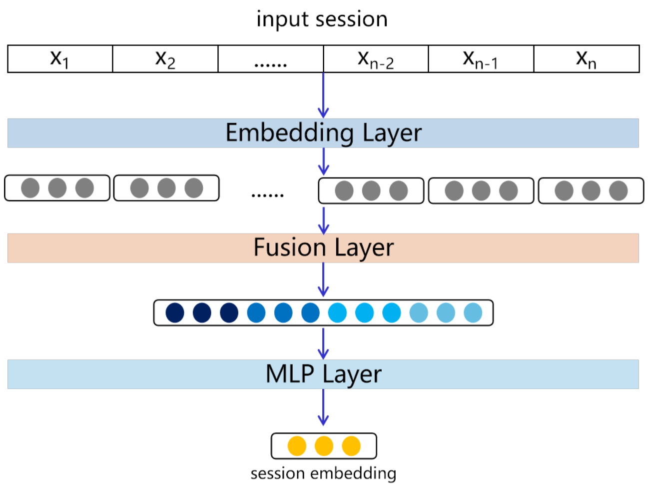

Figure 2: The architecture proposed in this paper to obtain session embedding.

structure of the session encoder is shown in Figure 2, where the Fusion Layer does not contain any trainable parameters, and the operation can be described as Equation 3. Where x_i represents the k-dimensional embedding of the i-th item in the input session, and $[\cdot]$ represents the concat operation.

$$U = [\sum_{0<i<n-3} \frac{x_i}{n-4}, \frac{x_{n-3}+x_{n-2}}{2}, x_{n-1}, x_n] \tag{3}$$

The MLP Layer consists of two fully connected layers and an activation function. The MLP Layer will reduce the dimension of the vector from $4k$ to k. The structure of the target encoder consists of a simple embedding layer, and its parameters are shared with the embedding layer in the session encoder.

Our final optimization target is equation 4, which makes the session embedding and label item embedding as similar as possible. N is the number of negative samples, U_i is the session embedding, Y_i^* and Y_i^- are the embeddings of positive and negative samples respectively, and $cos(\cdot)$ is the cosine similarity calculation function.

$$\mathcal{L} = -log \frac{e^{cos(U_i, Y_i^+)/\tau}}{\sum_{j=1}^{N} e^{cos(U_i, Y_j^-)/\tau}} \tag{4}$$

In the specific implementation, we used bert-whitening[5] to obtain the k-dimensional embedding of each item's title as the initial parameter of the Embedding Layer. We chose the xlm-roberta-large[3] that was pre-trained on multilingual datasets as the backbone. In order to obtain better transfer effects on task 2, we embedded items with the same id in different locales into the same embedding and used data from all locales at the same time during training.

4 RANKING

4.1 Ranking Feature

We designed a large number of features in the ranking stage, especially a large number of co-occurrence matrices of different scales were designed to generate features. In order to avoid training prediction leakage and make full use of all data on task 2, when generating features for each data, we will use all other data except this data. After our optimization, this will not cause high time complexity.

4.1.1 Item-CF. Equation 5 is the main idea of our implementation of Item-CF. We assume that each session comes from a different user, U_i represents the set of sequence interactions with item i, I_u is the historical interaction item sequence of user u, and l_i represents the position of item i in this sequence.

$$\begin{gathered} sim(x,y) = \frac{1}{\sqrt{|U_i||U_j|}} \sum_{u \in U_i \cap U_j} \frac{\alpha^{|l_i - l_j| - 1}}{log(1+|I_u|)}, \\ \alpha = \begin{cases} 0.8, l_i < l_j \\ 0.7, l_i > l_j \end{cases} \end{gathered} \tag{5}$$

Afterwards, we combine the session and recall items and use $sim(\cdot)$ to calculate the score as a feature (such as the score between the last item x_n in the session and the recall item y, $sim(x_n, y)$, and the average score between all items in the session and the recall item $\sum_{i=1}^{n} sim(x_i, y)/n$, etc.).

4.1.2 Co-occurrence Matrix Feature. A lot of our work in the Ranking stage is to build reasonable and effective co-occurrence matrices and use these co-occurrence matrices(as described in Section 3.1) to build a large number of features. The importance of these features even exceeds all other features. We use formula Equation 6 to calculate the score given by the co-occurrence matrix, where $P_i(x,y)$ represents the number of co-occurrences of item x and item y in the co-occurrence matrix P_i we constructed, and I represents all item sets. Of course, we will also generate multiple features for a co-occurrence matrix (as described in Section 4.1.1).

$$score_i(x,y) = \frac{P_i(x,y)}{max(1, \sum_{k \in I} P(x,k))} \tag{6}$$

4.1.3 Attribute Feature. We use the bert-whitening [6] method to calculate the similarity of item titles as features. In addition, we also use the price, material, brand, author, and locale of the item to build a series of features, such as the proportion of items with the same brand in the session as the recall item and the deviation of the recall item price from the average price of items in the session. It is worth mentioning that building one-hot features for locales and training data from different locales together can also improve the model's performance.

4.1.4 MLP Model Feature. The score calculated by the model proposed in Section 3.2 is used as a feature.

4.1.5 Transfer Co-occurrence Matrix Feature For Task 2. In order to perform transfer learning from locales with small amounts of data to locales with large amounts of data, we construct a transfer co-occurrence matrix to generate features. We selected some co-occurrence matrices constructed from the data in Task 1 and used them directly in Task 2. The score calculation method is the same as Equation 6, except that when item x or y does not exist in the locale of the co-occurrence matrix, the score is -1.

4.2 Ranking Model

The ranking model we used is XGBoost[1]. We tested the objectives of "rank" and "binary" and found that learning classification tasks can achieve higher scores. In addition, we found that lowering the score of goods that did not appear in the test set can significantly improve the final score.

5 RESULT

The results of our proposed scenario for tasks 1 and 2 are shown in Table 2 and 3, respectively. With our solution, our team ADIA won 6th place on Task1 and 4th place on Task2.

Table 2: Top 10 scores of task1.

Rank	Team Name	Task 1 Scores
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	xut	0.39566

Table 3: Top 10 scores of task2.

Rank	Team Name	Task 2 Scores
1	NVIDIA-Merlin	0.46845
2	MGTV-REC	0.465780
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

We analyzed the feature importance based on the information gain. Tables 4 and 5 respectively show the feature importance analysis of task 1 and task 2. Since we designed multiple features for each type of feature (for example, we have multiple item-cf features), we only show the highest value of importance. It can be found that the co-occurrence matrix feature we designed is the most important type of feature, which is more in line with our expectations because we spent a lot of time on this.

6 CONCLUSION

In this paper, we have introduced our solutions in KDD CUP 2023. Our approach relies primarily on well-designed various co-occurrence matrices and MLP models. In the end, our solution won 6th place in Task 1 and 4th place in Task 2. In the future, we will focus on GNN and LLMs. They are all well worth exploring in the recommended field. Especially with the rapid development of LLMs recently, it is important to find a good way to use LLMs for recommendation.

Table 4: Feature importance analysis for Task 1.

Feature description.	Importance
A well-designed co-occurrence matrix feature.	15085
A score of the MLP model output.	8428
A feature calculated based on the attribute "Brand".	2100
A feature calculated according to item-cf.	1041
A feature calculated based on the attribute "Author".	660
A feature calculated based on the similarity of the title.	519
A locale feature.	240
A feature calculated based on the attribute "Price".	153
A feature calculated based on the attribute "Material".	101

Table 5: Feature importance analysis for Task 2.

Feature description.	Importance
A well-designed co-occurrence matrix feature.	42739
A score of the MLP model output.	8428
A feature calculated based on the attribute "Brand".	3167
A well-designed transfer co-occurrence matrix feature.	1791
A feature calculated based on the similarity of the title.	1274
A feature calculated according to item-cf.	681
A feature calculated based on the attribute "Author".	481
A feature calculated based on the attribute "Price".	314
A feature calculated based on the attribute "Material".	141
A locale feature.	130

REFERENCES

[1] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16)*. Association for Computing Machinery, New York, NY, USA, 785–794. https://doi.org/10.1145/2939672.2939785

[2] Zeyuan Chen, He Wang, Xiangyu Zhu, Haiyan Wu, Congcong Gu, Shumeng Liu, Jinchao Huang, and Wei Zhang. 2022. A Practical Two-stage Ranking Framework for Cross-market Recommendation. *arXiv preprint arXiv:2204.12682* (2022).

[3] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised Cross-lingual Representation Learning at Scale. *CoRR* abs/1911.02116 (2019). arXiv:1911.02116 http://arxiv.org/abs/1911.02116

[4] 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. 2023. Amazon-M2: A Multilingual Multi-locale Shopping Session Dataset for Recommendation and Text Generation. arXiv:2307.09688 [cs.IR]

[5] Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening Sentence Representations for Better Semantics and Faster Retrieval. arXiv:2103.15316 [cs.CL]

[6] Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316* (2021).