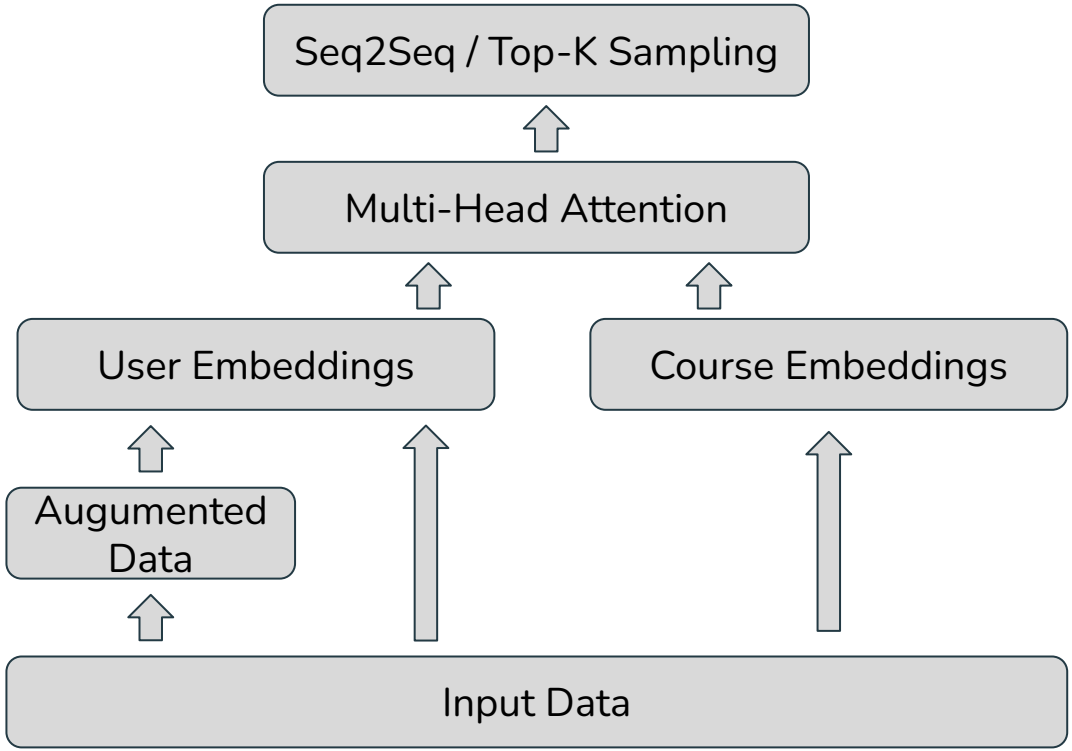
Preference Prediction Challenge

Goal for challenge

- learn the correlation between different courses
- predict the courses the user would buy in the future

FlowChart


Seq2Seq / Top-K Sampling
Multi-Head Attention
User Embeddings
Course Embeddings
Augumented Data
Input Data

First Part :

Data Preprocessing

Data Augmentation

Extra user data may help to improve the performance!

Generate new user data from the original data !!

Randomly mask some interest from a user to produce a “human clone”.

We found that there are most choices in “interest”. Masking some (<10%) of them leads to minimum noise but produce lots of “human clone”. Ω(2^(n/10))

Data Augmentation

user1, interest1 [redacted] interest3

user2, interest1 interest2

user3, [redacted] interest2 interest3 [redacted]

user4, interest1 interest2 interest3

user5, [redacted] interest2

user1, interest1 interest2 interest3

user2, interest1 interest2

user3, interest1 interest2 interest3 interest4

user4, interest1 interest2 interest3

user5, interest1 interest2

aug_user6, interest1 interest3

aug_user7, interest2

aug_user8, interest1 interest2 interest3

aug_user9, interest2

By data augmentation, we have more data now !!

Data preprocessing : courses

Pick important features of courses : target group, groups, subgroups, topics.

Concatenate each parts seperated by Chinese special token, like [目標客群] etc.


```
{
    "text": "[目標客群]熱愛彩妝的人[課程類別]生活品味[課程子類別]更多生活品味,護膚保養與化妝[課程分類]更多生活品味,護膚保養與化妝",
    "id": "61888e868f154b000781b45a"
}
```

Data preprocessing : users

like how we preprocess the courses data,

we pick important features of users : gender, occupation, interest, recreation


```
"id": "5fedf958af850a915c86362c",
"text": "[性別]女性[職業]科技業[興趣]投資理財_理財,攝影_動態攝影,攝影_影像創作,攝影_影視創作,攝影_後製剪輯,語言_英文[娛樂]旅行旅遊,桌遊,運動健身,金融理財",
"labels": [
    318,
    573,
    625,
    628,
    564,
    304,
    468,
    522,
    580
]
```

Data preprocessing : users

In addition, we cluster similar interests of one user

For example, if there is a user have some interests like following in the original data

user id, ….. 攝影_動態攝影, 語言_英文, 攝影_後製剪輯, 語言_日文 …….. ,

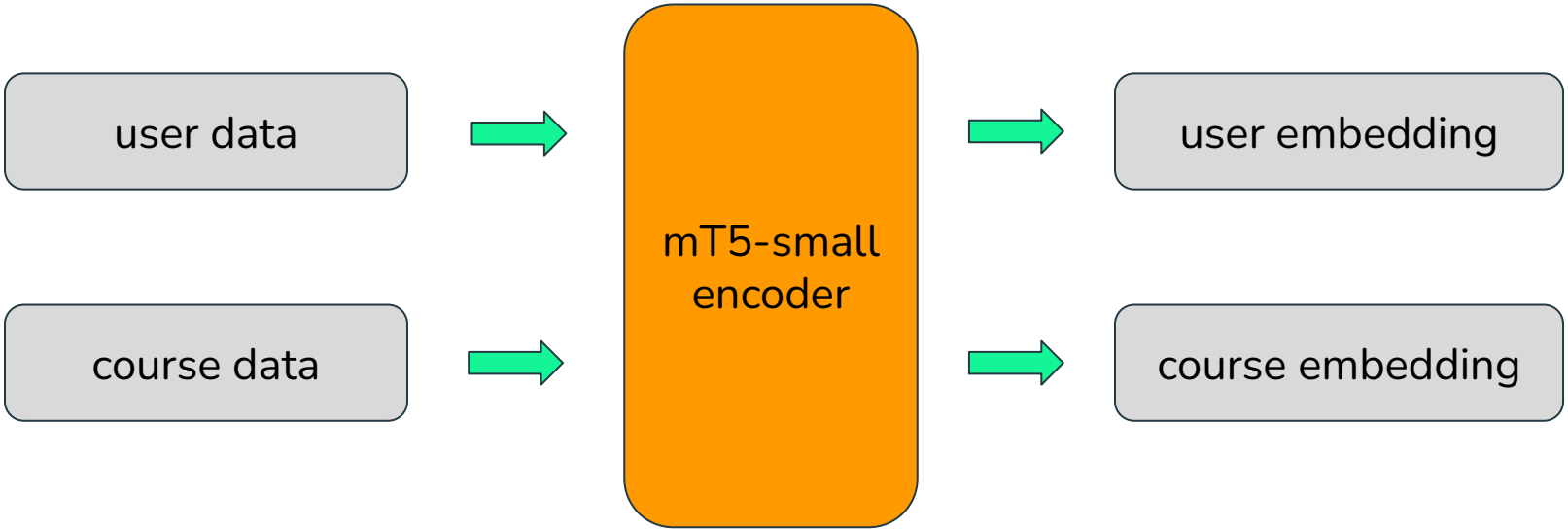
After preprocessing, the data of this user become

user id, ….. 攝影_動態攝影, 攝影_後製剪輯, 語言_英文, 語言_日文 …….. ,

We expect clustered interests are more reasonable for model!

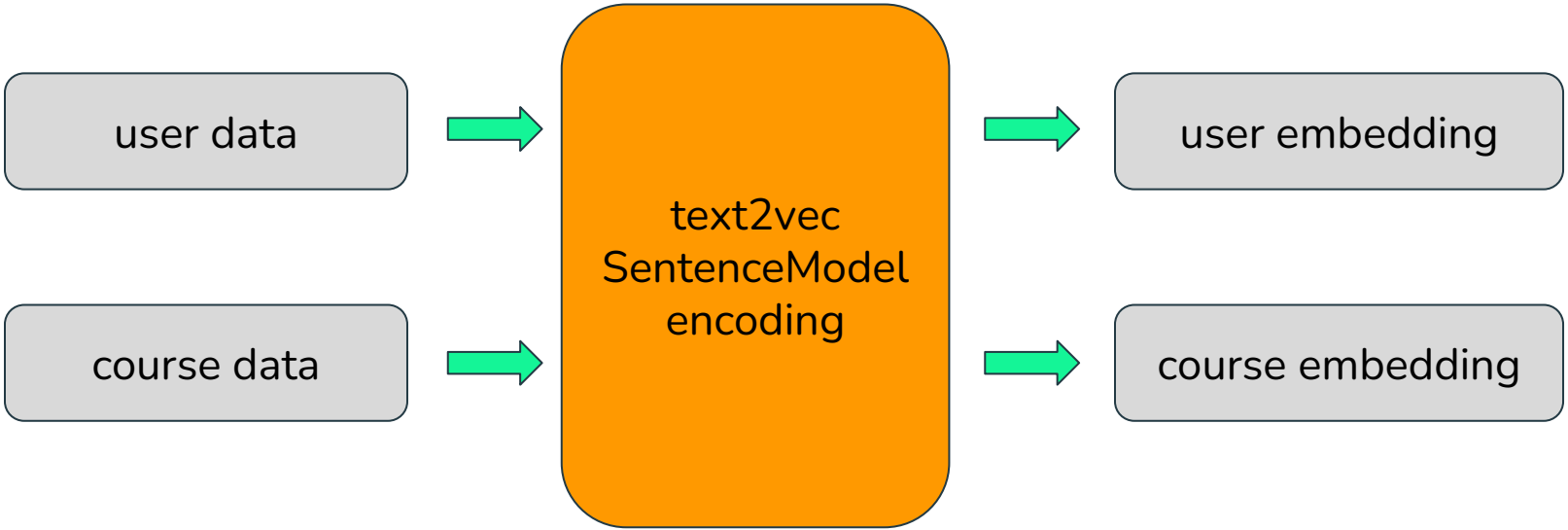
User and Course Embedding - mT5

pre-trained model : mT5-small



User and Course Embedding - text2vec

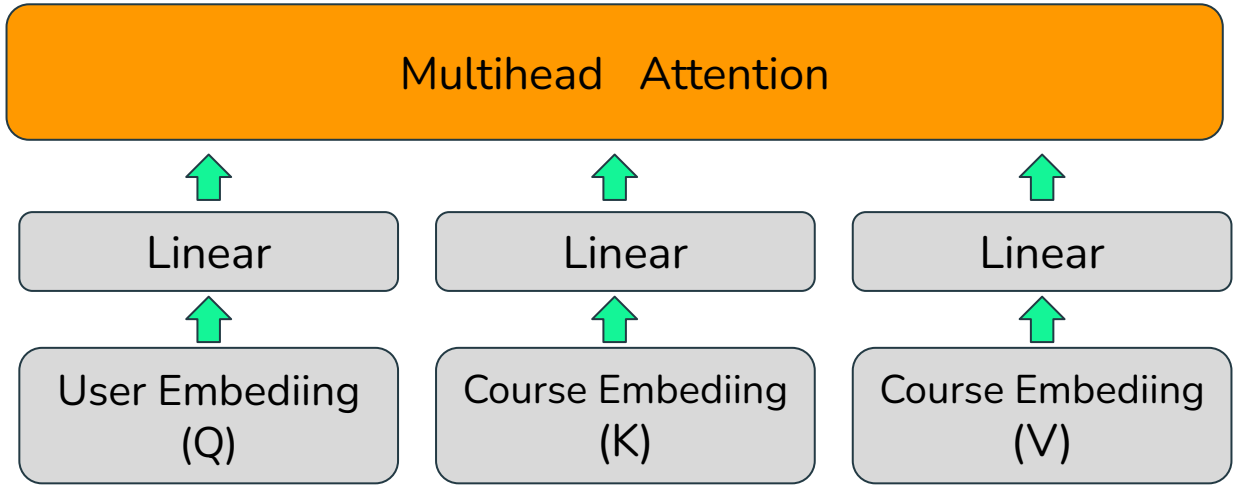
turn text into vectors : text2vec-base-chinese SentenceModel



Second Part :

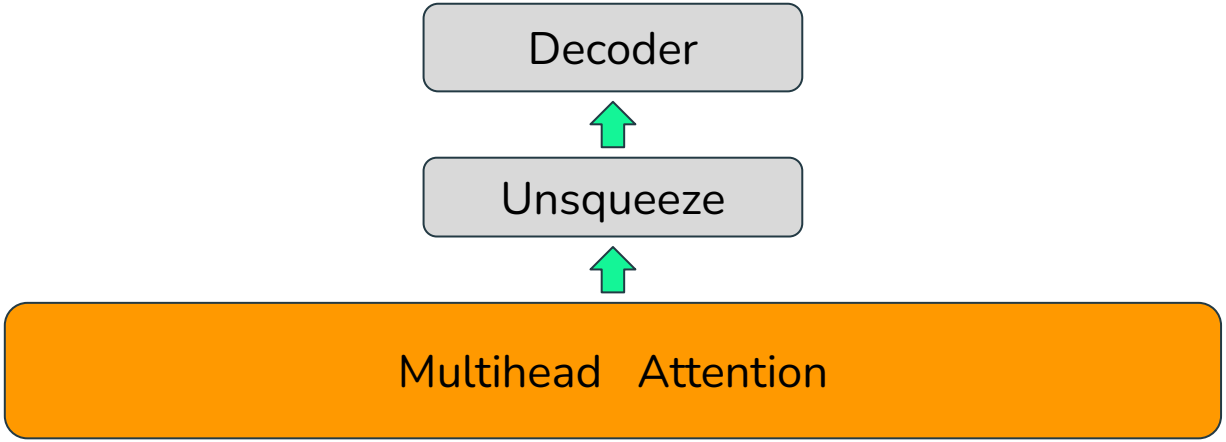
Model Structure

Multihead Attention on the embedding



number of heads : 4

Model 1 : seq2seq



Model 1 : seq2seq

The output of the decoder is the course prediction.

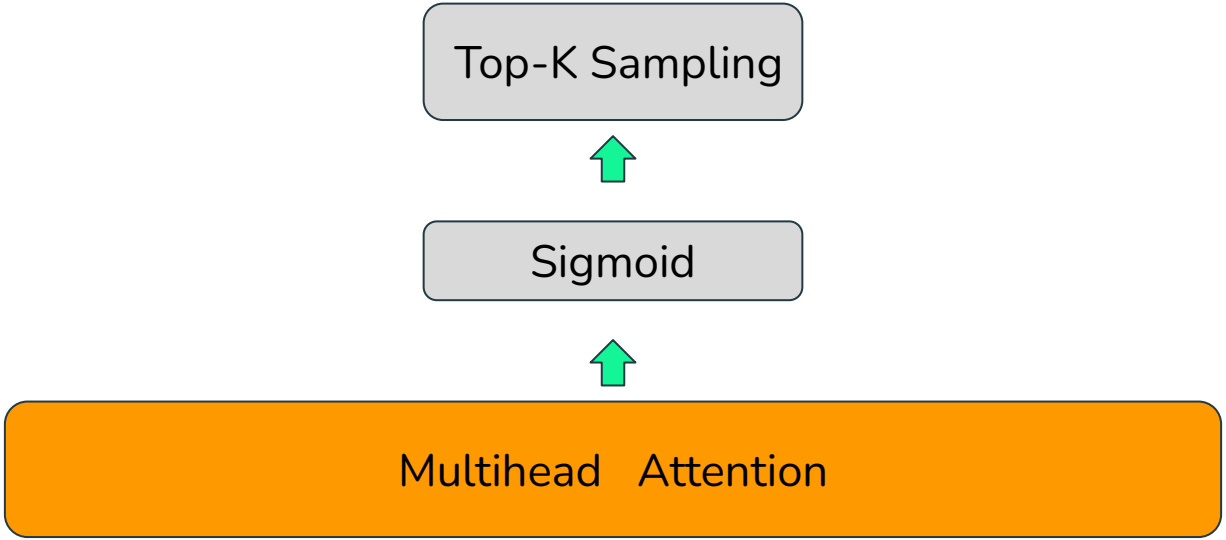
As for decoder, we try two types of decoder : mT5-small (transformer) and LSTM decoder.

In unseen domain for predicting courses, after 10 epochs, mT5-small decoder and LSTM decoder have similar performance on training set and validation set.

As for performance,

0.08 on training set, and 0.01 on validation set.

Model 2 : top-k sample value



Model 2 : top-k sample value

In this model, we convert the original task into a binary classification problem, that is, for labels, must be either 0 or 1. In training, we use binary crossentropy as our loss function, compute loss with label and prediction.

As for embedding, we train course embedding and user embedding from scratch.

Then, the output of model will pass a sigmoid function to get the probability of being bought for every class. We choose top-50 for our prediction.

0.21 on training set, 0.11 on validation set and 0.085 on test set.

Model 2 (cond.) postprocessing : k-nearest neighbors

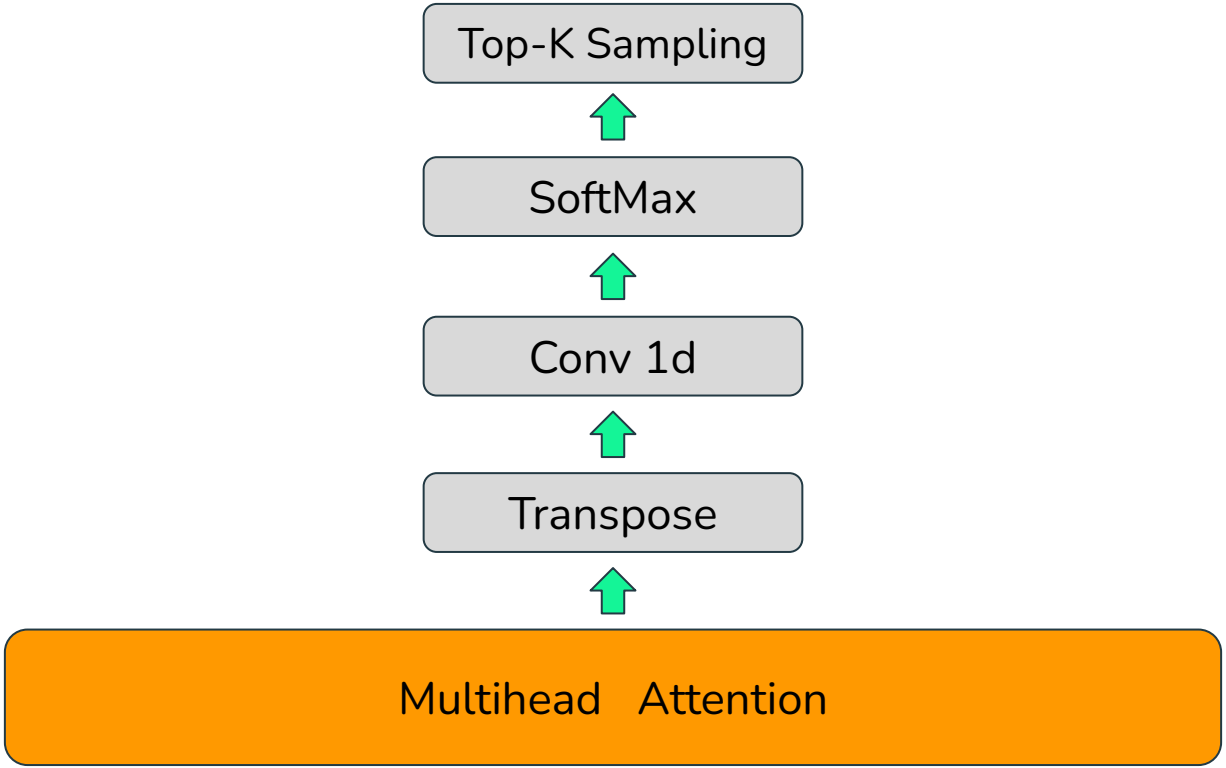
We find that there are some classes which aren't in the training set.

We try to represent these unseen classes and some rarely seen classes with other common classes with high similarity.

Before top-k sampling, we represent the score these unseen and rarely seen classes as the mean score of k common classes with highest similarity.

After postprocessing, score on test set improved to 0.089

Model 3 : top-k sample value



Model 3 : top-k sample value

In this model, like the model 2, we convert the original task into a binary classification problem and using use binary crossentropy as our loss function.

Unlike model 2, we use text2vec to turn user and course data into embeddings.

Then, the output of model will pass a softmax function to get the probability of being bought for every class. We choose top-50 for our prediction.

0.16 on training set, 0.10 on validation set and 0.08 on test set.

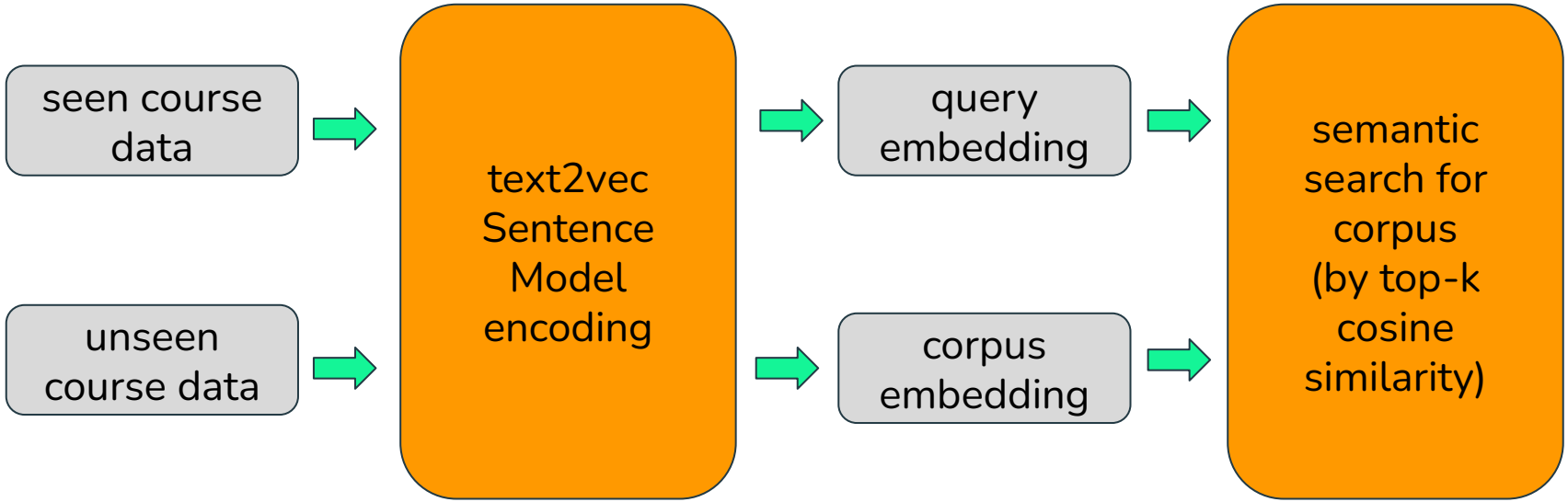
Course Similarity

Why course similarity?

There are 728 courses in total. Yet, in the training set, not all these courses appear in the data. We cannot get the relation between these unseen courses, users and other courses only by training model.

Course Similarity

CoSENT (cosine sentence)



Model 3 (cond.) with course similarity

This time, we consider those courses don’t appear in the training set.

In the data preprocessing, we have already get the similarity between courses. Then, we add some unseen courses which have high similarity with original labels into label.

By doing so, we can use the information of those unseen courses to help prediction. And we also get improvement after using this strategy.
0.25 on training set, 0.133 on validation set and 0.114 on test set.

Third Part :

Result Analysis

Results on **unseen course**

	Model 1 (Seq2Seq)	Model 2 (Top-K Sampling)	Model 2 (cond.) (with k-nearest neighbor)
Training Score	0.08	0.21	0.21
Validation Score	0.01	0.11	0.11
Test Score	N/A	0.085	0.089

	Model 3 (Top-K Sampling)	Model 3 (cond.) (with course similarity)
Training Score	0.16	0.25
Validation Score	0.10	0.133
Test Score	0.08	0.114

Other tasks

Seen/Unseen subgroup: replace courses with subgroups

Seen user: Same as unseen user, but remove the bought courses at output

Conclusion

Those courses which don’t appear in the training set is a main factor of the performance of model. In model 2 and 3, when we use some method to consider these unseen courses as input, we can get a improvement on performance of our model.

Thanks for listening!!