Solution For NeurIPS Education Challenge 2020 from TAL ML Team

TAL Machine Learning Team[*]
TAL Education Group, Beijing, China

1 Task 1: Predict Student Responses – Right or Wrong

1.1 Model

Task 1 is to predict whether students can answer questions correctly. We borrow ideas from masked language model (MLM) task in Bert. We first create answer sequence for each student and randomly mask the correction labels of some questions. We use Bert and stacked-Bidirectional RNNs to encode answer sequence and make predictions on the masked labels. Input features include sequence of question ids, whether students correctly answer the question (a masked padding is used for those masked questions). For long question sequences, we simply cut it to multiple sequences with maximum length of 100. The framework is shown in Figure 1.

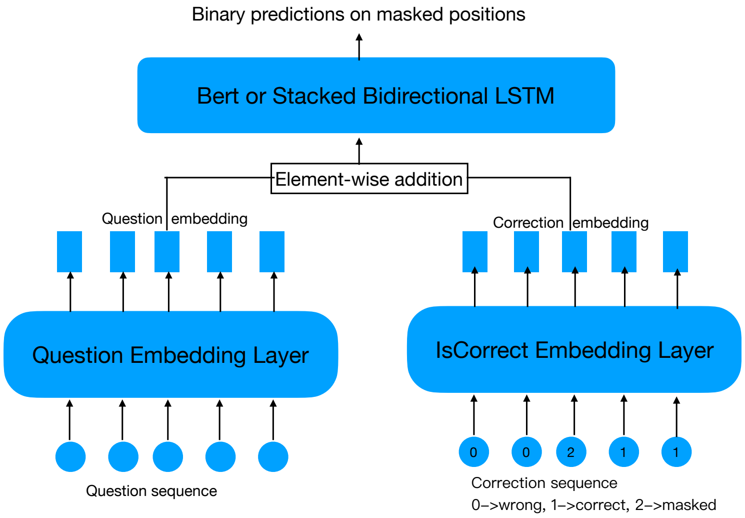
Binary predictions on masked positions
Bert or Stacked Bidirectional LSTM
Element-wise addition
Question embedding
Correction embedding
Question Embedding Layer
IsCorrect Embedding Layer
0
0
2
1
1
Question sequence
Correction sequence
0–>wrong, 1–>correct, 2–>masked

Figure 1: The framework for task 1.

1.2 Parameter

We compute cross entropy loss on masked positions and use Adam as our optimizer. We tuned these parameters:

- Stacked-RNN: hidden neurons $\in \{200, 300, 400\}$, number of RNN layers $\in \{2, 4\}$, learning rate $\in \{0.01, 0.001\}$, batch size $\in \{256, 512, 1024\}$;

[*] Authors: Guowei Xu, Jiaohao Chen, Hang Li, Yu Kang, Tianqiao Liu, Yang Hao, Wenbiao Ding, Zitao Liu. All authors contributed equally to this research.

1st NeurIPS Education Challenge (NeurIPS 2020), Vancouver, Canada.

- Bert: hidden neurons $\in \{200, 400, 600\}$, learning rate $\in \{0.01, 0.001, 0.0001\}$, batch size $\in \{512, 1024\}$, number of layers $\{2, 3, 4, 6\}$, number of heads $in\{4, 6, 8\}$;

1.3 Ensemble

During prediction, we ensemble 124 Bert and Stacked-BiLSTM models. Probability on test data is averaged and positive threshold is 0.5.

2 Task 2: Predict Student Responses – Answer Prediction

2.1 Model

Task 2 is to predict which answer a student will choose given historical data. We first start with both deep learning based CTR models and Transformer-based knowledge tracing models. Deep learning based CTR models predict students' behavior from user embeddings learning from user similarities among students and question embeddings learning from item similarities among questions. We conducted experiments with models including DeepFM, ONN, FibiNet and so on with different hyper-parameters. On the other hand, knowledge tracing models learn students' knowledge states through each timestamp. With the Transformer structures, we randomly masked some labels from the answer sequence for model training. We train the Transformer model in a multi-task manner, with both losses from task 1 and task 2 together. The framework is shown in Figure 2.

2.2 Ensemble

For our best solution, we ensemble three ONN models(training with 5/7/10 epochs) and 5 Transformer KT models. The results of different models are voted with different weights, which are calculated based on each model's performance on the public board.

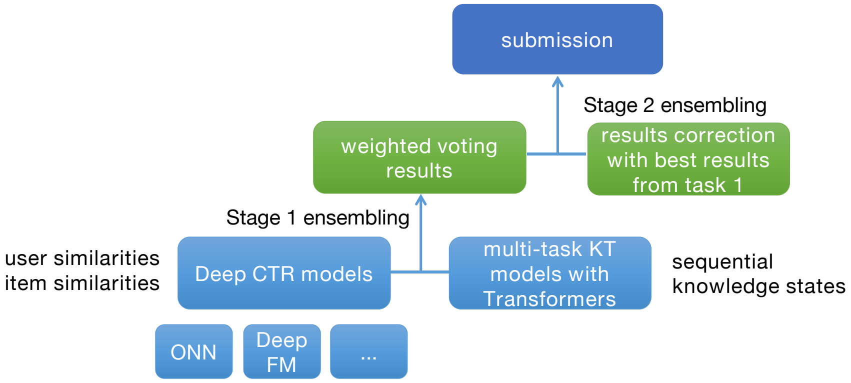

Figure 2: The framework for task 2.

Then we perform results correction with best results from task 1. Since task 1 and task 2 share questions and users, before we choose which answer the student will choose, we first focus on whether she or he can answer the question correctly. If the student will answer it correctly, predicted by task 1, the correct answer will be adopted instead of the voted output from models in task 2.

3 Task 3: Global Question Quality Assessment

In task 3, we were asked to evaluate the question quality.

3.1 Method

The proposed method consists of 5 subcomponents, we utilize the z-score standardization to standardizing scores on the same scale:

1. Similar to the baseline method, we first use the entropy of the options as one of our evaluation metrics.

$$H(choice) = -\sum_{i \in \{A,B,C,D\}} p(i)\log(p(i)) \tag{1}$$

   where i indicates the option "A", "B","C","D", and $p(i)$ indicates the percentage of this option.

2. We extend the option entropy to the right & wrong entropy, which reflects the difficulty of the question from another perspective.

$$H(right\&wrong) = -\sum_{i \in \{0,1\}} p(i)\log(p(i)) \tag{2}$$

3. We can find student confidence score for their individual response from the answer meta data, the addition of this information will help assessing the quality of the questions, thus we utilize the average confidence score of each question as an extra information.
4. We calculate the right & wrong entropy conditioned on the group Id, since different groups stand for different levels of knowledge acquisition, thus we utilize conditional entropy here to incorporate the group information in our method.

$$H(right\&wrong|groupId) = -\sum_{i} p(groupId = i)H(right|groupId = i) \tag{3}$$

5. We calculate the right & wrong entropy conditioned on the quiz Id.

$$H(right\&wrong|quizId) = -\sum_{i} p(quizId = i)H(right|quizId = i) \tag{4}$$

The final metric is as follows:

$$\begin{aligned} Score(i) = {} & H(choice) + 0.7 * H(right\&wrong|group) \\ & + 0.1 * H(right\&wrong|quiz) + H(right\&wrong) - confidence\_score \end{aligned} \tag{5}$$

4 Task 4: Personalized Questions

It's challenging to learn about a student with 10 trial-by-trial personalized questions. Our solution aims to:

1. obtain a robust baseline B_0 at timestamp 0 for each question $*$ student.
2. design an update function F and $B_n = F(B_{n-1}, Q, Correctness)$ after a student answer a question for $n \in [1, 10]$. B_n is the expectations to answer each question correctly by timestamp n.
3. select the most informative questions for each student.

Baseline at Time 0 We calculate the average accuracy of each question as B_0. Hence at timestamp 0, each student is considered to have the same probability to answer a certain question correctly. B_0 is an array of size

Update Function After a student's correctness of a question Q is provided, we should update the expectation of how likely she/he can answer OTHER questions correctly. With train data provided, we can capture the relations of each pair of questions, by (1) Item embeddings learned from Deep CTR methods, or (2) statistical relations such as Pearson correlations, or conditional probabilities of answer task 1 correctly given the result on task 2. We tried both ways and the Pearson correlation matrix performs best. The learning rate of updating depends on how far the current response(1 or 0) is from her/his expectation on last trial.

Question Selection We borrow ideas from active learning to choose questions with the most uncertainty for each student respectively, that is, given the expectation array of a student, choose the question with an expectation close to 0.5 most.