TMU System for SLAM-2018

Masahiro Kaneko[†] **Tomoyuki Kajiwara**[†‡] **Mamoru Komachi**[†]
[†]Graduate School of Systems Design, Tokyo Metropolitan University, Tokyo, Japan
[‡]Institute for Datability Science, Osaka University, Osaka, Japan
kaneko-masahiro@ed.tmu.ac.jp
kajiwara@ids.osaka-u.ac.jp
komachi@tmu.ac.jp

Abstract

We introduce the TMU systems for the second language acquisition modeling shared task 2018 (Settles et al., 2018). To model learner error patterns, it is necessary to maintain a considerable amount of information regarding the type of exercises learners have been learning in the past and the manner in which they answered them. Tracking an enormous learner's learning history and their correct and mistaken answers is essential to predict the learner's future mistakes. Therefore, we propose a model which tracks the learner's learning history efficiently. Our systems ranked fourth in the English and Spanish subtasks, and fifth in the French subtask.

1 Introduction

The second language acquisition modeling (SLAM) is an interesting research topic in the fields of psychology, linguistics, and pedagogy as well as engineering. Popular language learning applications such as Duolingo accumulate learning data of language learners on a large-scale; thus, there has been an increasing interest for SLAM using machine learning using such data. In this study on SLAM, we aim to clarify both: (1) the inherent nature of second language learning, and (2) effective machine learning/natural language processing (ML/NLP) engineering strategies to build personalized adaptive learning systems.

In order to predict the learner's future mistakes, it is important to track a huge history of what and how exercises were solved by that learner and be able to model it. Therefore, we propose a model that can efficiently track a learner's learning history. (Piech et al., 2015; Khajah et al., 2014, 2016)

```
correct: She is my mother and he is my father
learner: she is    mother and he is    fhader
label:    0   0  1    0      0  0  0  1    1
```

Figure 1: An exercise example. Given exercise is a "correct" input. Outputs are "1" each time a learner makes a mistake

2 2018 Duolingo Shared Task on SLAM

We used data from Duolingo in this shared task. Duolingo is the most popular language-learning online application. Learners solve the exercises and this shared task use only 3 type of exercises. Exercise (a) is a *reverse_translate* item, where learners translate written prompt from the language they know into the language they are learning. Exercise (b) is a *reverse_tap* item, where learners construct an answer given a set of words and distractors in the second language. Exercise (c) is a *listen* item, where learners listen and transcribe an utterance in the second language. In this shared task, There are 3 exercise data of the following groups of second language learners:

- English learners (who already speak Spanish)
- Spanish learners (who already speak English)
- French learners (who already speak English)

The Duolingo data set, which contains more than 2 million annotated words, is created from the answers submitted by more than 6,000 learners during their first 30 days. In the related exercises, learners answer questions related to the second language they are learning; thus, they inevitably make various mistakes during the course. In this task, we predict mistakes on word level given an exercise. Figure 1 is an exercise example. Given a "correct" exercise as input a system has to predict labels as output.

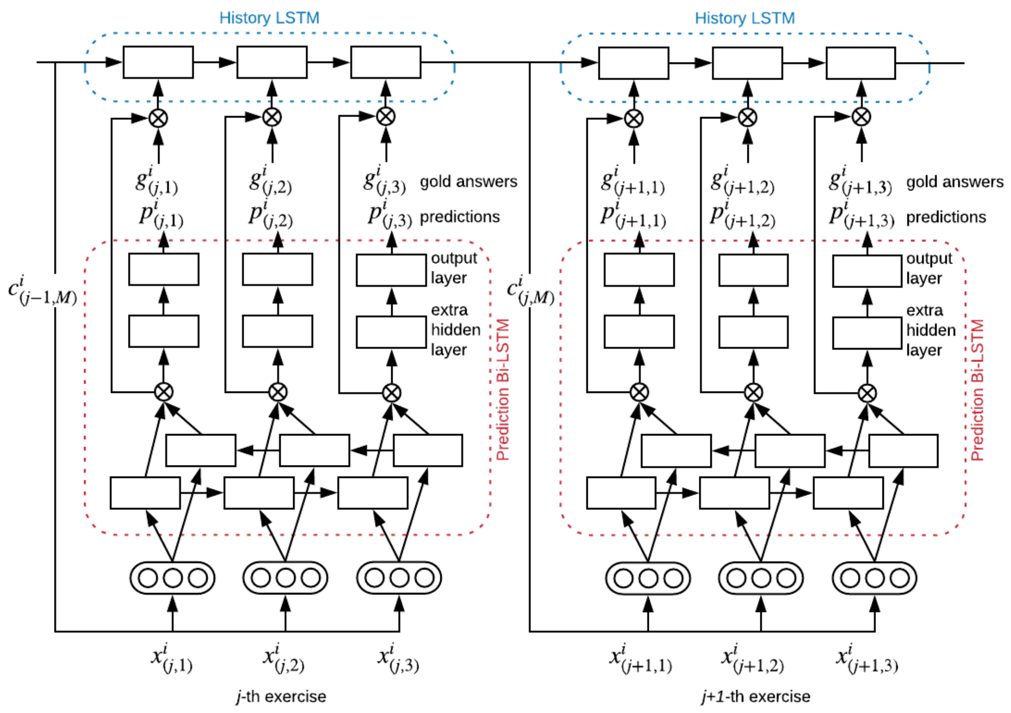
History LSTM
History LSTM
gold answers
predictions
output layer
extra hidden layer
Prediction Bi-LSTM
$c^i_{(j-1,M)}$
$c^i_{(j,M)}$
$g^i_{(j,1)}$ $g^i_{(j,2)}$ $g^i_{(j,3)}$
$p^i_{(j,1)}$ $p^i_{(j,2)}$ $p^i_{(j,3)}$
$g^i_{(j+1,1)}$ $g^i_{(j+1,2)}$ $g^i_{(j+1,3)}$
$p^i_{(j+1,1)}$ $p^i_{(j+1,2)}$ $p^i_{(j+1,3)}$
$x^i_{(j,1)}$ $x^i_{(j,2)}$ $x^i_{(j,3)}$
j-th exercise
$x^i_{(j+1,1)}$ $x^i_{(j+1,2)}$ $x^i_{(j+1,3)}$
$j+1$-th exercise

Figure 2: Architecture of the proposed TMU system.

In general, most tokens are perfect matches; however, the remainder of the tokens are either missing or spelled incorrectly (ignoring capitalization, punctuation, and accents). The former is assigned the label “0” (OK), while the latter is assigned the label “1” (Mistake).

3 TMU System

To track a lot of learner's histories, our proposed TMU system has two components: (1) a base component that predicts whether a learner has made a mistake for the given word in an exercise (Fig. 2, Prediction Bi-LSTM) and (2) a component that tracks a specific learner's information regarding the learned exercises and the words that he or she might have mistaken (Fig. 2, History LSTM). It is expected to track huge history of the learned exercise by inputting the hidden state of the Prediction model to the History LSTM.

In prediction, we receive exercise as input and make predictions on word-level. Using Bi-LSTM for sequence labeling on exercise level, e.g., information as POS tags or dependency edge labels, allows us to share information within each exercise for better prediction. We perform training by feeding input exercises arranged in a chronological order for each learner.

3.1 Features

Table 1 lists all the features used by our system. We use features (1-7) included in the dataset distributed by the task organizers as well as the tracking history (8) (Section 3.3) and labels for language identification (9). We trained a single model with three languages, including English, Spanish, and French; in addition, we used the language identification feature to distinguish them.

There are three types of inputs for the Bi-LSTM. The first input includes word-level features that indicate information changing for each word in an exercise. In particular, word surface and POS are used as word-level features. The second input consists of exercise-level features. In particular, days, session, format, time, and history are used as exercise-level features. The third input includes learner-level features. For this, learner and language features are extracted for each learner.

3.2 Prediction Bidirectional LSTM

We used bidirectional LSTM (Bi-LSTM) to predict whether a learner has mistaken each word in an exercise. The k-th word and POS of the j-th exercise of the i-th learner are converted into $e^i_{(j,k)}$

	Feature	Embeddings	Description
1	Word	$e^i_{(j,k)} \in \mathbb{R}^{d_e \times 1}$	Word Surface
2	POS	$p^i_{(j,k)} \in \mathbb{R}^{d_p \times 1}$	Part of Speech
3	Session	$s^i_j \in \mathbb{R}^{d_s \times 1}$	Lesson, Practice or Test
4	Format	$f^i_j \in \mathbb{R}^{d_f \times 1}$	Reverse_translate, Reverse_tap, or Listen
5	Days	$b^i_j \in \mathbb{R}^{1 \times 1}$	Number of Days Since the Start for Each Learner
6	Time	$t^i_j \in \mathbb{R}^{1 \times 1}$	Amount of Time to Construct and Submit Answers for Each Learner
7	User	$u^i \in \mathbb{R}^{d_u \times 1}$	Unique Identifier for Each Learner
8	History	$c^i_{(j-1,M)} \in \mathbb{R}^{d_c \times 1}$	Last Hidden Layer of History LSTM
9	Language	$l^i \in \mathbb{R}^{d_l \times 1}$	English, Spanish, French

Table 1: Features used in our system. i: Learner's ID; j: Exercise Number of the i-th Learner; k: Word's and POS's index of the j-th Exercise. d_e: Word Embedding Size; d_p: POS Embedding Size; d_s: Session Embedding Size; d_f: Format Embedding Size; d_u: User Embedding Size; d_c: History Embedding Size; M: Total Sentence Length of All Previous Exercises; d_l: Language Embedding Size.

and $p^i_{(j,k)}$ distributed representations, respectively. Further, the session and format of the j-th exercise of the i-th learner are converted into s^i_j and f^i_j distributed representations, respectively. Days and time are represented as b^i_j and t^i_j, respectively. User and language are converted into u^i and l^i distributed representations, respectively. History is the last hidden state $c^i_{(j-1,M)}$ of the History LSTM, which will be described later (Section 3.3).

The inputs of the Bi-LSTM are given as $x^i_{(j,1)}, x^i_{(j,2)}, \cdots, x^i_{(j,N)}$. where, $x^i_{(j,k)} = [e^i_{(j,k)}; p^i_{(j,k)}; s^i_j; f^i_j; b^i_j; t^i_j; c^i_{(j-1,M)}; u^i; l^i]$ is the concatenation of all features and N is the length of the j-th exercise. $x^i_{(j,k)}$ is converted into the forward hidden state $\overrightarrow{h^i_{(j,k)}} \in \mathbb{R}^{d_h \times 1}$ and backward hidden state $\overleftarrow{h^i_{(j,k)}} \in \mathbb{R}^{d_h \times 1}$ using LSTM, where d_h is the hidden size. The final hidden state $h^i_{(j,k)} \in \mathbb{R}^{2d_h \times 1}$ is acquired by concatenating $\overrightarrow{h^i_{(j,k)}}$ and $\overleftarrow{h^i_{(j,k)}}$. Further, $h^i_{(j,k)}$ is fed into the extra hidden layer:

$$\hat{h}^i_{(j,k)} = \mathrm{ReLU}(W_h h^i_{(j,k)} + b_h) \qquad (1)$$

where $\hat{h}^i_{(j,k)} \in \mathbb{R}^{d_{\hat{h}} \times 1}$ is an extra hidden layer output, $W_h \in \mathbb{R}^{d_{\hat{h}} \times d_h}$ is a weighting matrix, and $b_h \in \mathbb{R}^{d_{\hat{h}} \times 1}$ is a bias. The extra hidden layer output $\hat{h}^i_{(j,k)}$ is linearly transformed using the output layer as follows and the probability distribution $p^i_{(j,k)} \in \mathbb{R}^{t \times 1}$ of the true/false tag is acquired using the softmax function, where t is the size of the tag, which is set to 2 in our study.

$$p^i_{(j,k)} = \mathrm{softmax}(W_{\hat{h}} \hat{h}^i_{(j,k)} + b_{\hat{h}}) \qquad (2)$$

where $W_{\hat{h}} \in \mathbb{R}^{t \times d_{\hat{h}}}$ is a weighting matrix and $b_{\hat{h}} \in \mathbb{R}^{t \times 1}$ is a bias.

3.3 History LSTM

As previously mentioned, to correctly predict each learner's mistakes, it is important to consider not only the history of learned exercises, but also the learner's answers to exercises. Thus, the History LSTM tracks all previous information regarding the learned exercises and how they were answered by each learner.

For each j-th exercise, $o^i_{(j,1)}, o^i_{(j,2)}, \cdots, o^i_{(j,N)}$ is given as an input to the j-th History LSTM, where $o^i_{(j,k)} = [h^i_{(j,k)}; g^i_{(j,k)}]$. $h^i_{(j,k)}$ (Section 3.2) is considered as information about the j-th exercise of the i-th learner and $g^i_{(j,k)} \in \mathbb{R}^{1 \times 1}$ is the gold answer of the i-th learner to the j-th exercise. In addition, the first hidden state and cell memory of the j-th History LSTM is initialized with the last hidden state and cell memory of the previous j-1-th History LSTM. The hidden state $c^i_{(j,1)}$ is created from $o^i_{(j,1)}$ using the LSTM for the next step of the Prediction Bi-LSTM.

3.4 Training

The objective function is defined as follows:

$$L_\theta = \frac{1}{|D|} \sum_{(x,y) \in D} \log p(y|x;\theta) \qquad (3)$$

where D is the training data and θ represents model parameters. We use Backpropagation Through Time (BPTT) for training.

In general, low-frequency words are replaced by *unk* word to learn *unk* vector. However, in our study, unknown words appear not because they

Language	Train	Dev	Test
English	936,782	3,000	114,586
Spanish	824,899	3,000	93,145
French	367,402	3,000	41,753

Table 2: Number of exercises for each language.

have low-frequency, but because they have not been learned yet. Hence, we use words that appear for the first time in an exercise to be replaced by *unk* word to learn *unk* vector. In addition, we use words without unk replacement to track the history for the History LSTM.

The final loss is calculated as follows:

$$\overline{L_\theta} = \alpha L_\theta^{unk} + (1-\alpha) L_\theta^{orig} \quad (4)$$

where αL_θ^{unk} is calculated by replacing the word appearing for the first time with *unk*, while $(1-\alpha)L_\theta^{orig}$ is calculated using this word itself. In particular, α expresses the degree of emphasis placed on *unk* and a learned word. For example, when a word "Japanese" appears for the first time, then:

Original exercise: *I am Japanese*
Replaced by *unk*: *I am* $<unk>$

If the *unk* does not exist in any exercise, $\overline{L_\theta}$ has the same value as L_θ^{orig}.

3.5 Testing

During our test, predictions were made on exercises of the test data arranged in chronological order for each learner. We update History LSTM using output and hidden state of Prediction Bi-LSTM. Test data does not have gold answers unlike training data. Hence, each system used its own converted probability outputs of the Prediction Bi-LSTM component with arg max as gold answers.

In addition, we performed ensemble predictions. The parameters of ensemble models are initialized with different values. As the final prediction result, we used the average of the probability outputs of each Prediction Bi-LSTM. Each system used its own converted probability outputs of the Prediction Bi-LSTM component as gold answers.

4 Experiments

4.1 Experiment settings

Table 2 shows the number of exercises for train, dev and test data for each language. The hyper parameters of our model are listed in Table 3. All words that appeared in the training data were included in the vocabulary. Preliminary experiments showed that the AUROC of the one model trained on data of three languages was higher than those models trained for each language. Therefore, we trained a single model with three language tracks, including English, Spanish and French. Especially, AUROC increased for low-resource French language.

Parameter	Value
d_e: Word Embedding Size	100
d_p: POS Embedding Size	20
d_s: Session Embedding Size	20
d_f: Format Embedding Size	20
d_u: User Embedding Size	50
d_l: Language Embedding Size	20
d_c: Hidden Size (History)	200
d_h: Hidden Size (Prediction)	100
$d_{\hat{h}}$: Extra Hidden Size	50
Minibatch size	32
BPTT	18
Optimizer	Adadelta
Learning rate	0.1
Initialization parameters	[-0.1, +0.1]
α, Eq. (4)	0.01
Dev, (Section 3.5)	3,000
Ensemble, (Section 3.5)	10

Table 3: Hyper parameter values.

Each model of the ensemble uses different dev and training sets randomly sampled from the data. In particular, since we needed to evaluate the learning results of Future Days of each learner, we combined the provided official training and dev sets and arranged exercises in chronological order of Days for each learner. Next, we randomly sampled exercises from final learning exercises of learners to create a dev set and the remaining data were used as training data.

4.2 Results

Table 4 lists the results of SLAM for English learners, Spanish learners, and French learners. The systems are ranked by their AUROC. The TMU system ranked fourth in English and Spanish subtasks, while it ranked fifth in the French subtask.

4.3 Analysis of Tracking History

In order to confirm the importance of history tracking, we compared the model that considers history (W/ History Model) with the model that

English	**Spanish**	**French**
0.861 SanaLabs	0.838 SanaLabs	0.857 SanaLabs
0.860 singsound-xushuyao	0.835 alexrich	0.854 singsound-xushuyao
0.858 alexrich	0.834 singsound-xushuyao	0.858 alexrich
0.848 TMU	**0.823 TMU**	0.843 zz
0.846 zz	0.818 zz	**0.839 TMU**
0.841 Cam	0.807 Cam	0.834 Cam
0.828 btomosch	0.802 btomosch	0.822 btomosch
0.821 nihalnayak	0.801 LambdaLearning	0.815 LambdaLearning
0.821 LambdaLearning	0.790 Grotoco	0.813 Grotoco
0.816 Grotoco	0.790 nihalnayak	0.811 nihalnayak
0.815 jilljenn	0.788 ymatusevich	0.808 jilljenn
0.813 ymatusevich	0.787 jilljenn	0.808 ymatusevich
0.796 renhk	0.773 renhk	0.806 caseykennington
0.787 zlb241	0.745 SLAM_baseline	0.795 renhk
0.773 SLAM_baseline	0.681 zlb241	0.770 SLAM_baseline

Table 4: SLAM official evaluation results. Systems are ranked by AUROC.

Model	**AUROC**
W/ History Model	0.834
W/O History Model	0.648

Table 5: The history model has an effect to improve AUROC on English subtask.

does not consider history (W/O History Model) on the dev set for English. The W/O History Model used only the Prediction Bi-LSTM component which does not use the history feature. For experiments using this model, we used a single model trained only on the English corpus. The default split of training set and dev set was 824,012 exercises and 115,770 exercises, respectively. Both aforementioned models used the same parameters as listed in Table 3.

Table 5 lists our evaluation results[1]. It can be observed that the AUROC of prediction of the W/ History Model case is considerably higher than that of the W/O History Model. As we expected, it is important to consider what learner have learned in the past and how they responded to it in order to improve future predictions.

5 Conclusion

In this study, we described the TMU system for the 2018 SLAM Shared Task. Our system is based on RNN; It has two components: (1) Bi-LSTM for predicting learners' error and (2) LSTM for tracking learners' learning history.

In this work, we have not used any language-specific information. As future work, we plan to exploit additional data for each language, such as pre-trained word representations, n-grams, and character-based features. Additionally, we hope to incorporate word difficulty features (Kajiwara and Komachi, 2018). In particular, the more complex a word is, the more difficult it likely is to be learned.

[1]The performance is slightly different from the one reported in Table 3 because of the difference in models and ensembling.

References

T. Kajiwara and M. Komachi. 2018. Complex Word Identification Based on Frequency in a Learner Corpus. In *Proceedings of The 13th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*.

M. Khajah, R. V. Lindsey, and M. C. Mozer. 2016. How Deep Is Knowledge Tracing? *CoRR*.

M. Khajah, R. Wing, R. V. Lindsey, and M. C. Mozer. 2014. Integrating Latent-Factor and Knowledge-Tracing Models to Predict Individual Differences in Learning. In *Educational Data Mining 2014*. Citeseer.

C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. Guibas, and J. Sohl-Dickstein. 2015. Deep Knowledge Tracing. In *Advances in Neural Information Processing Systems 28*, pages 505–513.

B. Settles, C. Brust, E. Gustafson, M. Hagiwara, and N. Madnani. 2018. Second Language Acquisition Modeling. In *Proceedings of the NAACL-HLT Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*.