TMU System for SLAM-2018

†Masahiro Kaneko ††Tomoyuki Kajiwara † Mamoru Komachi kaneko-masahiro@ed.tmu.ac.jp
†: Tokyo Metropolitan University ‡: Osaka University

Second language acquisition modeling (SLAM)

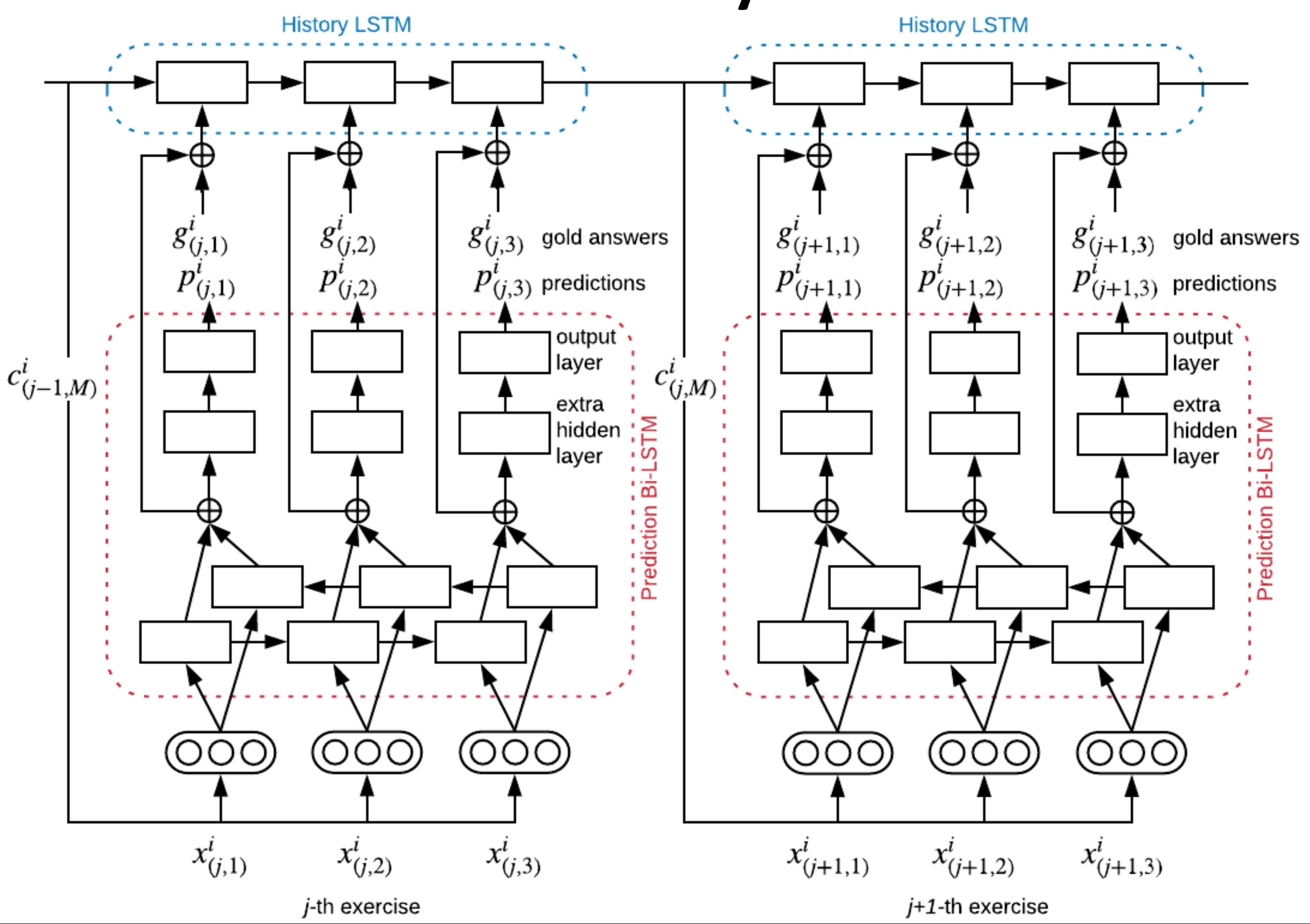
This task aims to predict future mistakes for each user to help users to efficiently learn foreign language

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

Strategy of the TMU system

To predict the learner's future mistakes, it is important to track a history of what and how exercises were solved by that learner.
TMU system has two components:
(1) Prediction Bi-LSTM: predicts whether a learner has made a mistake for the given word in an exercise
(2) History LSTM: tracks a specific learner's information regarding the learned exercises and the words that he or she might have mistaken

Architecture of the TMU system



Experiment settings

	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	Language	$l^i \in \mathbb{R}^{d_l \times 1}$	English, Spanish, French
9	History	$c^i_{(j-1,M)} \in \mathbb{R}^{d_c \times 1}$	Last Hidden Layer of History LSTM

Features used in TMU system

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

Number of exercises for each language.

- A single model with three language tracks, including English, Spanish and French
- Ensemble uses different dev and training sets randomly sampled from the data

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	0.01
Dev	3,000
Ensemble	10

Hyper parameters

Training

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

The objective function

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

Final loss

Original exercise: *I am Japanese*
Replaced by *new*: *I am <new>*

Words that appear for the first time in an exercise are replaced by *new* word to learn *new* vector

Analysis of Tracking History

It is important to consider what learner have learned in the past and how they responded to it in order to improve future predictions

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

The history model improved AUROC on English subtask

SLAM evaluation results. AUROC.

English	Spanish	French
0.861 SanaLabs	0.838 SanaLabs	0.857 SanaLabs
0.861 singsound	0.835 NYU	0.854 singsound
0.859 NYU	0.835 singsound	0.854 NYU
0.848 TMU	**0.823 TMU**	0.843 CECL
0.846 CECL	0.818 CECL	**0.839 TMU**
0.841 Cambridge	0.807 Cambridge	0.834 Cambridge

There are few teams that directly model the learning history as sequential data. SanaLabs also model as sequential data. singsound divide features into several types and build encoder for each of them. On the other hand, we consider everything in one model.

Conclusion & Future work

Our system is based on RNN;
It has two components:
(1) Bi-LSTM for predicting learners' error
(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