CLUF: a Neural Model for Second Language Acquisition Modeling

Shuyao Xu
Singsound Inc.
Beijing, China
xushuy@singsound.com

Jin Chen
Singsound Inc.
Beijing, China
chenjin@singsound.com

Long Qin
Singsound Inc.
Beijing, China
qinlong@singsound.com

Abstract

Second Language Acquisition Modeling is the task to predict whether a second language learner would respond correctly in future exercises based on their learning history. In this paper, we propose a neural network based system to utilize rich contextual, linguistic and user information. Our neural model consists of a Context encoder, a Linguistic feature encoder, a User information encoder and a Format information encoder (CLUF). Furthermore, a decoder is introduced to combine such encoded features and make final predictions. Our system ranked in first place in the English track and second place in the Spanish and French track with an AUROC score of 0.861, 0.835 and 0.854 respectively.

1 Introduction

Education systems that can adapt to the presenting of educational materials according to students' personal learning needs have great potential. Specifically, in the area of second language learning, we try to predict whether the learning materials are too easy or too hard for language learners. Therefore, we study the Second Language Acquisition Modeling (SLAM) task to build a model of the language learning process.

Bayesian Knowledge Tracing (BKT) (Corbett and Anderson, 1994; Pardos and Heffernan, 2010; Pelánek, 2017) that models students' knowledge over time is a well-established problem. It takes a Hidden Markov Model (HMM) with binary hidden states to represent knowledge acquisition for each concept separately. BKT had been successfully applied to subjects like mathematics and programming, where a limited number of concepts can be predefined. However, in language learning, it's difficult to define a small number of concepts, especially when the vocabulary size increases over time. Deep Knowledge Tracing (DKT) (Piech et al., 2015; Wilson et al., 2016) is a recent implementation of knowledge tracing which uses Recurrent Neural Networks (RNNs) to model student's learning trace. Although RNNs and its commonly used variants, such as Gated Recurrent Units (Cho et al., 2014) and Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997), are capable of exploring dynamic temporal behavior for a time sequence, it's hard to model extremely long learning history that can range over months even years. Half-life Regression (Settles and Meeder, 2016) is a novel approach for the SLAM task, which combines a psycholinguistic model of human memory with modern machine learning techniques. It had demonstrated state-of-art performance for predicting student recall rates.

Mapping symbols, such as characters or words, into a continuous space is a popular method in natural language processing (Hinton, 1986; Mikolov et al., 2013; Pennington et al., 2014; Mikolov et al., 2017). It achieved remarkable success in many tasks, for example, neural language modeling (Bengio et al., 2003; Collobert and Weston, 2008; Mikolov et al., 2010), machine translation (Sutskever et al., 2014; Bahdanau et al., 2015), text classification (Lai et al., 2015; Zhang et al., 2015; Conneau et al., 2017), sentiment analysis (dos Santos and Gatti, 2014; Poria et al., 2015) and machine reading comprehension (Xiong et al., 2017; Hu et al., 2017). In this work, we introduce a similar neural approach for the SLAM task, where we use neural encoders to extract features from each exercise as well as metadata about student and session. To be specific, we build a Context encoder, a Linguistic feature encoder, a User information encoder and a Format information encoder (CLUF) to calculate high-level representations from characters, words, part-of-speech (POS) labels, syntactic dependency labels, user id and country, exercise type, client, etc.

Track	Set	Users	Exercises	Unique Tokens	Positive Ratio (%)	OOV Ratio (%)
en_es	Train	2593	824012	1967	12.6	-
	Dev	2592	115770	1839	14.3	3.4
	Test	2593	114586	1879	-	4.5
es_en	Train	2643	731896	2525	14.1	-
	Dev	2640	96003	2353	15.7	7.6
	Test	2641	93145	2459	-	10.0
fr_en	Train	1213	326792	1941	16.2	-
	Dev	1206	43610	1671	17.6	7.1
	Test	1206	41753	1707	-	5.9

Table 1: The SLAM dataset statistics

2 Dataset

The Duolingo SLAM dataset (Settles et al., 2018) is organized into three language tracks:

- en_es: English learners (who already speak Spanish)
- es_en: Spanish learners (who already speak English)
- fr_en: French learners (who already speak English)

According to Table 1, most tokens (more than 80%) are perfect matches and are given the label 0 for "OK". Tokens that are missing or spelled incorrectly (ignoring capitalization, punctuation, and accents) are given the label 1 denoting a mistake. Across the three language tracks, en_es has the lowest positive ratio, while es_en has the highest out-of-vocabulary (OOV) ratio.

Table 2 shows the features provided with the SLAM dataset. In our system, we used all features except the morphology features and syntactic dependency edges, as we did not get any improvement during experiments. Perhaps it is because that the neural networks already encoded similar information from characters, words and their syntactic dependency labels.

Category	Features
Context	word surface form
Linguistic	part of speech
	morphology features
	syntactic dependency edges
	syntactic dependency labels
User	user id
	countries
	days in course
Format	client
	session type
	exercise format
	response time

Table 2: Features provided with the SLAM task

3 Method

We used in total four encoders to model the students' learning behavior. Inputs to these encoders are embeddings learned from one-hot representations of raw features. The context encoder consists of a character level LSTM encoder and a word level LSTM encoder. The linguistic feature encoder is also a LSTM model, where POS and syntactic dependency embedding are concatenated together and then fed into a multilayer LSTM unit. At last, user encoder and format encoder are both fully-connected neural networks. The user encoder takes account of user id, users' nationality and other user related information, while the format encoder encodes exercise format, session type, client type and time used for the exercise. The decoder combines the outputs of these encoders and then makes predictions through a sigmoid unit.

3.1 Context Encoder

The context encoder operates at both the word level and the character level. The word level encoding is capable of capturing better semantics and longer dependency than the character level encoding. But learning new words is a key part in language learning. By modeling the character sequence, we may be able to learn certain word

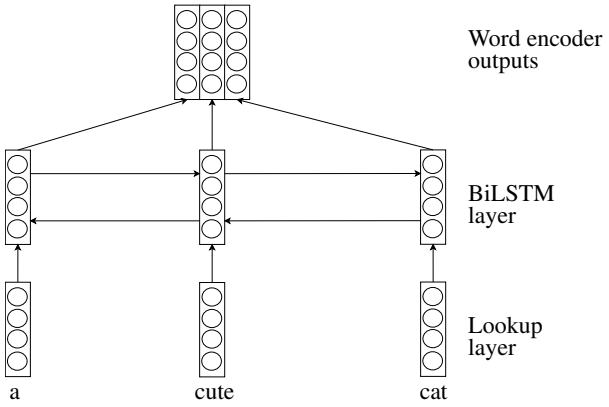

Figure 1: The Word Level Context Encoder

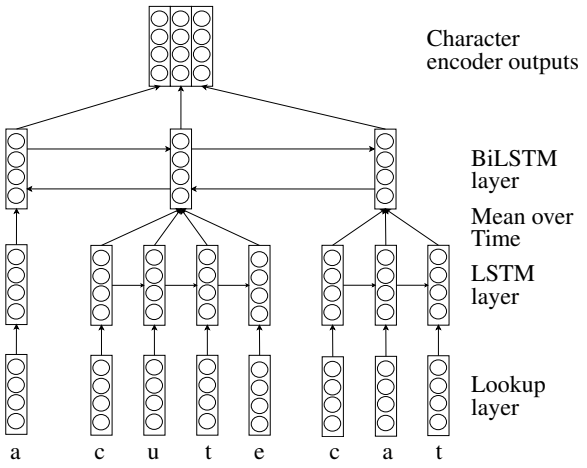

Figure 2: The Character Level Context Encoder

formation rules, therefore partially avoid the OOV problem.

The word level context encoder is a Bidirectional LSTM model. Given a sequence of words represented as one-hot vectors $(w_1,\ w_2,\ ...,\ w_N)$, we can get the word embedding of w_t as

$$x_t = E_w \cdot w_t,$$

where E_w is the word embedding matrix, which is learned during training.

Given the input vector x_t, the forward, backward, and combined activations of the j-th hidden layer are computed as

$$\begin{aligned} f_t^j &= LSTM(f_{t-1}^j,\ f_t^{j-1}) \\ b_t^j &= LSTM(b_{t+1}^j,\ b_t^{j-1}) \\ g_t &= [f_t^{K_0},\ b_t^{K_0}], \end{aligned}$$

where K_0 is the number of layers of the network, $j = 1, 2, ..., K_0$.

The character level context encoder is a hierarchical LSTM model. Given a sequence of one-hot representations of characters in word w_t, $(c_1,\ c_2,\ ...,\ c_M)$, we can get the embedding of c_i as

$$h_i^0 = E_c \cdot c_i,$$

where E_c is the character embedding matrix, which is learned during training.

The outputs of the lookup layer are then fed into a multilayer LSTM unit

$$\begin{aligned} h_i^j &= LSTM(h_{i-1}^j,\ h_i^{j-1}) \\ H_{w_t} &= (h_1^{K_1},\ h_2^{K_1},\ ...,\ h_M^{K_1}), \end{aligned}$$

where K_1 is the number of layers of the LSTM, $j = 1, 2, ..., K_1$.

The mean-over-time (MoT) layer takes H_{w_t} as inputs

$$h_{w_t} = \frac{1}{M} \sum_{i=1}^{M} h_i^{K_1},$$

Then the outputs of the MoT layer (h_{w_1}, h_{w_2}, ..., h_{w_N}) are inputs to a Bidirectional LSTM model,

$$\begin{aligned} \hat{f}_t^j &= LSTM(\hat{f}_{t-1}^j,\ \hat{f}_t^{j-1}) \\ \hat{b}_t^j &= LSTM(\hat{b}_{t+1}^j,\ \hat{b}_t^{j-1}) \\ \hat{g}_t &= [\hat{f}_t^{K_2},\ \hat{b}_t^{K_2}], \end{aligned}$$

where K_2 is the number of layers of the BiLSTM, $j = 1, 2, ..., K_2$.

The final outputs of the context encoder are computed as:

$$O = (o_1,\ o_2,\ ...,\ o_N),$$

where $o_t = g_t + \hat{g}_t$.

3.2 Linguistic Feature Encoder

The linguistic feature encoder is also a LSTM model. Similar to the context encoder, we trained embedding representations of the POS labels and the syntactic dependency labels. The POS embeddings and syntactic dependency embeddings are concatenated together and then fed into a LSTM unit,

$$\begin{aligned} l_t^0 &= [pos_t,\ dep_t] \\ l_t^j &= LSTM(l_{t-1}^j,\ l_t^{j-1}) \\ L &= (l_1^{K_3},\ l_2^{K_3},\ ...,\ l_N^{K_3}), \end{aligned}$$

where pos_t is the POS embedding of word w_t and dep_t is the syntactic dependency label embedding of word w_t. j is the layer index, and we have K_3 layers in this LSTM unit.

3.3 User Encoder

The user encoder is a one-layer fully-connected feedforward network. The encoder takes user metadata as inputs

$$\mu^0 = [u,\ s,\ days]$$
$$\mu^1 = tanh(W_\mu \cdot \mu^0\ +\ b_\mu),$$

where u is the embedding of the user id, s is the embedding of the user's nationality and $days$ is the time since the student started learning this language. W_μ, b_μ are trained network parameters. We used the tanh activation function for the user encoder.

3.4 Format Encoder

Similar to the user encoder, the format encoder is also a one-layer fully-connected feedforward network. The inputs are format, session, client, and the response time,

$$f^0 = [format,\ session,\ client,\ time]$$
$$f^1 = tanh(W_f \cdot f^0 +\ b_f),$$

where W_f, b_f are trainable parameters.

3.5 Decoder

The decoder takes the outputs (O, L, μ^1, f^1) of the context encoder, linguistic encoder, user encoder and format encoder as inputs. The prediction for word w_t in the given sequence (w_1, w_2, ..., w_N) is computed as

$$\nu = \sigma(W_\nu \cdot [\mu^1,\ f^1] +\ b_\nu)$$
$$\gamma_t = \sigma(W_\gamma \cdot [l_t^{K_3},\ o_t] +\ b_\gamma)$$
$$p_t = \sigma(W_p \cdot (\nu \odot \gamma_t) +\ b_p),$$

where W_ν, b_ν, W_γ, b_γ, W_p, and b_p are trainable parameters. For decoding, we used the sigmoid activation function σ.

3.6 Training

The model is trained to minimize the following loss function

$$Loss = -\frac{1}{N}\sum_{t=1}^{N}(\alpha y_t \cdot log(p_t) + (1-\alpha)(1-y_t) \cdot log(1-p_t)),$$

where α is the hyper parameter to balance the negative and positive samples and y_t is the label of the time step t. In our experiment, we set α to 0.7.

Team	AUROC	F1
SanaLabs	0.861	0.561
our model	0.861	0.559
alexrich	0.859	0.468
Masahiro	0.848	0.476
zz	0.846	0.414
Cam	0.841	0.479
btomosch	0.829	0.424
LambdaLearning	0.821	0.389
nihalnayak	0.821	0.376
...	...	...
baseline	0.774	0.190

Table 3: Results of the en_es track.

Team	AUROC	F1
SanaLabs	0.838	0.530
our model	0.835	0.524
alexrich	0.835	0.420
Masahiro	0.824	0.439
zz	0.818	0.390
Cam	0.807	0.435
btomosch	0.803	0.375
LambdaLearning	0.801	0.344
Grotoco	0.791	0.452
...	...	...
baseline	0.746	0.175

Table 4: Results of the es_en track.

4 Experiments and Results

4.1 Experiments

We considered the words that appear less than five times in the training data as unknown token. For students with more than one nationality, only the first one was used.

The embedding size was set to 100, and the Dropout (Srivastava et al., 2014) regularization was applied, where the dropout rate was set to 0.5. We used the Adam optimization algorithm (Kingma and Ba, 2014) with a learning rate of 0.001. The word level context encoder was a two-layer Bidirectional LSTM. The character level context encoder had one LSTM layer for encoding each word and three Bidirectional LSTM layers above the MoT layer. Furthermore, the linguistic

Team	AUROC	F1
SanaLabs	0.857	0.573
our model	0.854	0.569
alexrich	0.854	0.493
zz	0.843	0.487
Masahiro	0.839	0.502
Cam	0.835	0.508
btomosch	0.823	0.442
LambdaLearning	0.815	0.415
Grotoco	0.813	0.502
...	...	...
baseline	0.771	0.281

Table 5: Results of the fr_en track.

Term	en_es	es_en	fr_en
Relative impr (%)	11.24	11.93	9.72

Table 6: The relative improvement over the baseline

encoder was a two-layer LSTM. Both of the user encoder and format encoder were one-layer fully-connected feedforward networks.

4.2 Results

The evaluation metrics for the SLAM task were the Area Under the Receiver Operation Characteristic (AUROC) curve and the F1 score.

As provided in Table 3, Table 4 and Table 5, our model achieved the AUROC score of 0.861, 0.835, and 0.854 and the F1 score of 0.559, 0.524 and 0.569 for the en_es, es_en, and fr_en track, respectively. We ranked in first place in the en_es track and second place in the es_en and fr_en track.

Table 6 shows that CLUF gained significant improvements on all tracks compared to the baseline model. The improvement on the en_es and es_en track were close, while the improvement on the fr_en track was a bit lower. We think this is because the fr_en (327k exercises) track has much less training data than the en_es (824k exercises) and es_en (732k exercises) track.

4.3 Discussion

Our intuition behind CLUF is to factorize raw features into four independent parts: 1) word surface form models the word formation rules; 2) the linguistic encoder is to provide linguistic and syntactic dependency information; 3) the user part explores students' second language acquisition skills over time; 4) the format encoder measures the difficulty level of different exercises on various clients.

Model	AUROC	F1
CLUF	0.846	0.554
LUF	0.775	0.446
CUF	0.843	0.552
CLF	0.813	0.501
CLU	0.779	0.467

Table 7: Encoder analysis. LUF has no context encoder; CUF has no linguistic encoder; CLF has no user encoder; CLU is the model without format encoder.

Table 7 shows the performance of our CLUF model when excluding one of the context, linguistic, user and format encoder. We can see that the performance drops substantially if we don't use the contextual or format features. On the other hand, excluding the linguistic features does not affect the performance much. At last, we can achieve fairly good performance even if we don't use any user information.

5 Conclusion

We presented a neural network based model, CLUF, for the SLAM task. We encoded the contextual, linguistic, user and format features separately. Our system achieved one of the best results in this task. Moreover, our CLUF model was language invariant, as it performed approximately equally well across three language tracks. We further explored how effective each encoder was. We found that the context encoder was the most effective one, while the linguistic encoder was the least effective one.

Acknowledgments

We thank Duolingo and Educational Testing Service for organizing this novel and interesting task and releasing the SLAM dataset.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *international conference on learning representations*.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic lan-

guage model. *Journal of machine learning research*, 3(Feb):1137–1155.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2017. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 1107–1116.

Albert T Corbett and John R Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278.

Geoffrey E Hinton. 1986. Learning distributed representations of concepts. In *Proceedings of the eighth annual conference of the cognitive science society*, volume 1, page 12. Amherst, MA.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Minghao Hu, Yuxing Peng, and Xipeng Qiu. 2017. Reinforced mnemonic reader for machine comprehension. *CoRR, abs/1705.02798*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *AAAI*, volume 333, pages 2267–2273.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2017. Advances in pre-training distributed word representations. *arXiv preprint arXiv:1712.09405*.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Zachary A Pardos and Neil T Heffernan. 2010. Modeling individualization in a bayesian networks implementation of knowledge tracing. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 255–266. Springer.

Radek Pelánek. 2017. Bayesian knowledge tracing, logistic models, and beyond: an overview of learner modeling techniques. *User Modeling and User-Adapted Interaction*, 27(3-5):313–350.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 505–513. Curran Associates, Inc.

Soujanya Poria, Erik Cambria, and Alexander Gelbukh. 2015. Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2539–2544.

Cicero dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78.

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)*. ACL.

Burr Settles and Brendan Meeder. 2016. A trainable spaced repetition model for language learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1848–1858.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Kevin H Wilson, Xiaolu Xiong, Mohammad Khajah, Robert V Lindsey, Siyuan Zhao, Yan Karklin, Eric G Van Inwegen, Bojian Han, Chaitanya Ekanadham, Joseph E Beck, et al. 2016. Estimating student proficiency: Deep learning is not the panacea. In *In Neural Information Processing Systems, Workshop on Machine Learning for Education*.

Caiming Xiong, Victor Zhong, and Richard Socher. 2017. Dynamic coattention networks for question answering. *international conference on learning representations*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.