11-785 Project Midterm Report: Second Language Acquisition Modeling: A Deep Learning Approach.

Xinhe Zhang
Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
xinhez@andrew.cmu.edu
Xinyue Zhang
Department of Institute for Software Research
Carnegie Mellon University
Pittsburgh, PA 15213
xzhang4@andrew.cmu.edu
Xueqian Zhang
Department of Information System Managemnet
Carnegie Mellon University
Pittsburgh, PA 15213
xueqianz@andrew.cmu.edu

November 14, 2019

Abstract

In this project, we are going to examine how effective are the different deep learning architectures in the second language acquisition (SLA) modeling task (Settles et al., 2018). Specifically, given a history of errors made by learners of a second language, the model needs to predict the probability that the user is likely to make a mistake for a specific word in the future. All models are trained and tested with the SLA modeling data set released by Duolingo, a popular online language-learning app, which consists of more than 7M words produced by more than 6K learners of English, Spanish and French from the app.

1 Problem Statement

The emergence of computer-based educational apps presents us with vast amounts of student learning data, and thus, opens the door for personalized education.

However, unlike general science subjects, language learning involves more skills such as the interaction of lexical knowledge, morpho-feature processing, etc., that are not easy to test. Even less effort was made to address beginner studies.

In this project, we are going to examine how effective are different deep learning architectures in the SLA modeling, a sequence to classification task. That is, given a history of errors made by learners of a second language, the model needs to predict the probability that the user is likely to make a mistake for a specific word in the future. The data set consists of 7M tokenized words, labeled with errors made by more than 6.4K learners of English, Spanish, and French, during their first 30 days of learning with Duolingo (an award-winning popular online language-learning app). We are going to tackle this task from an in-depth learning perspective and study how effective could this approach model second language learners' behavior.

2 Literature Review

2.1 Feature Engineering

Effective features that capture how humans learn a language are important to language acquisition modeling. Some of the example features are word features (word frequency, age-of-acquisition children typically learn the word, cognates word), user features (users' motivation, users' diligence), and temporal features (the number of time the exact same sentence had been seen before by the user and users' past experience with the particular instance word) (Rich et al., 2018). Time since the word was last seen, length of sentence the user is learning, and preceding word are also important features for predicting the word level mistakes (Osika et al., 2018).

2.2 Deep Learning Model

Recurrent neural networks (RNNs) have been shown to achieve good results for sequential prediction tasks. Shuyao Xu, Jin Chen, and Long Qin's works have proved that RNN architecture using four types of encoders can do well on language acquisition modeling tasks. The four encoders represent token context, linguistic information, user data, and exercise format, respectively, and the decoder integrated information from all four encoders (Xu et al., 2018). Motivated by the observation that RNNs work well for sequential data, and Gradient Boosting Decision Tree (GBDT) is usually the best performing non-neural model for tabular data, SanaLabs used a combination RNN with LSTM architecture and GBDT ensemble for language acquisition modeling (Osika et al., 2018). Transformer, a novel neural network architecture based on a self-attention mechanism is also proved to work well for language understanding (Vaswani et al., 2017)

3 Method or Proposed Solutions

3.1 Baseline model: Sequence to Sequence (Seq2seq)

We chose a simple Seq2seq model as our baseline implementation. For our baseline solution, we only used the token as the input. The input was first converted to indices through a word2index dictionary. The Seq2seq model consists of a Recurrent Neural Network (RNN) encoder and an RNN decoder. The encoder first processes the input indices with an embedding layer. The output from the embedding layer is then fed into a single hidden layer RNN with 8 hidden neurons. The final hidden state from the encoder, together with the output of the decoder to be generated, is passed into the decoder, which is composed of a single hidden layer RNN with 8 hidden neurons and a dense layer convert the output to a vector of 2 probabilities. We use CrossEntropyLoss to apply the backpropagation. The model is trained for 10 epochs to obtain our final result.

3.2 Transformer features

We decide to implement two models to improve the baseline model, including Transformer and BERT. Transformer, composed of encoder and decoder, is a model based on the Attention model, so the Transformer model will decide which part of the input is more important every time received input through a fully connected layer. For the encoder part, each layer of the Transformer model is made up of two sub-layer, including a multi-head self-attention layer and a fully connected feed-forward layer, and the Transformer model adds a residual connection and normalization for each sub-layer. For the decoder part, every layer includes three sub-layers. First, the sub-layer is a masked multi-head self-attention layer, the second sub-layer is a multi-head attention layer, and the third sub-layer is still a fully connected feed-forward layer. The second sub-layer is not a self-attention layer because the input of it comes from not only the output of encoder but also the output of the previous decoder. While RNN is a model based on time order, the Transformer is based on position order and uses sine and cosine to implement positional encoding. The transformer has less complexity per layer and can parallelize the computation during encoding. What's more, the self-attention layer makes what the Transformer model has learned more explainable from the human perspective.

3.3 BERT/hand-craft features

For future work, we propose to use BERT embeddings to extract features vectors from text data, which will be high-quality feature inputs to downstream models. This is inspired by the observation that BERT produces word representations that are dynamically informed by the words around them, which match how learners learn a new language by understanding the context of each word/phrase. We also propose to incorporate handcrafted features that are generated from a psychological perspective to our model. For example, use the mean or the

Table 1: Aligned reference sequence and the user input.

learner:	wen	can		help
reference:	When	can	I	help
label:	1	0	1	0

Table 2: Summary of the Duolingo SLA modeling data set.

		TRAIN	DEV	TEST
Language	Users	Tokens (Err)	Tokens (Err)	Tokens (Err)
English from Spanish	2.6K	2.6M (13%)	387K (14%)	387K (15%)
Spanish from English	2.6K	2.0M (14%)	289K (16%)	282K (16%)
French from English	1.2K	927k (16%)	138K (18%)	136K (18%)
Overal	6.4K	5.5M (14%)	814K (15%)	804K (16%)

median number of exercises within each learning hour as a feature to represent the user's motivation and use the entropy of the learning frequency distribution overtimes to represent the user's diligence.

4 Data set

Data is collected within the Duolingo app for English, Spanish, and French learners during their first 30 days of studying. In each lesson, the users were presented with several questions prompts, for which they need to input their translations. After the app selects a reference answer based on the string edit distance (Levenshtein, 1966), two token sequences are aligned. A label is then assigned to each token, with 1 indicating the user made a mistake at that index, and 0 if the user is correct. Table 1 illustrates such alignment of the two sequences given prompt "¿Cuándo puedo ayudar?".

Table 2 summaries the data set statistics. Each user data entry is partitioned into three sets, 80% for TRAIN, 10% for DEV, and 10% for TEST. Every entry starts with a 2-line header (marked by the # character): first, the question prompt and second, the user information including user ID, country code, days using Duolingo, platform (web, Android, or iOS), question format (reverse_translation, reverse_tap, or listen) and time taken in seconds. Each data entry groups the tokens from one question prompt. Table 3 illustrates one data entry given prompt "¿Cuándo puedo ayudar?".

Note that we do not have the actual learner responses in this data set, only the closest reference answers are provided.

Table 3: Data entry example.

token ID	token	morpho-syntactic features	label
oMGsnnH/0101	When	ADV PronType=In advmod 4	1
oMGsnnH/0102	can	AUX VerbForm=Fin aux 4	0
oMGsnnH/0103	I	PRON Case=Nom|Number=Sing nsubj 4	1
oMGsnnH/0104	help	VERB VerbForm=Inf ROOT 0	0

Table 4: Baseline model evaluations.

Model	accuracy	AUC	F1
Random	0.500	0.500	0.243
Seq2seq	0.839	0.631	0.047

5 Evaluation

The output of the model should be pairs of token IDs and the probabilities that the learner will make a mistake for this token.

We use three metrics to evaluate model performance. The primary evaluation metric is the area under the ROC curve (AUC). AUC is a well-known performance measurement for classification problems and can be interpreted as the probability that the system will rank a randomly-chosen error above a randomly-chosen non-error. This ranking especially helps with personalized learning, e.g., if we wish to prioritize words or exercises for an individual learner's review based on how likely they are to make mistakes at a given time (Settles et al., 2018).

We also use the classification accuracy and F1 score to assist with the evaluation. The metric accuracy measures how closely the model behaves like the learner. The F1 score, the harmonic mean of precision and recall, is commonly used in similar skewed class labeling tasks (Ng et al., 2013)

Table 4 presents our initial baseline sequence to sequence model results.

6 Timeline

Start Date	End Date	Objective
Sep 23	Oct 7	Project Proposal.
Oct 7	Oct 14	Design data processing and features.
Oct 14	Oct 28	Design and implement baseline model.
Oct 28	Nov 4	Evaluate the baseline model.
Nov 4	Nov 11	Design baseline model improvement.
Nov 11	Nov 18	Implement Transformer model and evaluate.
Nov 18	Nov 25	Implement BERT model and evaluate.
Nov 25	Nov 3	Evaluate the overall result Poster presentation and final project report.

7 Division of Work

7.1 Xinhe Zhang

- Implement the project pipeline.
- Design and implement the baseline model.
- Evaluate the overall result, prepare for poster presentation and write final project report.

7.2 Xinyue Zhang

- Design data processing and features.
- Implement Transformer model and evaluate.
- Evaluate the overall result, prepare for poster presentation and write final project report.

7.3 Xueqian Zhang

- Design data processing and features.
- Implement BERT model and evaluate.
- Evaluate the overall result, prepare for poster presentation and write final project report.

References

V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.

H. Ng, w. siew mei, Y. Wu, C. Hadiwinoto, and J. Tetreault. The conll-2013 shared task on grammatical error correction. pages 1–12, 08 2013.

A. Osika, S. Nilsson, A. Sydorchuk, F. Sahin, and A. Huss. Second language acquisition modeling: An ensemble approach. In *Proceedings of the NAACL-HLT Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*. ACL, 2018.

A. S. Rich, P. J. Osborn, P. D. J. Halpern, A. Rothe, and T. M. Gureckis. Modeling second-language learning from a psychological perspective. In *Proceedings of the NAACL-HLT Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*. ACL, 2018.

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

A. Vaswani, N. Shazeera, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Łukasz Kaiser, and I. Polosukhin. Attention is all you need. In *In Advances in neural information processing systems*, pages 5998–6008, 2017.

S. Xu, J. Chen, and L. Qin. A neural model for second language acquisition modeling. In *Proceedings of the NAACL-HLT Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*. ACL, 2018.