Modeling Second-Language Learning from a Psychological Perspective

Alexander S. Rich   **Pamela Osborn Popp**   **David J. Halpern**
Anselm Rothe   **Todd M. Gureckis**
Department of Psychology, New York University
`{asr443,pamop,david.halpern,anselm,todd.gureckis}@nyu.edu`

Abstract

This is the abstract

1   Introduction

Educational software that aims to teach people new skills, languages, and even academic subjects have become increasingly popular. The widespread deployment of these tools has created interesting opportunities to study the process of learning in extremely large samples of learners in naturalistic situations. The Duolingo shared task on Second Lanuage Acquisition Modeling (SLAM) was a competitive modeling challenge run in early 2018 (Settles et al., 2018). The challenge, organized by Duolingo[1], a popular second language learning app, was to use log data from thousand of users completing millions of exercises predict patterns of future translation mistakes in held-out data. The data set was divided into three sets covering students learning to translate English to Spanish (ES→EN), Spanish to English (ES→EN), and French to English (FR→EN). This paper reports the approach used by our team which ended in third place for the ES→EN data set, second place for the ES→EN data set, and third place for FR→EN.

Our lab's interest in this competition comes from our emphasis on studying human learning and memory from a psychological perspective. While the Duolingo SLAM dataset heavily emphasizes language learning (e.g., syntax, grammar, semantics), it also touches on basic learning and memory phenomena (e.g., retrieval of the translated word from memory). The study of the psychological processes supporting second-language acquisition has long been a topic of interest in the learning and memory literature (Atkinson, 1972b,a; Pavlik and Anderson, 2008, e.g.,).

---

[1] http://duolingo.com

In fact, much of the classic work on psychology of memory involved aspects of verbal language learning, which helped to identify a number of important and robust phenomena in memory including the retention curve and the advantage for spaced over massed practice (Ruth, 1928; Rubin and Wenzel, 1996; Cepeda et al., 2006).

Todd: I want to transition to something about how our approach tried to stick to some psychological principles but i'm not sure... maybe need to wait until al the later sections are fleshed out to lay out what the "concept" is behind our approach.

One idea is to talk about while the research on learnign and memory is important it might be hard to graft those principles into datasets like these because of subject variability, etc... As a result it is unlikely that straightfoward cognitive models will work but instead need to combine maching learning methods that can create much more complex rules and decisions with features that reflect aspects of what makes something easy or hard to learn.

(Tubridy et al., in review(a)) (Tubridy et al., in review(b)) (Corbett and Anderson, 1995) (Henry L. Roediger and Karpicke, 2006; Anderson et al., 1994; Mozer et al., 2009)

2   Task Approach

We approached the task as a binary classification problem over instances (i.e., single words within an exercise). Our solution can be divided into two components—constructing a set of features that is highly informative about whether the user will answer an instance correctly, and designing a model that can achieve high performance using this feature set.

2.1   Feature Engineering

We used a variety of features, including features directly present in the training data, features con-

structed using the training data, and features that use information external to the training data. Except where otherwise specified, categorical variables were one-hot encoded.

2.1.1 Exercise features

We encoded the exercise number, client, session, format, and duration (i.e., number of seconds to complete the exercise), as well as the time since the user started using Duolingo for the first time.

2.1.2 Word features

Using spaCy[2], we lemmatized each word to produce a root word. Both the root word token and the original token were used as categorical features. Due to their high cardinality, these features were not one-hot encoded but were preserved in single columns and handled in this form by the model (as described below).

Along with the tokens themselves we encoded ==each== instance word's part of speech, morphological features, and dependency edge label. (We noticed that some words in the original dataset were paired with the wrong morphological features, particularly near where punctuation had been removed from the sentence. To fix this, we reprocessed the data using Google SyntaxNet[3].)

We also encoded word length and several word characteristics gleaned from external data sources. The word frequency effect suggests that uncommon words are harder to process than common words; readers will look longer at low-frequency words and perform worse in word-identification tasks for these than for high-frequency words (Rayner, 1998). We therefore included a feature that encoded the frequency of each word in the language being acquired, calculated from Speer et al. (2017). Additionally, cognates, or words sharing a common linguistic derivation, are easier to learn than words with dissimilar translations (De Groot and Keijzer, 2000). As an approximate measure of linguistic similarity, we used the Levenshtein edit distance between the word tokens and their translations scaled by the length of the longer word. We found translations using Google Translate[4] and calculated the Levenshtein distance to reflect the letter-by-letter similarity of the word and its translation (Hyyrö, 2001). ==DAVID FILL IN HERE WITH AOA.==

---
[2]https://spacy.io/
[3]https://github.com/ljm625/syntaxnet-rest-api
[4]https://cloud.google.com/translate/

2.1.3 User features

Just as we did for word tokens, we encoded the user ID as a single-column, high-cardinality feature. We also calculated several other user-level features.
==ANSELM HERE==
Feature 1: Motivation
Feature 2: Consistency

2.1.4 Positional features

To account for the effects of surrounding words on the difficulty of an instance, we created several features related to the instance word's context in the exercise. These included the token of the previous word, the next word, and the instance word's root in the parse tree, all stored in single columns as with the instance token itself. We also included the part of speech of each of these context words as additional features. When there was no previous word, next word, or parse root word, a special `None` token or `None` part of speech was used.

2.1.5 Temporal features

A user's probability of succeeding on an instance is likely related to their prior experience with that instance. To capture this, we calculated several features related to past experience. We encoded the number of times the current exercise's exact sentence had been seen before by the user. We also encoded a set of features recording past experience with the particular instance word. These features were encoded separately for the instance token and for the instance root word created by lemmatization. (Perruchet and Vinter, 1998; Chun and Phelps, 1999) ==chun is good, maybe something other that PARSER as it is more a statistcal learning model==

For each token (and root) we tracked user performance through four weighted error averages. At the user's first encounter of the token, each error term E starts at zero. After an encounter with an instance of the token with label L, it is updated according to the equation

$$E \leftarrow E + \alpha(L - E)$$

where α determines the speed of error updating. The four feature weighted error terms use $\alpha = \{.3, .1, .03, .01\}$, allowing both short-run and long-run changes in a user's error rate with a token to be tracked. Note that in cases where a token appears multiple times in an exercise, a sin-

gle update of the error features is conducted using the mean of the token labels. Along with the error tracking features, for each token we calculated the number of labeled, unlabeled, and total encounters; time since last labeled encounter and last encounter; and whether the instance is the first encounter with the token.

In the training data, all instances are labeled as correct or incorrect, so the label for the previous encounter is always available. In the test data, labels are unavailable, so predictions must be made using a mix of labeled and unlabeled past encounters. To generate training-set features that are comparable to test-set features, we selectively ignored some labels when encoding temporal features on the training set. Specifically, for each user we first calculated the number of exercises n in the true test set. Then, when encoding the features for each instance, we selected a random integer r in the range $[1, n]$, and ignored labels in the prior r exercises. That is, we encode features for the current instance as though other instances in those prior exercises were unlabeled, and ignore updates to the error averages from those exercises. The result of this process is that each instance in the training set is encoded as though it were between one and n exercises into the test set.

2.2 Modeling

After featurizing the training data, we trained gradient boosting decision tree (GBDT) models to minimize log loss. GBDT works by iteratively building regression trees, each of which seeks to minimize the residual loss from prior trees. This allows it to capture non-linear effects and high-order interactions among features. We used the LightGBM[5] implementation of GBDT (Ke et al., 2017).

For continuous-valued features, GBDT can split a leaf at any point, creating different predicted values above and below that threshold. For categories that are one-hot encoded, it can split a leaf on any of the category's features. This means that for a category with thousands of values, potentially thousands of tree splits would be needed to capture its relation to the target. Fortunately, LightGBM implements an algorithm for partitioning the values of a categorical feature into two groups based on their relevence to the current loss, and create a single split to divide those groups (Fisher, 1958).

---

[5] http://lightgbm.readthedocs.io/

Parameter	fr_en	en_es	es_en	all
num_leaves	256	512	512	1024
learning_rate	.05	.05	.05	.05
min_data_in_leaf	100	100	100	100
num_boost_rounds	750	650	600	750
cat_smooth	200	200	200	200
feature_fraction	.7	.7	.7	.7
max_cat_threshold	32	32	32	64

Table 1: Parameters of final LightGBM models. See LightGBM documentation for more information; all other parameters were left at their default values.

Thus, as alluded to above, high-cardinality features like token and user were encoded as single columns and handled as categories by LightGBM.

We trained a model for each of the three language tracks of en_es, es_en, and fr_en, and also trained a model on the combined data from all three tracks, adding an additional "language" feature. Following model training, we averaged the predictions of each single-language model with that of the all-language model to form our final predictions.

To tune model hyper-parameters and evaluate the usefulness of features, we first trained the models on the "train" data set and evaluated them on the "dev" data set. Once the model structure was finalized, we trained on the combined "train" and "dev" data and produced predictions for the "test" data. The LightGBM hyperparameters used for each model are listed in Table 1.

2.3 Performance

The AUROC of our final predictions was .8585 on en_es, .8350 on es_en, and .8540 on fr_en. We did not attempt to optimize the model's F1 score, but the model's F1 score could likely be improved (at the cost of increased log loss) by finding the rescaling of the "dev" predicted probabilities that maximized the F1 score at the 0.5 threshold, and applying this rescaling to the "test" predicted probabilities.

3 Feature Removal Experiments

4 Acknowledgments

References

Michael C Anderson, Robert A Bjork, and Elizabeth L Bjork. 1994. Remembering can cause forgetting: Retrieval dynamics in long-term memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 20:1063–1087.

R.C. Atkinson. 1972a. Ingredients for a theory of instruction. *American Psychologist*, 27:921–931.

R.C. Atkinson. 1972b. Optimizing the learning of a second-language vocabulary. *Journal of Experimental Psychology*, 96:124–129.

N.J. Cepeda, H. Pashler, E. Vul, J.T. Wixted, and D. Rohrer. 2006. Distributed practice in verbal recall tasks: A review and quantitative synthesis. *Psychological Bulletin*, 132(3):354–380.

M.M. Chun and E.A. Phelps. 1999. Memory deficits for implicit contextual information in amnesic subjects with hippocampal damage. *Nature Neuroscience*, 2(9):844–847.

AT Corbett and JR Anderson. 1995. Knowledge tracking: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4:253–278.

Annette De Groot and Rineke Keijzer. 2000. What is hard to learn is easy to forget: The roles of word concreteness, cognate status, and word frequency in foreign-language vocabulary learning and forgetting. *Language Learning*, 50(1):1–56.

Walter D Fisher. 1958. On grouping for maximum homogeneity. *Journal of the American statistical Association*, 53(284):789–798.

III Henry L. Roediger and Jeffrey D. Karpicke. 2006. Test-enhanced learning: Taking memory tests improves long-term retention. *Psychological Science*, 17(3):249–255. PMID: 16507066.

Heikki Hyyrö. 2001. Explaining and extending the bit-parallel approximate string matching algorithm of myers. *Technical report in Journal of the ACM*, page 408.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, pages 3149–3157.

M. C. Mozer, H. Pashler, N. Cepeda, R. Lindsey, and E. Vul. 2009. Predicting the optimal spacing of study: A multiscale context model of memory. In *Advances in Neural Information Processing Systems 22*, pages 1321–1329, La Jolla, CA. NIPS Foundation.

P.I. Pavlik and J.R. Anderson. 2008. Using a model to compute the optimal schedule of practice. *Journal of Experimental Psychology: Applied*, 14(2):101–117.

Perruchet Perruchet and Annie Vinter. 1998. Parser: A model for word segmentation. *Journal of Memory and Language*, 39:246–263.

Keith Rayner. 1998. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, 124(3):372.

D.C. Rubin and A.E. Wenzel. 1996. One hundred years of forgetting: A quantitative description of retention. *Psychological Review*, 103(4):734–760.

T.C. Ruth. 1928. Factors influencing the relative economy of massed and distributed practice in learning. *Psychological Review*, 35:19–45.

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.

Robert Speer, Joshua Chin, Andrew Lin, Sara Jewett, and Lance Nathan. 2017. Luminosoinsight/wordfreq: v1.7.

S. Tubridy, D. Halpern, L. Davachi, and T.M. Gureckis. in review(a). A neuro-cognitive model for predicting the fate of individual memories. In *Proceedings of the 40th Annual Conference of the Cognitive Science Society*.

S. Tubridy, D. Halpern, V. Wang, C. Gasser, P. Osborn-Popp, L. Davachi, and T.M. Gureckis. in review(b). Knowledge tracing using the brain. In *Proceedings of Educational Data Mining (EDM) 2018*.