Modeling Second-Language Learning from a Psychological Perspective

Alexander S. Rich **Pamela J. 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 academic subjects have become increasingly popular. The wide-spread 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 Lanugage 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 thousands of users completing millions of exercises to predict patterns of future translation mistakes in held-out data. The data set was divided into three sets covering Spanish speakers learning English (`en_es`), English speakers learning Spanish (`es_en`), and English speakers learning French (`fr_en`). This paper reports the approach used by our team which ended in third place for the `en_es` data set, second place for `es_en`, 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.,). In fact, much of the classic work on the 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-

[1]http://duolingo.com

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). A number of studies have also shown that age-of-acquisition is another strong predictor of word processing and lexical retrieval difficulty that is somewhat independent of word frequency (Brysbaert and Cortese, 2011; Ferrand et al., 2011). We therefore included mean age-of-acquisition (in English) as a feature, derived from published age-of-acquisition norms for 30,000 English words from (Kuperman et al., 2012) which covered many of the words present in the dataset. 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).

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 that related to the "learning type" of a user. In particular, we encoded features to estimate the long-term motivation and diligence of a user. These features could help predict how users interact with old and novel words they encounter.

To estimate users' motivation we grouped their exercises into "bursts." Bursts were separated by at least 24 hours. We used three concrete features about these bursts, namely the mean and median number of exercises within bursts as well as the total number of bursts of a given user. Users that were more motivated had potentially bursts with more exercises. Simultaneously, a larger total number of bursts indicated that the burst length estimate was more trustworthy.

To estimate users' diligenceconsistency? we speculated that a very diligent user might be using the app regularly at the same time of day, perhaps following a study schedule, compared to a less diligent user whose schedule might vary more. The data set did not provide a variable with the time of day, which would have been an interesting feature on its own. Instead, we were able to extract for each exercise the time of day relative to the first time a user had used the app, ranging from 0 to 1 (with 0 indicating the same time, 0.25 indicating a relative shift by 6 hours, etc.). We then discretized this variable into 20-minute bins and computed the entropy of the empirical frequency distribution over these bins. A lower entropy score indicated less variability in the times of day a user started their exercises.

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

[2] https://spacy.io/

[3] https://github.com/ljm625/syntaxnet-rest-api

[4] https://cloud.google.com/translate/

the exercise. These included the token of the previous word, the next word, and the instance word's root in the dependency 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 dependency-tree 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 single 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.

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.

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). Thus, as alluded to above, high-cardinality fea-

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

tures 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 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

To better understand which features or groups of features were most important to our model's predictions, we conducted a set of experiments in which we lesioned (i.e., removed) a group of features and re-trained the model on the `train` set, evaluating performance on the `dev` set. For simplicity, we ran each of the lesioned models on all language data and report the average performance. We did not run individual-language models as we did for our primary model.

The results of the lesion experiments are shown in Figure 1. The models are as follows.

none: all features are included.

user: all user-level features, including the user ID and other calculated features like entropy and measures of exercise bursts, are removed.

userid & user other: only user ID or only the calculated features user features, respectively, are removed.

word: Token and token root IDs; previous, next, and dependency-tree root word IDs; and morphological, part of speech, and dependency tree features are removed.

word id & word other: only word IDs or only other word features, respectively, are removed.

neighbors: Both word IDs and other word features are removed, but only for the previous, next, and dependency-tree root words. Information about the present word is maintained.

external: External information about the word, including corpus frequency, Levenshtein distance from translation, and age of acquisition, are removed.

temporal: Temporal information, including number and timing of past encounters with the word and error tracking information, is removed.

Interestingly, we found that for both user-level and word-level features, the bulk of the model's predictive power could be achieved using ID's alone, represented as high-cardinality categorical features. Removing other word features, such as morphological features and part of speech, created only a small degradation of performance. In the case of users, removing features such as entropy and average exercise burst length led to a tiny increase of performance. In the case of both users and words, though, we find that in the absence of ID features the other features are helpful and leads to better performance than removing all features. We also found that removing all information about neighboring words and the dependency-parse root word degraded performance. This confirms that word context matters, and suggests that users commonly make errors in word order, subject–verb matching and other grammatical rules.

Our external word features—Levenshtein distance to translation, frequency, and age of acquisition—provided a slight boost to model performance, showing the benefit of considering from a psychological and linguistic perspective what makes a word hard to learn. Adding temporal features about past encounters and errors helped the models, but not as much as we expected. While not included in the final model, we had also tried augmenting the temporal feature set with more

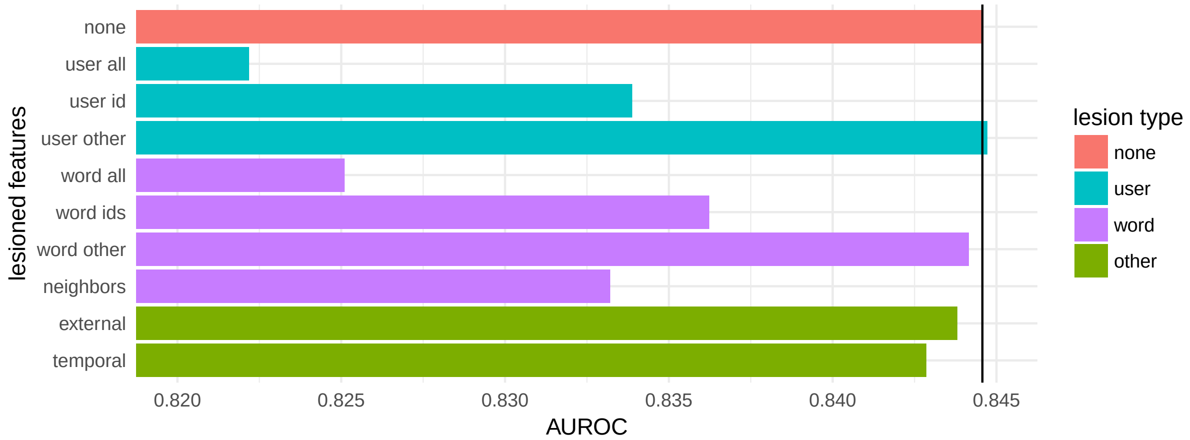

Figure 1: Performance on `dev` of models trained on all `train` data, with different groups of lesioned features. See main text for description of lesion groups

features related to massing and spacing of encounters with a word, but found it did not improve performance. This is perhaps not surprising given how small the benefit of the existing temporal features are in our model.

While not plotted above, we also ran a model lesioning exercise-level features including client, session type, format, and exercise duration. This model achieved an AUROC of .787, far lower than any other lesion. This points to the importance of how a question is asked for user performance, reflecting insights from psychology such as the difference between recall and recognition memory (Yonelinas, 2002).

4 Discussion

When approaching the Duolingo SLAM task, we hoped to leverage psychological insights in building our model. We found that in some cases, such as the age of acquisition, this was helpful. In general, though, our model gained its power not from hand-crafted features but from applying a powerful inference technique (gradient boosted trees) to raw input about user and word IDs and exercise features.

There are multiple reasons for the limited applicability of psychology to this competition. First, computational psychological models, while useful, are not well-suited to generating highly accurate predictions from large data sets. Because they are designed not for prediction but for explanation, they tend to use a small number of input variables and allow those variables to interact in limited ways. In contrast, gradient boosted trees, as well as other cutting-edge techniques like deep learning. can extract high-level interactions among hundreds of features. Though they are highly opaque, require a lot of data, and are not amenable to explanation, these models excel at prediction.

Second, it is possible that our ability to use theories of learning, including ideas about massed and spaced practice, was disrupted by the fact that the data may have been adaptively created using these very principles (Settles and Meeder, 2016). If Duolingo adaptively sequenced the spacing of trials based on past errors, then the relationship between future errors and past spacing may have differed from that found in the psychological literature (Cepeda et al., 2006).

Finally, a task in which more true generalization was required might have allowed psychologically inspired features to perform more competitively. In this task, there is a large amount of labeled training data for every user and for most words. This allow simple ID-based features to work. However, for a newly-encountered user or word, an ID is useless. Theory-driven features, in contrast, can generalize to new settings. Thus, if we were asked to generalize to a completely new language such as German, many parts of our model would falter but word frequency, age of acquisition, and Levenshtein distance to first-language translation would still likely be predictive.

5 Acknowledgments

This research was supported by NSF grant DRL-1631436 and seed funds from the NYU Dean for Science.

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.

Marc Brysbaert and Michael J Cortese. 2011. Do the effects of subjective frequency and age of acquisition survive better word frequency norms? *Quarterly Journal of Experimental Psychology*, 64(3):545–559.

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.

Ludovic Ferrand, Marc Brysbaert, Emmanuel Keuleers, Boris New, Patrick Bonin, Alain Méot, Maria Augustinova, and Christophe Pallier. 2011. Comparing word processing times in naming, lexical decision, and progressive demasking: Evidence from chronolex. *Frontiers in psychology*, 2:306.

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.

Victor Kuperman, Hans Stadthagen-Gonzalez, and Marc Brysbaert. 2012. Age-of-acquisition ratings for 30,000 english words. *Behavior Research Methods*, 44(4):978–990.

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.

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.

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.*

Andrew P Yonelinas. 2002. The nature of recollection and familiarity: A review of 30 years of research. *Journal of memory and language*, 46(3):441–517.