Second Language Acquisition Modeling

Burr Settles* **Chris Brust*** **Erin Gustafson*** **Masato Hagiwara*** **Nitin Madnani†**
*Duolingo, Pittsburgh, PA, USA †ETS, Princeton, NJ, USA
{burr,chrisb,erin,masato}@duolingo.com nmadnani@ets.org

Abstract

We present the task of *second language acquisition (SLA) modeling*. Given a history of errors made by learners of a second language, the task is to predict errors that they are likely to make at arbitrary points in the future. We describe a large corpus of more than 7M words produced by more than 6k learners of English, Spanish, and French using Duolingo, a popular online language-learning app. Then we report on the results of a shared task challenge aimed studying the SLA task via this corpus, which attracted 15 teams and synthesized work from various fields including cognitive science, linguistics, and machine learning.

1 Introduction

As computer-based educational apps increase in popularity, they generate vast amounts of student learning data which can be harnessed to drive personalized instruction. While there have been some recent advances for educational software in domains like mathematics, learning a language is more nuanced, involving the interaction of lexical knowledge, morpho-syntactic processing, and several other skills. Furthermore, most work that has applied natural language processing to language learner data has focused on intermediate-to-advanced students of English, particularly in assessment settings. Much less work has been devoted to beginners, learners of languages other than English, or ongoing study over time.

We propose *second language acquisition (SLA) modeling* as a new computational task to help broaden our understanding in this area. First, we describe a new corpus of language learner data, containing more than 7.1M words, annotated for production errors that were made by more than 6.4k learners of English, Spanish, and French, during their first 30 days of learning with Duolingo (a popular online language-learning app).

Then we report on the results of a "shared task" challenge organized by the authors using this SLA modeling corpus, which brought together 15 research teams. Our goal for this work is threefold: (1) to synthesize years of research in cognitive science, linguistics, and machine learning, (2) to facilitate cross-dialog among these disciplines through a common large-scale empirical task, and in so doing (3) to shed light on the most effective approaches to SLA modeling.

2 Shared Task Description

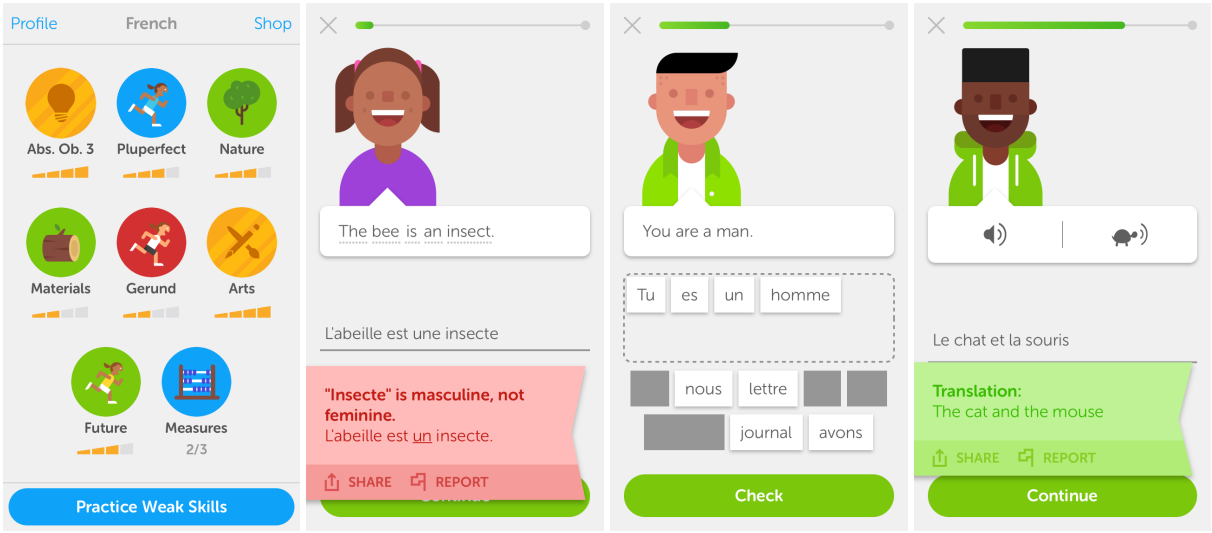
Our learner trace data comes from Duolingo: a free, award-winning, online language-learning platform. Since launching in 2012, more than 200 million learners worldwide have enrolled in Duolingo's game-like courses, either via the website[1] or mobile apps.

Figure 1(a) is a screen-shot of the home screen, which specifies the game-like curriculum. Each icon represents a skill, aimed at teaching thematically or grammatically grouped words or concepts. Learners can tap an icon to access lessons of new material, or to review material once all lessons are completed. Learners can also choose to get a personalized practice session that reviews previously-learned material from anywhere in the course by tapping the "practice weak skills" button.

2.1 Corpus Collection

To create the SLA modeling corpus, we sampled from Duolingo users who registered for a course and reached at least the tenth row of skill icons within the month of November 2015. By limiting the data to new users who reach this level of the course, we hope to better capture beginners' broader language-learning process, including repeated interaction with vocabulary and grammar

[1] https://www.duolingo.com



(a) home screen (b) reverse_translate (c) reverse_tap (d) listen

Figure 1: Duolingo screen-shots for an English-speaking student learning French (iPhone app, 2017). (a) The home screen, where learners can choose to do a "skill" lesson to learn new material, or get a personalized practice session by tapping the "practice weak skills" button. (b–d) Examples of the three exercise types included in our shared task experiments, which require the student to construct responses in the language they are learning.

over time. Note that we excluded all learners who took a placement test to skip ahead in the course, since these learners are likely more advanced.

2.2 Three Language Tracks

An important question for SLA modeling is: to what extent does an approach generalize across languages? While the majority of Duolingo users learn English—which can significantly improve job prospects and quality of life (Pinon and Haydon, 2010)—Spanish and French are the second and third most popular courses. To encourage researchers to explore language-agnostic features, or unified cross-lingual modeling approaches, we created three tracks: English learners (who speak Spanish), Spanish learners (who speak English), and French learners (who speak English).

2.3 Label Prediction Task

The goal of the task is as follows: given a history of token-level errors made by the learner in the learning language (L2), accurately predict the errors they will make in the future. In particular, we focus on three Duolingo exercise formats that require the learners to engage in *active recall*, that is, they must construct answers in the L2 through translation or transcription.

Figure 1(b) illustrates a *reverse translate* item, where learners are given a prompt in the language they know (e.g., their L1 or native language), and translate it into the L2. Figure 1(c) illustrates a *reverse tap* item, which is a simpler version of the same format: learners construct an answer using a bank of words and distractors. Figure 1(d) is a *listen* item, where learners hear an utterance in the L2 they are learning, and must transcribe it. Duolingo does include many other exercise formats, but we focus on these three in the current work, since constructing L2 responses through translation or transcription is associated with deeper levels of processing, which in turn is more strongly associated with learning (Craik and Tulving, 1975).

learner:	wen	can		help	
reference:	when	can	I	help	?
label:	✗	✓	✗	✓	

Figure 2: An illustration of how data labels are generated. Learner responses are aligned with the most similar reference answer, and tokens from the reference that do not match are labeled errors.

Since each exercise can have multiple correct answers (due to synonyms, homophones, or ambiguities in tense, number, formality, etc.), Duolingo uses a finite-state machine to align the learner's response to the most similar reference answer form a large set of acceptable responses, based on token string edit distance (Levenshtein, 1966). For example, Figure 1(b) shows an example of corrective feedback based on such an alignment.

Figure 2 shows how we use these alignments to generate labels for the SLA modeling task. In this case, an English (from Spanish) learner was asked to translate, "¿Cuándo puedo ayudar?" and wrote "wen can help" instead of "When can I help?" This produces two errors (a typo and a missing pronoun). We ignore capitalization, punctuation, and accents when matching tokens.

2.4 Data Set Format

Sample data from the resulting corpus can be found in Figure 3. Each token from the reference answer is labeled according to the alignment with the learner's response (the final column: `0` for correct and `1` for incorrect). Tokens are grouped together by exercise, including user-, exercise-, and session-level meta-data in the previous line (marked by the `#` character). We included all exercises done by the users sampled from the 30-day data collection window.

The overall format is inspired by the Universal Dependencies (UD) format[2]. Column 1 is a unique B64-encoded token ID, column 2 is a token (word), and columns 3–6 are morpho-syntactic features from the UD tag set (part of speech, morphology features, and dependency parse labels and edges). These were generated by processing the aligned reference answers with Google SyntaxNet (Andor et al., 2016). Because UD tags are meant to be language-agnostic, it was our goal to help make cross-lingual SLA modeling more straightforward by providing these features.

Exercise meta-data includes the following:

- **user**: 8-character unique anonymous user ID for each learner (B64-encoded)
- **countries**: 2-character ISO country codes from which this learner has done exercises
- **days**: number of days since the learner started learning this language on Duolingo
- **client**: session device platform
- **session**: session type (e.g., lesson or practice)
- **format**: exercise format (see Figure 1)
- **time**: the time (in seconds) it took the learner to submit a response for this exercise.

Lesson sessions (about 77% of the data set) are where new words or concepts are introduced, although lessons also include previously-learned material (e.g., each exercise attempts to introduce only one new word or inflection, so all other tokens should have been seen by the student before). Practice sessions (22%) should contain only previously-seen words and concepts. Test sessions (1%) are mini-quizzes that allow a student to skip out of a single skill in the curriculum (i.e., the student may have never seen this content before in the Duolingo app, but may well have had prior knowledge before starting the course).

It is worth mentioning that for the shared task, we did not provide actual learner responses, only the closest reference answers. Releasing such data (at least in the TEST set) would by definition give away the labels and might undermine the task. However, we plan to release a future version of the corpus that is enhanced with additional meta-data, including the actual learner responses.

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

Table 1: Summary of the SLA modeling data set.

2.5 Challenge Timeline

The data were released in two phases. In phase 1 (8 weeks), TRAIN and DEV partitions were released with labels, along with a baseline system and evaluation script, for system development. In phase 2 (10 days), the TEST partition was released without labels, and teams submitted predictions to CodaLab[3] for blind evaluation. To allow teams to compare different system parameters or features, they were allowed to submit up to 10 predictions total (up to 2 per day) during this phase.

Table 1 reports summary statistics for each of the data partitions for all three tracks. We created TRAIN, DEV, and TEST partitions as follows. For each user, the first 80% of their exercises were placed in the TRAIN set, the subsequent 10% in DEV, and the final 10% in TEST. Hence the three data partitions are sequential, and contain ordered observations for all users.

Note that because the three data partitions are sequential, and the DEV set contains observations that are potentially valuable for making TEST set predictions, most teams opted to combine the TRAIN and DEV sets to train their systems in final phase 2 evaluations.

[2] http://universaldependencies.org

[3] http://codalab.org

```
# user:XEinXf5+  countries:CO  days:2.678  client:web  session:practice  format:reverse_translate  time:6
oMGsnnH/0101   When      ADV    PronType=Int|fPOS=ADV++WRB                                          advmod     4  1
oMGsnnH/0102   can       AUX    VerbForm=Fin|fPOS=AUX++MD                                           aux        4  0
oMGsnnH/0103   I         PRON   Case=Nom|Number=Sing|Person=1|PronType=Prs|fPOS=PRON++PRP           nsubj      4  1
oMGsnnH/0104   help      VERB   VerbForm=Inf|fPOS=VERB++VB                                          ROOT       0  0

# user:XEinXf5+  countries:CO  days:5.707  client:android  session:practice  format:reverse_translate  time:22
W+QU2fm70301   He        PRON   Case=Nom|Gender=Masc|Number=Sing|Person=3|PronType=Prs|fPOS=PRON++PRP  nsubj      3  0
W+QU2fm70302   's        AUX    Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin|fPOS=AUX++VBZ   aux        3  1
W+QU2fm70303   wearing   VERB   Tense=Pres|VerbForm=Part|fPOS=VERB++VBG                             ROOT       0  0
W+QU2fm70304   two       NUM    NumType=Card|fPOS=NUM++CD                                           nummod     5  0
W+QU2fm70305   shirts    NOUN   Number=Plur|fPOS=NOUN++NNS                                          dobj       3  0

# user:XEinXf5+  countries:CO  days:10.302  client:web  session:lesson  format:reverse_translate  time:28
vOeGrMgP0101   We        PRON   Case=Nom|Number=Plur|Person=1|PronType=Prs|fPOS=PRON++PRP           nsubj      2  0
vOeGrMgP0102   eat       VERB   Mood=Ind|Tense=Pres|VerbForm=Fin|fPOS=VERB++VBP                     ROOT       0  1
vOeGrMgP0103   cheese    NOUN   Degree=Pos|fPOS=ADJ++JJ                                             dobj       2  1
vOeGrMgP0104   and       CONJ   fPOS=CONJ++CC                                                       cc         2  0
vOeGrMgP0105   they      PRON   Case=Nom|Number=Plur|Person=3|PronType=Prs|fPOS=PRON++PRP           nsubj      6  0
vOeGrMgP0106   eat       VERB   Mood=Ind|Tense=Pres|VerbForm=Fin|fPOS=VERB++VBP                     conj       2  1
vOeGrMgP0107   fish      NOUN   fPOS=X++FW                                                          dobj       6  0
```

Figure 3: Sample exercise data from an English learner over time: roughly two, five, and ten days into the course.

2.6 Evaluation

We use area under the ROC curve (AUC) as the primary evaluation metric for SLA modeling (Fawcett, 2006). AUC is a common measure of ranking quality in classification tasks, and can be interpreted as the probability that the system will rank a randomly-chosen error above a randomly-chosen non-error. We argue that this notion of ranking quality is particularly useful for evaluating systems that might be used for 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 have forgotten or make errors at a given point in time.

We also report F1 score—the harmonic mean of precision and recall—as a secondary metric, since it is more common in similar skewed-class labeling tasks (e.g., Ng et al., 2013). Note, however, that F1 can be significantly improved simply by tuning the classification threshold (fixed at 0.5 for our evaluations) without affecting AUC.

3 Results

A total of 15 teams participated in the task, of which 13 responded to a brief survey about their approach, and 11 submitted system description papers. All but two of these teams submitted predictions for all three language tracks.

Official shared task results are reported in Table 2. System ranks are determined by sorting teams according to AUC, and using DeLong's test (DeLong et al., 1988) to identify statistical ties. For the remainder of this section, we provide a summary of each team's approach, ordered by the team's average rank across all three tracks. Certain teams are marked with modeling choice indicators ($\diamondsuit, \clubsuit, \ddagger$), which we discuss further in §5.

SanaLabs (Nilsson et al., 2018) used a combination of recurrent neural network (RNN) predictions with those of a Gradient Boosted Decision Tree (GBDT) ensemble, trained independently for each track. This was motivated by the observation that RNNs work well for sequence data, while GBDTs are often the best-performing non-neural model for shared tasks using tabular data. They also engineered several token context features, and learner/token history features such as number of times seen, time since last practice, etc.

singsound (Xu et al., 2018) used an RNN architecture using four types of encoders, representing different types of features: token context, linguistic information, user data, and exercise format. The RNN decoder integrated information from all four encoders. Ablation experiments revealed the context encoder (representing the token) contributed the most to model performance, while the linguistic encoder (representing grammatical information) contributed the least.

NYU (Rich et al., 2018) used an ensemble of GBDTs with features engineered based on psychological theories of cognition. Predictions for each track were averaged between a track-specific model and a unified model (trained on data from all three tracks). In addition to the word, user, and exercise features provided, the authors included word lemmas, corpus frequency, L1-L2 cognates, and features indicating user motivation and diligence (derived from usage patterns), and others. Ablation studies indicated that most of the performance was due to the user and token features.

English Track			
↑	Team	AUC	F1
1	SanaLabs ♢♣	.861	.561
1	singsound ♢	.861	.559
3	NYU ♣‡	.859	.468
4	TMU ♢‡	.848	.476
5	CECL ‡	.846	.414
6	Cambridge ♢	.841	.479
7	UCSD ♣	.829	.424
8	nihalnayak	.821	.376
8	LambdaLab ♣	.821	.389
10	Grotoco	.817	.462
11	jilljenn	.815	.329
12	ymatusevych	.813	.381
13	renhk	.797	.448
14	zlb241	.787	.003
15	SLAM_baseline	.774	.190

Spanish Track			
↑	Team	AUC	F1
1	SanaLabs ♢♣	.838	.530
2	NYU ♣‡	.835	.420
2	singsound ♢	.835	.524
4	TMU ♢‡	.824	.439
5	CECL ‡	.818	.390
6	Cambridge ♢	.807	.435
7	UCSD ♣	.803	.375
7	LambdaLab ♣	.801	.344
9	Grotoco	.791	.452
9	nihalnayak	.790	.338
11	ymatusevych	.789	.347
11	jilljenn	.788	.306
13	renhk	.773	.432
14	SLAM_baseline	.746	.175
15	zlb241	.682	.389

French Track			
↑	Team	AUC	F1
1	SanaLabs ♢♣	.857	.573
2	singsound ♢	.854	.569
2	NYU ♣‡	.854	.493
4	CECL ‡	.843	.487
5	TMU ♢‡	.839	.502
6	Cambridge ♢	.835	.508
7	UCSD ♣	.823	.442
8	LambdaLab ♣	.815	.415
8	Grotoco	.813	.502
10	nihalnayak	.811	.431
10	jilljenn	.809	.406
10	ymatusevych	.808	.441
13	simplelinear	.807	.394
14	renhk	.796	.481
15	SLAM_baseline	.771	.281

Table 2: Final results. Ranks (↑) are determined by statistical ties (see text). Markers indicate which systems include recurrent neural architectures (♢), decision tree ensembles (♣), or a multitask model across all tracks (‡).

TMU (Kaneko et al., 2018) used a combination of two bidirectional RNNs—the first to predict potential user errors at a given token, and a second to track the history of previous answers by each user. These networks were jointly trained through a unified objective function. The authors did not engineer any additional features, but did train a single model for all three tracks (using a track ID feature to distinguish among them).

CECL (Bestgen, 2018) used a logistic regression approach. The base feature set was expanded to include many feature conjunctions, including word n-grams crossed with the token, user, format, and session features provided with the data set.

Cambridge (Yuan, 2018) trained two RNNs—a sequence labeler, and a sequence-to-sequence model taking into account previous answers—and found that averaging their predictions yielded the best results. They focused on the English track, experimenting with additional features derived from other English learner corpora. Hyper-parameters were tuned for English and used as-is for other tracks, with comparable results.

UCSD (Tomoschuk and Lovelett, 2018) used a random forest classifier with a set of engineered features motivated by previous research in memory and linguistic effects in SLA, including "word neighborhoods," corpus frequency, cognates, and repetition/experience with a given word. The system also included features specific to each user, such as mean and variance of error rates.

LambdaLab (Chen et al., 2018) used GBDT models independently for each track, deriving their features from confirmatory analysis of psychologically-motivated hypotheses on the TRAIN set. These include proxies for student engagement, spacing effect, response time, etc.

nihalnayak (Nayak and Rao, 2018) used a logistic regression model similar to the baseline, but added features inspired by research in code-mixed language-learning where context plays an important role. In particular, they included word, part of speech, and metaphone features for previous:current and current:next token pairs.

Grotoco (Klerke et al., 2018) also used logistic regression, including word lemmas, frequency, cognates, and user-specific features such as word error rate. Interestingly, the authors found that ignoring each user's first day of exercise data improved their predictions, suggesting that learners first needed to familiarize themselves with app before their data were reliable for modeling.

jilljenn (Vie, 2018) used a deep factorization machine (DeepFM), a neural architecture developed for click-through rate prediction in recommender systems. This model allows learning from both lower-order and higher-order induced features and their interactions. The DeepFM outperformed a simple logistic regression baseline without much additional feature engineering.

Other teams did not submit system description papers. However, according to a task organizer survey **ymatusevych** used a linear model with multilingual word embeddings, corpus frequency, and several L1-L2 features such as cognates. Additionally, **simplelinear** used an ensemble of some sort (for the French track only). **renhk** and **zlb241** provided no details about their systems.

SLAM_baseline is the baseline system provided by the task organizers. It is a simple logistic regression using data set features, trained separately for each track using stochastic gradient descent on the TRAIN set only.

4 Related Work

SLA modeling is a rich problem, and presents a opportunity to synthesize work from various subfields in cognitive science, linguistics, and machine learning. This section highlights a few key concepts from these fields, and how they relate to the approaches taken by shared task participants.

Item response theory (IRT) is a common psychometric modeling approach used in educational software (e.g., Chen et al., 2005). In its simplest form (Rasch, 1980), an IRT model is a logistic regression with two weights: one representing the learner's ability (i.e., user ID), and the other representing the difficulty of the exercise or test item (i.e., token ID). An extension of this idea is the *additive factor model* (Cen et al., 2008) which adds additional "knowledge components" (e.g., lexical, morphological, or syntactic features). Teams that employed linear models (including our baseline) are essentially all additive factor IRT models.

For decades, tutoring systems have also employed sequence models like HMMs to perform *knowledge tracing* (Corbett and Anderson, 1995), a way of estimating a learner's mastery of knowledge over time. RNN-based approaches that encode user performance over time (i.e., that span across exercises) are therefore variants of *deep knowledge tracing* (Piech et al., 2015).

Relatedly, the *spacing effect* (Dempster, 1989) is the observation that people will not only learn but also forget over time, and they remember more effectively through scheduled practices that are spaced out. Settles and Meeder (2016) and Ridgeway et al. (2017) recently proposed non-linear regressions that explicitly encode the rate of forgetting as part of a decision surface, however none of the current teams chose to do this. Instead, forgetting was either modeled through engineered features (e.g., user/token histories), or opaquely handled by sequential RNN architectures.

SLA modeling also bears some similarity to research in *grammatical error detection* (Leacock et al., 2010) and *correction* (Ng et al., 2013). For these tasks, a model is given a (possibly ill-formed) sequence of words produced by a learner, and the task is to identify which are mistakes. SLA modeling is in some sense the opposite: given a well-formed sequence of words that a learner should be able to produce, identify where they are likely to make mistakes. Given these similarities, a few teams adapted state-of-the-art GEC/GED approaches to create their SLA modeling systems.

Finally, *multitask learning* (e.g., Caruana, 1997) is the idea that machine learning systems can do better at multiple related tasks by trying to solve them simultaneously. For example, recent work in machine translation has demonstrated gains through learning to translate multiple languages with a unified model (Dong et al., 2015). Similarly, the three language tracks in this work presented an opportunity to explore a unified multitask framework, which a few teams did with positive results.

5 Meta-Analyses

In this section, we analyze the various modeling choices explored by the different teams in order to shed light on what kinds of algorithmic and feature engineering decisions appear to be useful for the SLA modeling task.

5.1 Learning Algorithms

Here we attempt to answer the question of whether particular machine learning algorithms have a significant impact on task performance. For example, the results in Table 2 suggest that the algorithmic choices indicated by (♢, ♣, ‡) are particularly effective. Is this actually the case?

To answer this question, we partitioned the TEST set into 6.4k subsets (one for each learner), and computed per-user AUC scores for each team's predictions (83.9k observations total). We also coded each team with indicator variables to describe their algorithmic approach, and used a regression analysis to determine if these algorithmic variations had any significant effects on learner-specific AUC scores.

To analyze this properly, however, we need to determine whether the differences among modeling choices are actually meaningful, or can simply be explained by sampling error due to random variations among users, teams, or tracks. To do this, we use a *linear mixed-effects model* (cf., Baayen, 2008, Ch. 7). In addition to modeling the *fixed* effects of the various learning algorithms, we can also model the *random* effects represented by the

Fixed effects (algorithm choices)	Effect	p-value	
Intercept	.786	$<.001$	***
Recurrent neural network (♢)	$+.028$	.012	*
Decision tree ensemble (♣)	$+.018$	.055	.
Linear model (e.g., IRT)	$-.006$	.541	
Multitask model (‡)	$+.023$	.017	*
Random effects	**St. Dev.**		
User ID	$\pm.086$		
Team ID	$\pm.013$		
Track ID	$\pm.011$		

Table 3: Mixed-effects analysis of learning algorithms.

Features used	Popularity	Effect	
Word (surface form)	▮▮▮▮▮▮▮▮▮▮	$+.005$	
User ID	▮▮▮▮▮▮▮▮▮▮	$+.014$	
Part of speech	▮▮▮▮▮▮▮▮▮▮	$-.008$	
Dependency labels	▮▮▮▮▮▮▮▮	$-.011$	
Morphology features	▮▮▮▮▮▮▮▮	$-.021$	
Response time	▮▮▮▮▮▮▮▮	$+.028$	*
Days in course	▮▮▮▮▮▮▮	$+.023$	.
Client	▮▮▮▮▮▮▮	$+.005$	
Countries	▮▮▮▮▮▮	$+.012$	
Dependency edges	▮▮▮▮	$-.000$	
Session	▮▮▮	$+.014$	
Word corpus frequency	▮▮▮▮▮	$+.008$	
Spaced repetition features	▮▮▮▮▮	$+.013$	
L1-L2 cognates	▮▮▮▮	$+.001$	
Word embeddings	▮▮▮	$+.020$	
Word stem/root/lemma	▮▮▮	$+.007$	

Table 4: Summary of system features—both provided (top) and team-engineered (bottom)—with team popularity and univariate mixed-effects estimates.

user ID (learners may vary by ability), the team ID (teams may differ in other aspects not captured by our schema, e.g., the hardware used), and the track ID (tracks may vary inherently in difficulty).

Table 3 presents a mixed-effects analysis for the algorithm variations used by at least 3 teams. The intercept can be interpreted as the "average" AUC of .786. Controlling for the random effects of user (which exhibits a wide standard deviation of $\pm.086$ AUC), team ($\pm.013$), and track ($\pm.011$), three of the algorithmic choices are at least marginally significant ($p < .1$). For example, we might expect a system that uses RNNs to model learner mastery over time would add $+.028$ to learner-specific AUC (all else being equal). Note that most teams' systems that were not based on RNNs or tree ensembles used logistic regression, hence the "linear model" effect is negligible (effectively treated as a control condition in the analysis).

These results suggest two key insights for SLA modeling. First, *non-linear algorithms* are particularly desirable[4], and second, *multitask learning* approaches that share information across tracks (i.e., languages) are also effective.

5.2 Feature Sets

We would also like to get a sense of which features, if any, significantly affect system performance. Table 4 lists features provided with the SLA modeling data set, as well as several newly-engineered feature types that were employed by at least three teams (note that the precise details may vary from team to team, but in our view aim to capture the same phenomena). We also include each feature's popularity and an effect estimate[5].

Broadly speaking, results suggest that feature engineering had a much smaller impact on system performance than the choice of learning algorithm. Only "response time" and "days in course" showed even marginally significant trends.

Of particular interest is the observation that *morpho-syntactic* features (described in §2.4) actually seem to have weakly negative effects. This echoes **singsound**'s finding that their linguistic encoder contributed the least to system performance, and **Cambridge** determined through ablation studies that these features in fact hurt their system. One reasonable explanation is that these automatically-generated features contain too many systematic parsing errors to provide value. (Note that **NYU** artificially introduced punctuation to the exercises and re-parsed the data in their work.)

As for newly-engineered features, word information such as frequency, semantic embeddings, and stemming were popular. It may be that these features showed such little return because our corpus was too biased toward beginners—thus representing a very narrow sample of language—for these features to be meaningful. Cognate features were an interesting idea used by a few teams, and may have been more useful if the data included

[4] Interestingly, the only linear model to rank among the top 5 (**CECL**) relied on combinatorial feature conjunctions—which effectively alter the decision surface to be non-linear with respect to the original features. The RNN hidden nodes and GBDT constituent trees from other top systems may in fact be learning to represent these same feature conjunctions.

[5] This is similar to the analysis in §5.1, except that we regress on each feature separately. That is, a feature is the only fixed effect in the model (alongside intercept), while still controlling for user, team, and track random effects.

users from a wider variety of different L1 language backgrounds. Spaced repetition features also exhibited marginal (but statistically insignificant) gains. We posit that the 30-day window we used for data collection was simply not long enough for these features to capture more long-term learning (and forgetting) trends.

5.3 Ensemble Analysis

Another interesting research question is: what is the upper-bound for this task? This can be estimated by treating each team's best submission as an independent system, and combining the results using ensemble methods in a variety of ways. Such analyses have been previously applied to other shared task challenges and meta-analyses (e.g., Malmasi et al., 2017).

The **oracle** system is meant to be an upper-bound: for each token in the TEST set, the oracle outputs the team prediction with the lowest error for that particular token. We also experiment with **stacking** (Wolpert, 1992) by training a logistic regression classifier using each team's prediction as an input feature[6]. Finally, we also pool system predictions together by taking their **average** (mean).

Table 5 reports AUC for various ensemble methods as well as some of the top performing team systems for all three tracks. Interestingly, the oracle is exceptionally accurate ($>$.993 AUC and $>$.884 F1, not shown). This indicates that the *potential* upper limit of performance on this task is quite high, since there exists a near-perfect ranking of tokens in the TEST set based only on predictions from these 15 diverse participating teams.

The stacking classifier produces significantly better rankings than any of the constituent systems alone, while the average (over all teams) ranked between the 3rd and 4th best system in all three tracks. Inspection of stacking model weights revealed that it largely learned to trust the top-performing systems, so we also tried simply averaging the top 3 systems for each track, and this method was statistically tied with stacking for the English and French tracks ($p = 0.002$ for Spanish). Interestingly, the highest-weighted team in each track's stacking model was **singsound** (+2.417 on average across the three models), followed by **NYU** (+1.632), whereas the top-performing team **SanaLabs** had a surprisingly lower weight (+0.841). This could be due to the fact that their system was itself an ensemble of an RNN and GBDT models, which were used (in isolation) by each of the other two teams. This seems to add further support for the effectiveness of combining these algorithms for the task.

[6]Note that we only have TEST set predictions for each team. While we averaged stacking classifier weights across 10 folds using cross-validation, the reported AUC is still likely an over-estimate, since the models were in some sense trained on the TEST set.

System	English	Spanish	French
Oracle	*.995*	*.996*	*.993*
Stacking	.867	.844	.863
Average (top 3)	.867	.843	.863
1st team	.861	.838	.857
2nd team	.861	.835	.854
3rd team	.859	.835	.854
Average (all)	.857	.832	.852
4th team	.848	.824	.843

Table 5: AUC results for the ensemble analysis.

6 Conclusion and Future Work

In this work, we presented the task of *second language acquisition (SLA) modeling*, described a large data set for studying this task, and reported on the results of a shared task challenge that explored this new domain. The task attracted strong participation from 15 teams, who represented a wide variety of fields including cognitive science, linguistics, and machine learning.

Among our key findings is the observation that, for this particular formulation of the task, the choice of learning algorithm appears to be more important than clever feature engineering. In particular, the most effective teams employed sequence models (e.g., RNNs) that can capture user performance over time, and tree ensembles (e.g., GBDTs) that can capture non-linear relationships among features. Furthermore, using a multitask framework—in this case, a unified model that leverages data from all three language tracks—can provide further improvements.

Still, many teams opted for a simpler algorithm (e.g., logistic regression) and concentrated instead on more psychologically-motivated features. While these teams did not always perform as well, several demonstrated through ablation studies that these features can be useful within the limitations of the algorithm. It is possible that the constraints of the SLA modeling data set (beginner language, homogeneous L1 language background, short 30-day time frame, etc.) prevented these features from being more useful across different

teams and learning algorithms. It would be interesting to revisit these ideas using a more diverse and longitudinal data set in the future.

To support ongoing research in SLA modeling, current and future releases of our data set will be publicly maintained online at: https://doi.org/10.7910/DVN/8SWHNO.

Acknowledgments

The authors would like to acknowledge Bożena Pająk, Joseph Rollinson, and Hideki Shima for their help planning and co-organizing the shared task. Eleanor Avrunin and Natalie Glance made significant contributions to early versions of the SLA modeling data set, and Anastassia Loukina and Kristen K. Reyher provided helpful advice regarding mixed-effects modeling. Finally, we would like to thank the organizers of the NAACL-HLT 2018 Workshop on Innovative Use of NLP for Building Educational Applications (BEA) for providing a forum for this work.

References

D. Andor, C. Alberti, D. Weiss, A. Severyn, A. Presta, K. Ganchev, S. Petrov, and M. Collins. 2016. Globally normalized transition-based neural networks. *CoRR*, abs/1603.06042.

R.H. Baayen. 2008. *Analyzing Linguistic Data: A Practical Introduction to Statistics using R*. Cambridge University Press.

Y. Bestgen. 2018. Predicting second language learner successes and mistakes by means of conjunctive features. In *Proceedings of the NAACL-HLT Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*. ACL.

R. Caruana. 1997. Multitask learning. *Machine Learning*, 28:41–75.

H. Cen, K. Koedinger, and B. Junker. 2008. Comparing two IRT models for conjunctive skills. In *Proceedings of the Conference on Intelligent Tutoring Systems (ITS)*, pages 796–798. Springer.

C.M. Chen, H.M. Lee, and Y.H. Chen. 2005. Personalized e-learning system using item response theory. *Computers & Education*, 44(3):237–255.

G. Chen, C. Hauff, and G.J. Houben. 2018. Feature engineering for second language acquisition modeling. In *Proceedings of the NAACL-HLT Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*. ACL.

A.T. Corbett and J.R. Anderson. 1995. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278.

F.I.M. Craik and E. Tulving. 1975. Depth of processing and the retention of words in episodic memory. *Journal of Experimental Psychology*, 104:268–294.

E.R. DeLong, D.M. DeLong, and D.L. Clarke-Pearson. 1988. Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, 44:837–845.

F.N. Dempster. 1989. Spacing effects and their implications for theory and practice. *Educational Psychology Review*, 1(4):309–330.

D. Dong, H. Wu, W. He, D. Yu, and H. Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 1723–1732. ACL.

T. Fawcett. 2006. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874.

M. Kaneko, T. Kajiwara, and M. Komachi. 2018. TMU system for SLAM-2018. In *Proceedings of the NAACL-HLT Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*. ACL.

S. Klerke, H.M. Alonso, and B. Plank. 2018. Grotoco@SLAM: Second language acquisition modeling with simple features, learners and task-wise models. In *Proceedings of the NAACL-HLT Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*. ACL.

C. Leacock, M. Chodorow, M. Gamon, and J. Tetreault. 2010. Automated grammatical error detection for language learners. *Synthesis Lectures on Human Language Technologies*, 3(1):1–134.

V.I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

S. Malmasi, K. Evanini, A. Cahill, J. Tetreault, R. Pugh, C. Hamill, D. Napolitano, and Y. Qian. 2017. A report on the 2017 Native Language Identification shared task. In *Proceedings of the EMNLP Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*, pages 62–75, Copenhagen, Denmark. ACL.

N.V. Nayak and A.R. Rao. 2018. Context based approach for second language acquisition. In *Proceedings of the NAACL-HLT Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*. ACL.

H.T. Ng, S.M. Wu, Y. Wu, C. Hadiwinoto, and J. Tetreault. 2013. The CoNLL-2013 shared task on grammatical error correction. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*, pages 1–12. ACL.

S. Nilsson, A. Osika, A. Sydorchuk, F. Sahin, and A. Huss. 2018. 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.

C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L.J. Guibas, and J. Sohl-Dickstein. 2015. Deep knowledge tracing. In *Advances in Neural Information Processing Systems (NIPS)*, pages 505–513.

R. Pinon and J. Haydon. 2010. The benefits of the English language for individuals and societies: Quantitative indicators from Cameroon, Nigeria, Rwanda, Bangladesh and Pakistan. Technical report, Euromonitor International for the British Council.

G. Rasch. 1980. *Probabilistic models for some intelligence and attainment tests*. The University of Chicago Press.

A. Rich, P.O. Popp, D. Halpern, A. Rothe, and T. Gureckis. 2018. 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.

K. Ridgeway, M.C. Mozer, and A.R. Bowles. 2017. Forgetting of foreign-language skills: A corpus-based analysis of online tutoring software. *Cognitive Science*, 41(4):924–949.

B. Settles and B. Meeder. 2016. A trainable spaced repetition model for language learning. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 1848–1858. ACL.

B. Tomoschuk and J. Lovelett. 2018. A memory-sensitive classification model of errors in early second language learning. In *Proceedings of the NAACL-HLT Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*. ACL.

J.J. Vie. 2018. Deep factorization machines for knowledge tracing. In *Proceedings of the NAACL-HLT Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*. ACL.

D.H. Wolpert. 1992. Stacked generalization. *Neural Networks*, 5(2):241–259.

S. Xu, J. Chen, and L. Qin. 2018. CLUF: 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.

Z. Yuan. 2018. Neural sequence modelling for learner error prediction. In *Proceedings of the NAACL-HLT Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*. ACL.