A Trainable Spaced Repetition Model for Language Learning

Burr Settles[*]
Duolingo
Pittsburgh, PA USA
burr@duolingo.com

Brendan Meeder[†]
Uber Advanced Technologies Center
Pittsburgh, PA USA
bmeeder@cs.cmu.edu

Abstract

We present *half-life regression (HLR)*, a novel model for spaced repetition practice with applications to second language acquisition. HLR combines psycholinguistic theory with modern machine learning techniques, indirectly estimating the "half-life" of a word or concept in a student's long-term memory. We use data from Duolingo — a popular online language learning application — to fit HLR models, reducing error by 45%+ compared to several baselines at predicting student recall rates. HLR model weights also shed light on which linguistic concepts are systematically challenging for second language learners. Finally, HLR was able to improve Duolingo daily student engagement by 12% in an operational user study.

1 Introduction

The *spacing effect* is the observation that people tend to remember things more effectively if they use *spaced repetition practice* (short study periods spread out over time) as opposed to *massed practice* (i.e., "cramming"). The phenomenon was first documented by Ebbinghaus (1885), using himself as a subject in several experiments to memorize verbal utterances. In one study, after a day of cramming he could accurately recite 12-syllable sequences (of gibberish, apparently). However, he could achieve comparable results with half as many practices spread out over three days.

The *lag effect* (Melton, 1970) is the related observation that people learn even better if the spacing between practices gradually increases. For example, a learning schedule might begin with review sessions a few seconds apart, then minutes, then hours, days, months, and so on, with each successive review stretching out over a longer and longer time interval.

The effects of spacing and lag are well-established in second language acquisition research (Atkinson, 1972; Bloom and Shuell, 1981; Cepeda et al., 2006; Pavlik Jr and Anderson, 2008), and benefits have also been shown for gymnastics, baseball pitching, video games, and many other skills. See Ruth (1928), Dempster (1989), and Donovan and Radosevich (1999) for thorough meta-analyses spanning several decades.

Most practical algorithms for spaced repetition are simple functions with a few hand-picked parameters. This is reasonable, since they were largely developed during the 1960s–80s, when people would have had to manage practice schedules without the aid of computers. However, the recent popularity of large-scale online learning software makes it possible to collect vast amounts of parallel student data, which can be used to empirically train richer statistical models.

In this work, we propose *half-life regression (HLR)* as a trainable spaced repetition algorithm, marrying psycholinguistically-inspired models of memory with modern machine learning techniques. We apply this model to real student learning data from Duolingo, a popular language learning app, and use it to improve its large-scale, operational, personalized learning system.

2 Duolingo

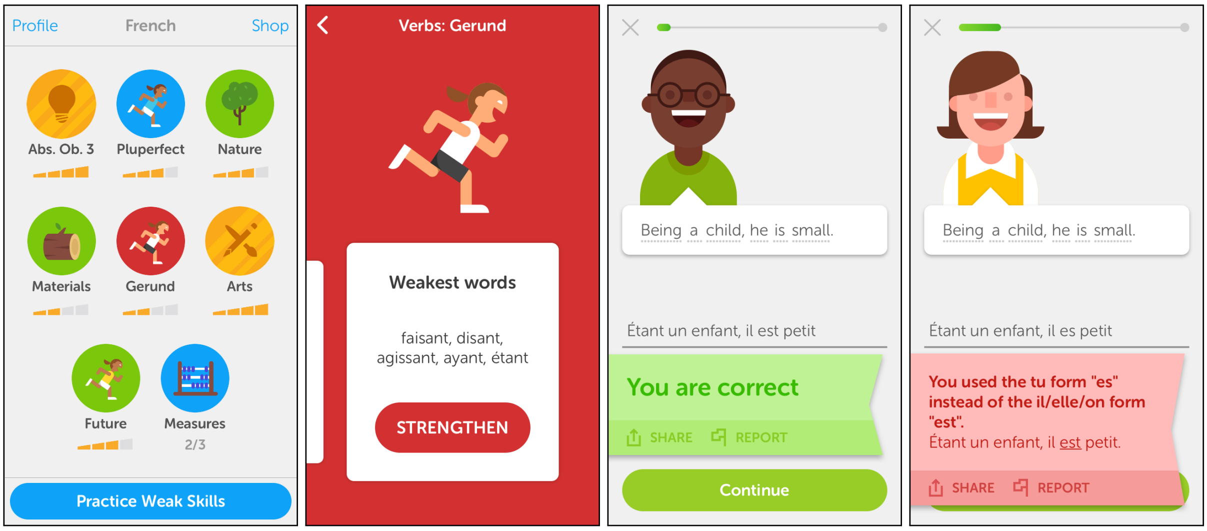
Duolingo is a free, award-winning, online language learning platform. Since launching in 2012, more than 150 million students from all over the world have enrolled in a Duolingo course, either via the website[1] or mobile apps for Android, iOS,

[*] Corresponding author.

[†] Research conducted at Duolingo.

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

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 1848–1858,
Berlin, Germany, August 7-12, 2016. ©2016 Association for Computational Linguistics



Weakest words

faisant, disant, agissant, ayant, étant

Being a child, he is small.

Étant un enfant, il est petit

You are correct

Being a child, he is small.

Étant un enfant, il es petit

You used the tu form "es" instead of the il/elle/on form "est".
Étant un enfant, il est petit.

(a) skill tree screen (b) skill screen (c) correct response (d) incorrect response

Figure 1: Duolingo screenshots for an English-speaking student learning French (iPhone app, 2016). (a) A course skill tree: golden skills have four bars and are "at full strength," while other skills have fewer bars and are due for practice. (b) A skill screen detail (for the *Gerund* skill), showing which words are predicted to need practice. (c,d) Grading and explanations for a translation exercise.

étant	un	enfant	il	est	petit
être.V.GER	*un*.DET.INDF.M.SG	*enfant*.N.SG	*il*.PN.M.P3.SG	*être*.V.PRES.P3.SG	*petit*.ADJ.M.SG

Figure 2: The French sentence from Figure 1(c,d) and its lexeme tags. Tags encode the root lexeme, part of speech, and morphological components (tense, gender, person, etc.) for each word in the exercise.

and Windows devices. For comparison, that is more than the total number of students in U.S. elementary and secondary schools combined. At least 80 language courses are currently available or under development[2] for the Duolingo platform. The most popular courses are for learning English, Spanish, French, and German, although there are also courses for minority languages (Irish Gaelic), and even constructed languages (Esperanto).

More than half of Duolingo students live in developing countries, where Internet access has more than tripled in the past three years (ITU and UNESCO, 2015). The majority of these students are using Duolingo to learn English, which can significantly improve their job prospects and quality of life (Pinon and Haydon, 2010).

2.1 System Overview

Duolingo uses a playfully illustrated, gamified design that combines point-reward incentives with implicit instruction (DeKeyser, 2008), mastery learning (Block et al., 1971), explanations (Fahy, 2004), and other best practices. Early research suggests that 34 hours of Duolingo is equivalent to a full semester of university-level Spanish instruction (Vesselinov and Grego, 2012).

Figure 1(a) shows an example **skill tree** for English speakers learning French. This specifies the game-like curriculum: each icon represents a **skill**, which in turn teaches a set of thematically or grammatically related words or concepts. Students tap an icon to access lessons of new material, or to practice previously-learned material. Figure 1(b) shows a screen for the French skill *Gerund*, which teaches common gerund verb forms such as *faisant* (doing) and *étant* (being). This skill, as well as several others, have already been completed by the student. However, the *Measures* skill in the bottom right of Figure 1(a) has one lesson remaining. After completing each row of skills, students "unlock" the next row of more advanced skills. This is a gamelike implementation of *mastery learning*, whereby students must reach a certain level of prerequisite knowledge before moving on to new material.

[2] https://incubator.duolingo.com

Each language course also contains a **corpus** (large database of available exercises) and a **lexeme tagger** (statistical NLP pipeline for automatically tagging and indexing the corpus; see the Appendix for details and a lexeme tag reference). Figure 1(c,d) shows an example translation exercise that might appear in the *Gerund* skill, and Figure 2 shows the lexeme tagger output for this sentence. Since this exercise is indexed with a gerund lexeme tag (*être*.V.GER in this case), it is available for lessons or practices in this skill.

The lexeme tagger also helps to provide corrective feedback. Educational researchers maintain that incorrect answers should be accompanied by *explanations*, not simply a "wrong" mark (Fahy, 2004). In Figure 1(d), the student incorrectly used the 2nd-person verb form *es* (*être*.V.PRES.P2.SG) instead of the 3rd-person *est* (*être*.V.PRES.P3.SG). If Duolingo is able to parse the student response and detect a known grammatical mistake such as this, it provides an explanation[3] in plain language. Each lesson continues until the student masters all of the **target words** being taught in the session, as estimated by a mixture model of short-term learning curves (Streeter, 2015).

2.2 Spaced Repetition and Practice

Once a lesson is completed, all the target words being taught in the lesson are added to the **student model**. This model captures what the student has learned, and estimates how well she can recall this knowledge at any given time. Spaced repetition is a key component of the student model: over time, the strength of a skill will decay in the student's long-term memory, and this model helps the student manage her practice schedule.

Duolingo uses **strength meters** to visualize the student model, as seen beneath each of the completed skill icons in Figure 1(a). These meters represent the average probability that the student can, at any moment, correctly recall a random target word from the lessons in this skill (more on this probability estimate in §3.3). At four bars, the skill is "golden" and considered fresh in the student's memory. At fewer bars, the skill has grown stale and may need practice. A student can tap the skill icon to access practice sessions and target her weakest words. For example, Figure 1(b) shows some weak words from the *Gerund* skill. Practice sessions are identical to lessons, except that the exercises are taken from those indexed with words (lexeme tags) due for practice according to student model. As time passes, strength meters continuously update and decay until the student practices.

3 Spaced Repetition Models

In this section, we describe several spaced repetition algorithms that might be incorporated into our student model. We begin with two common, established methods in language learning technology, and then present our half-life regression model which is a generalization of them.

3.1 The Pimsleur Method

Pimsleur (1967) was perhaps the first to make mainstream practical use of the spacing and lag effects, with his audio-based language learning program (now a franchise by Simon & Schuster). He referred to his method as *graduated-interval recall*, whereby new vocabulary is introduced and then tested at exponentially increasing intervals, interspersed with the introduction or review of other vocabulary. However, this approach is limited since the schedule is pre-recorded and cannot adapt to the learner's actual ability. Consider an English-speaking French student who easily learns a cognate like *pantalon* (pants), but struggles to remember *manteau* (coat). With the Pimsleur method, she is forced to practice both words at the same fixed, increasing schedule.

3.2 The Leitner System

Leitner (1972) proposed a different spaced repetition algorithm intended for use with flashcards. It is more adaptive than Pimsleur's, since the spacing intervals can increase or decrease depending on student performance. Figure 3 illustrates a popular variant of this method.

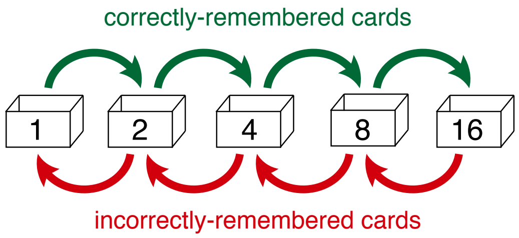


Figure 3: The Leitner System for flashcards.

The main idea is to have a few boxes that correspond to different practice intervals: 1-day, 2-day,

[3] If Duolingo cannot parse the precise nature of the mistake — e.g., because of a gross typographical error — it provides a "diff" of the student's response with the closest acceptable answer in the corpus (using Levenshtein distance).

4-day, and so on. All cards start out in the 1-day box, and if the student can remember an item after one day, it gets "promoted" to the 2-day box. Two days later, if she remembers it again, it gets promoted to the 4-day box, etc. Conversely, if she is incorrect, the card gets "demoted" to a shorter interval box. Using this approach, the hypothetical French student from §3.1 would quickly promote *pantalon* to a less frequent practice schedule, but continue reviewing *manteau* often until she can regularly remember it.

Several electronic flashcard programs use the Leitner system to schedule practice, by organizing items into "virtual" boxes. In fact, when it first launched, Duolingo used a variant similar to Figure 3 to manage skill meter decay and practice. The present research was motivated by the need for a more accurate model, in response to student complaints that the Leitner-based skill meters did not adequately reflect what they had learned.

3.3 Half-Life Regression: A New Approach

We now describe half-life regression (HLR), starting from psychological theory and combining it with modern machine learning techniques.

Central to the theory of memory is the *Ebbinghaus model*, also known as the *forgetting curve* (Ebbinghaus, 1885). This posits that memory decays exponentially over time:

$$p = 2^{-\Delta/h} \; . \qquad (1)$$

In this equation, p denotes the probability of correctly recalling an item (e.g., a word), which is a function of Δ, the *lag time* since the item was last practiced, and h, the *half-life* or measure of strength in the learner's long-term memory.

Figure 4(a) shows a forgetting curve (1) with half-life $h = 1$. Consider the following cases:

1. $\Delta = 0$. The word was just recently practiced, so $p = 2^0 = 1.0$, conforming to the idea that it is fresh in memory and should be recalled correctly regardless of half-life.

2. $\Delta = h$. The lag time is equal to the half-life, so $p = 2^{-1} = 0.5$, and the student is on the verge of being unable to remember.

3. $\Delta \gg h$. The word has not been practiced for a long time relative to its half-life, so it has probably been forgotten, e.g., $p \approx 0$.

Let $\mathbf{x}$ denote a feature vector that summarizes a student's previous exposure to a particular word, and let the parameter vector Θ contain weights that correspond to each feature variable in $\mathbf{x}$. Under the assumption that half-life should increase exponentially with each repeated exposure (a common practice in spacing and lag effect research), we let $\hat{h}_\Theta$ denote the estimated half-life, given by:

$$\hat{h}_\Theta = 2^{\Theta \cdot \mathbf{x}} \; . \qquad (2)$$

In fact, the Pimsleur and Leitner algorithms can be interpreted as special cases of (2) using a few fixed, hand-picked weights. See the Appendix for the derivation of Θ for these two methods.

For our purposes, however, we want to fit Θ empirically to learning trace data, and accommodate an arbitrarily large set of interesting features (we discuss these features more in §3.4). Suppose we have a data set $\mathcal{D} = \{\langle p, \Delta, \mathbf{x} \rangle_i\}_{i=1}^{D}$ made up of student-word practice sessions. Each data instance consists of the observed recall rate p[4], lag time Δ since the word was last seen, and a feature vector $\mathbf{x}$ designed to help personalize the learning experience. Our goal is to find the best model weights Θ^* to minimize some loss function ℓ:

$$\Theta^* = \arg\min_{\Theta} \sum_{i=1}^{D} \ell(\langle p, \Delta, \mathbf{x} \rangle_i; \Theta) \; . \qquad (3)$$

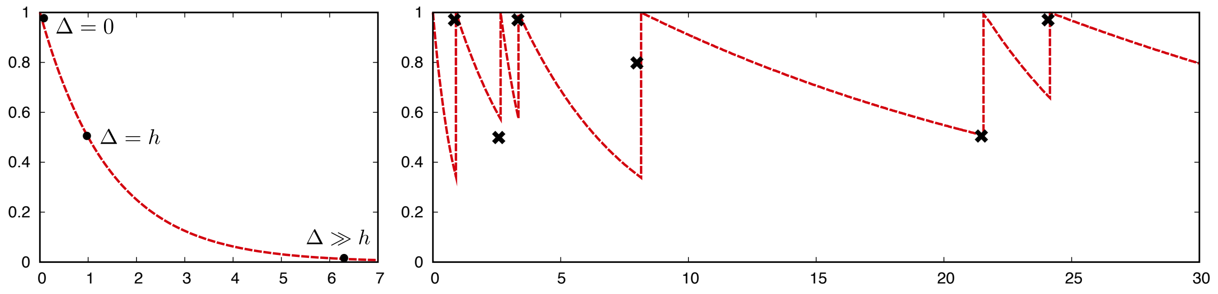
To illustrate, Figure 4(b) shows a student-word learning trace over the course of a month. Each ✖ indicates a data instance: the vertical position is the observed recall rate p for each practice session, and the horizontal distance between points is the lag time Δ between sessions. Combining (1) and (2), the model prediction $\hat{p}_\Theta = 2^{-\Delta/\hat{h}_\Theta}$ is plotted as a dashed line over time (which resets to 1.0 after each exposure, since $\Delta = 0$). The training loss function (3) aims to fit the predicted forgetting curves to observed data points for millions of student-word learning traces like this one.

We chose the L_2-regularized squared loss function, which in its basic form is given by:

$$\ell(\text{✖}; \Theta) = (p - \hat{p}_\Theta)^2 + \lambda \|\Theta\|_2^2 \; ,$$

where $\text{✖} = \langle p, \Delta, \mathbf{x} \rangle$ is shorthand for the training data instance, and λ is a parameter to control the regularization term and help prevent overfitting.

[4] In our setting, each data instance represents a full lession or practice session, which may include multiple exercises reviewing the same word. Thus p represents the *proportion* of times a word was recalled correctly in a particular session.



(a) Ebbinghaus model ($h = 1$)

(b) 30-day student-word learning trace and predicted forgetting curve

Figure 4: Forgetting curves. (a) Predicted recall rate as a function of lag time Δ and half-life $h = 1$. (b) Example student-word learning trace over 30 days: ✖ marks the observed recall rate p for each practice session, and half-life regression aims to fit model predictions $\hat{p}_\Theta$ (dashed lines) to these points.

In practice, we found it useful to optimize for the half-life h in addition to the observed recall rate p. Since we do not know the "true" half-life of a given word in the student's memory — this is a hypothetical construct — we approximate it algebraically from (1) using p and Δ. We solve for $h = \frac{-\Delta}{\log_2(p)}$ and use the final loss function:

$$\ell(\text{✖}; \Theta) = (p - \hat{p}_\Theta)^2 + \alpha(h - \hat{h}_\Theta)^2 + \lambda\|\Theta\|_2^2 ,$$

where α is a parameter to control the relative importance of the half-life term in the overall training objective function. Since ℓ is smooth with respect to Θ, we can fit the weights to student-word learning traces using gradient descent. See the Appendix for more details on our training and optimization procedures.

3.4 Feature Sets

In this work, we focused on features that were easily instrumented and available in the production Duolingo system, without adding latency to the student's user experience. These features fall into two broad categories:

- *Interaction features*: a set of counters summarizing each student's practice history with each word (lexeme tag). These include the total number of times a student has seen the word x_n, the number of times it was correctly recalled $x_\oplus$, and the number of times incorrect $x_\ominus$. These are intended to help the model make more personalized predictions.

- *Lexeme tag features*: a large, sparse set of indicator variables, one for each lexeme tag in the system (about 20k in total). These are intended to capture the inherent difficulty of each particular word (lexeme tag).

recall rate		lag (days)	feature vector $\mathbf{x}$			
p	($\oplus$/n)	Δ	x_n	$x_\oplus$	$x_\ominus$	$x_{\textit{être}.\text{V.GER}}$
1.0	(3/3)	0.6	3	2	1	1
0.5	(2/4)	1.7	6	5	1	1
1.0	(3/3)	0.7	10	7	3	1
0.8	(4/5)	4.7	13	10	3	1
0.5	(1/2)	13.5	18	14	4	1
1.0	(3/3)	2.6	20	15	5	1

Table 1: Example training instances. Each row corresponds to a data point in Figure 4(b) above, which is for a student learning the French word *étant* (lexeme tag *être*.V.GER).

To be more concrete, imagine that the trace in Figure 4(b) is for a student learning the French word *étant* (lexeme tag *être*.V.GER). Table 1 shows what $\langle p, \Delta, \mathbf{x}\rangle$ would look like for each session in the student's history with that word. The interaction features increase monotonically[5] over time, and $x_{\textit{être}.\text{V.GER}}$ is the only lexeme feature to "fire" for these instances (it has value 1, all other lexeme features have value 0). The model also includes a bias weight (intercept) not shown here.

4 Experiments

In this section, we compare variants of HLR with other spaced repetition algorithms in the context of Duolingo. First, we evaluate methods against historical log data, and analyze trained model weights for insight. We then describe two controlled user experiments where we deployed HLR as part of the student model in the production system.

[5] Note that in practice, we found that using the square root of interaction feature counts (e.g., $\sqrt{x_\oplus}$) yielded better results than the raw counts shown here.

Model	MAE↓	AUC↑	COR$_h$↑
HLR	**0.128***	0.538*	**0.201***
HLR -lex	**0.128***	0.537*	0.160*
HLR -h	0.350	0.528*	_-0.143_*
HLR -lex-h	0.350	0.528*	_-0.142_*
Leitner	0.235	**0.542***	_-0.098_*
Pimsleur	0.445	0.510*	_-0.132_*
LR	0.211	0.513*	n/a
LR -lex	0.212	0.514*	n/a
Constant $\bar{p} = 0.859$	0.175	n/a	n/a

Table 2: Evaluation results using historical log data (see text). Arrows indicate whether lower (↓) or higher (↑) scores are better. The best method for each metric is shown in bold, and statistically significant effects ($p < 0.001$) are marked with *.

4.1 Historical Log Data Evaluation

We collected two weeks of Duolingo log data, containing 12.9 million student-word lesson and practice session traces similar to Table 1 (for all students in all courses). We then compared three categories of spaced repetition algorithms:

- *Half-life regression (HLR)*, our model from §3.3. For ablation purposes, we consider four variants: with and without lexeme features (-lex), as well as with and without the half-life term in the loss function (-h).

- *Leitner and Pimsleur*, two established baselines that are special cases of HLR, using fixed weights. See the Appendix for a derivation of the model weights we used.

- *Logistic regression (LR)*, a standard machine learning[6] baseline. We evaluate two variants: with and without lexeme features (-lex).

We used the first 1 million instances of the data to tune the parameters for our training algorithm. After trying a handful of values, we settled on $\lambda = 0.1$, $\alpha = 0.01$, and learning rate $\eta = 0.001$. We used these same training parameters for HLR and LR experiments (the Leitner and Pimsleur models are fixed and do not require training).

Table 2 shows the evaluation results on the full data set of 12.9 million instances, using the first 90% for training and remaining 10% for testing. We consider several different evaluation measures for a comprehensive comparison:

- *Mean absolute error (MAE)* measures how closely predictions resemble their observed outcomes: $\frac{1}{D}\sum_{i=1}^{D}|p - \hat{p}_\Theta|_i$. Since the strength meters in Duolingo's interface are based on model predictions, we use MAE as a measure of prediction quality.

- *Area under the ROC curve (AUC)* — or the Wilcoxon rank-sum test — is a measure of ranking quality. Here, it represents the probability that a model ranks a random correctly-recalled word as more likely than a random incorrectly-recalled word. Since our model is used to prioritize words for practice, we use AUC to help evaluate these rankings.

- *Half-life correlation (COR$_h$)* is the Spearman rank correlation between $\hat{h}_\Theta$ and the algebraic estimate h described in §3.3. We use this as another measure of ranking quality.

For all three metrics, HLR with lexeme tag features is the best (or second best) approach, followed closely by HLR -lex (no lexeme tags). In fact, these are the only two approaches with MAE lower than a baseline constant prediction of the *average* recall rate in the training data (Table 2, bottom row). These HLR variants are also the only methods with positive COR$_h$, although this seems reasonable since they are the only two to directly optimize for it. While lexeme tag features made limited impact, the h term in the HLR loss function is clearly important: MAE more than doubles without it, and the -h variants are generally worse than the other baselines on at least one metric.

As stated in §3.2, Leitner was the spaced repetition algorithm used in Duolingo's production student model at the time of this study. The Leitner method did yield the highest AUC[7] values among the algorithms we tried. However, the top two HLR variants are not far behind, and they also reduce MAE compared to Leitner by least 45%.

[6] For LR models, we include the lag time $x_\triangle$ as an additional feature, since — unlike HLR — it isn't explicitly accounted for in the model. We experimented with polynomial and exponential transformations of this feature, as well, but found the raw lag time to work best.

[7] AUC of 0.5 implies random guessing (Fawcett, 2006), so the AUC values here may seem low. This is due in part to an inherently noisy prediction task, but also to a range restriction: $\bar{p} = 0.859$, so most words are recalled correctly and predictions tend to be high. Note that all reported AUC values are statistically significantly better than chance using a Wilcoxon rank sum test with continuity correction.

Lg.	Word	Lexeme Tag	θ_k
EN	camera	*camera*.N.SG	0.77
EN	ends	*end*.V.PRES.P3.SG	0.38
EN	circle	*circle*.N.SG	0.08
EN	rose	*rise*.V.PST	*-0.09*
EN	performed	*perform*.V.PP	*-0.48*
EN	writing	*write*.V.PRESP	*-0.81*
ES	liberal	*liberal*.ADJ.SG	0.83
ES	como	*comer*.V.PRES.P1.SG	0.40
ES	encuentra	*encontrar*.V.PRES.P3.SG	0.10
ES	está	*estar*.V.PRES.P3.SG	*-0.05*
ES	pensando	*pensar*.V.GER	*-0.33*
ES	quedado	*quedar*.V.PP.M.SG	*-0.73*
FR	visite	*visiter*.V.PRES.P3.SG	0.94
FR	suis	*être*.V.PRES.P1.SG	0.47
FR	trou	*trou*.N.M.SG	0.05
FR	dessous	*dessous*.ADV	*-0.06*
FR	ceci	*ceci*.PN.NT	*-0.45*
FR	fallait	*falloir*.V.IMPERF.P3.SG	*-0.91*
DE	Baby	*Baby*.N.NT.SG.ACC	0.87
DE	sprechen	*sprechen*.V.INF	0.56
DE	sehr	*sehr*.ADV	0.13
DE	den	*der*.DET.DEF.M.SG.ACC	*-0.07*
DE	Ihnen	*Sie*.PN.P3.PL.DAT.FORM	*-0.55*
DE	war	*sein*.V.IMPERF.P1.SG	*-1.10*

Table 3: Lexeme tag weights for English (EN), Spanish (ES), French (FR), and German (DE).

4.2 Model Weight Analysis

In addition to better predictions, HLR can capture the inherent difficulty of concepts that are encoded in the feature set. The "easier" concepts take on positive weights (less frequent practice resulting from longer half-lifes), while the "harder" concepts take on negative weights (more frequent practice resulting from shorter half-lifes).

Table 3 shows HLR model weights for several English, Spanish, French, and German lexeme tags. Positive weights are associated with cognates and words that are common, short, or morphologically simple to inflect; it is reasonable that these would be easier to recall correctly. Negative weights are associated with irregular forms, rare words, and grammatical constructs like past or present participles and imperfective aspect. These model weights can provide insight into the aspects of language that are more or less challenging for students of a second language.

	Daily Retention Activity		
Experiment	Any	Lesson	Practice
I. HLR (v. Leitner)	+0.3	+0.3	*-7.3**
II. HLR -lex (v. HLR)	+12.0*	+1.7*	+9.5*

Table 4: Change (%) in daily student retention for controlled user experiments. Statistically significant effects ($p < 0.001$) are marked with *.

4.3 User Experiment I

The evaluation in §4.1 suggests that HLR is a better approach than the Leitner algorithm originally used by Duolingo (cutting MAE nearly in half). To see what effect, if any, these gains have on actual student behavior, we ran controlled user experiments in the Duolingo production system.

We randomly assigned all students to one of two groups: HLR (experiment) or Leitner (control). The underlying spaced repetition algorithm determined strength meter values in the skill tree (e.g., Figure 1(a)) as well as the ranking of target words for practice sessions (e.g., Figure 1(b)), but otherwise the two conditions were identical. The experiment lasted six weeks and involved just under 1 million students.

For evaluation, we examined changes in *daily retention*: what percentage of students who engage in an activity return to do it again the following day? We used three retention metrics: any activity (including contributions to crowdsourced translations, online forum discussions, etc.), new lessons, and practice sessions.

Results are shown in the first row of Table 4. The HLR group showed a slight increase in overall activity and new lessons, but a significant decrease in practice. Prior to the experiment, many students claimed that they would practice instead of learning new material "just to keep the tree gold," but that practice sessions did not review what they thought they needed most. This drop in practice — plus positive anecdotal feedback about stength meter quality from the HLR group — led us to believe that HLR was actually better for student engagement, so we deployed it for all students.

4.4 User Experiment II

Several months later, active students pointed out that particular words or skills would decay rapidly, regardless of how often they practiced. Upon closer investigation, these complaints could be traced to lexeme tag features with highly negative weights in the HLR model (e.g., Table 3). This implied that some feature-based overfitting had occurred, despite the L_2 regularization term in the training procedure. Duolingo was also preparing to launch several new language courses at the time, and no training data yet existed to fit lexeme tag feature weights for these new languages.

Since the top two HLR variants were virtually tied in our §4.1 experiments, we hypothesized that using interaction features alone might alleviate both student frustration and the "cold-start" problem of training a model for new languages. In a follow-up experiment, we randomly assigned all students to one of two groups: HLR -lex (experiment) and HLR (control). The experiment lasted two weeks and involved 3.3 million students.

Results are shown in the second row of Table 4. All three retention metrics were significantly higher for the HLR -lex group. The most substantial increase was for any activity, although recurring lessons and practice sessions also improved (possibly as a byproduct of the overall activity increase). Anecdotally, vocal students from the HLR -lex group who previously complained about rapid decay under the HLR model were also positive about the change.

We deployed HLR -lex for all students, and believe that its improvements are at least partially responsible for the consistent 5% month-on-month growth in active Duolingo users since the model was launched.

5 Other Related Work

Just as we drew upon the theories of Ebbinghaus to derive HLR as an empirical spaced repetition model, there has been other recent work drawing on other (but related) theories of memory.

ACT-R (Anderson et al., 2004) is a cognitive architecture whose declarative memory module[8] takes the form of a power function, in contrast to the exponential form of the Ebbinghaus model and HLR. Pavlik and Anderson (2008) used ACT-R predictions to optimize a practice schedule for second-language vocabulary, although their setting was quite different from ours. They assumed fixed intervals between practice exercises within the same laboratory session, and found that they could improve short-term learning within a session. In contrast, we were concerned with making accurate recall predictions between multiple sessions "in the wild" on longer time scales. Evidence also suggests that manipulation between sessions can have greater impact on long-term learning (Cepeda et al., 2006).

Motivated by long-term learning goals, the multiscale context model (MCM) has also been proposed (Mozer et al., 2009). MCM combines two modern theories of the spacing effect (Staddon et al., 2002; Raaijmakers, 2003), assuming that each time an item is practiced it creates an additional item-specific forgetting curve that decays at a different rate. Each of these forgetting curves is exponential in form (similar to HLR), but are combined via weighted average, which approximates a power law (similar to ACT-R). The authors were able to fit models to controlled laboratory data for second-language vocabulary and a few other memory tasks, on times scales up to several months. We were unaware of MCM at the time of our work, and it is unclear if the additional computational overhead would scale to Duolingo's production system. Nevertheless, comparing to and integrating with these ideas is a promising direction for future work.

There has also been work on more heuristic spaced repetition models, such as SuperMemo (Woźniak, 1990). Variants of this algorithm are popular alternatives to Leitner in some flashcard software, leveraging additional parameters with complex interactions to determine spacing intervals for practice. To our knowledge, these additional parameters are hand-picked as well, but one can easily imagine fitting them empirically to real student log data, as we do with HLR.

6 Conclusion

We have introduced *half-life regression (HLR)*, a novel spaced repetition algorithm with applications to second language acquisition. HLR combines a psycholinguistic model of human memory with modern machine learning techniques, and generalizes two popular algorithms used in language learning technology: Leitner and Pimsleur. We can do this by incorporating arbitrarily rich features and fitting their weights to data. This approach is significantly more accurate at predicting student recall rates than either of the previous methods, and is also better than a conventional machine learning approach like logistic regression.

[8] Declarative (specifically semantic) memory is widely regarded to govern language vocabulary (Ullman, 2005).

One result we found surprising was that lexeme tag features failed to improve predictions much, and in fact seemed to frustrate the student learning experience due to over-fitting. Instead of the sparse indicator variables used here, it may be better to decompose lexeme tags into denser and more generic features of tag *components*[9] (e.g., part of speech, tense, gender, case), and also use corpus frequency, word length, etc. This representation might be able to capture useful and interesting regularities without negative side-effects.

Finally, while we conducted a cursory analysis of model weights in §4.2, an interesting next step would be to study such weights for even deeper insight. (Note that using lexeme tag component features, as suggested above, should make this anaysis more robust since features would be less sparse.) For example, one could see whether the ranking of vocabulary and/or grammar components by feature weight is correlated with external standards such as the CEFR (Council of Europe, 2001). This and other uses of HLR hold the potential to transform data-driven curriculum design.

Data and Code

To faciliatate research in this area, we have publicly released our data set and code from §4.1: https://github.com/duolingo/halflife-regression.

Acknowledgments

Thanks to our collaborators at Duolingo, particularly Karin Tsai, Itai Hass, and André Horie for help gathering data from various parts of the system. We also thank the anonymous reviewers for suggestions that improved the final manuscript.

References

J.R. Anderson, D. Bothell, M.D. Byrne, S. Douglass, C. Libiere, and Y. Qin. 2004. An intergrated theory of mind. *Psychological Review*, 111:1036–1060.

R.C. Atkinson. 1972. Optimizing the learning of a second-language vocabulary. *Journal of Experimental Psychology*, 96(1):124–129.

J.H. Block, P.W. Airasian, B.S. Bloom, and J.B. Carroll. 1971. *Mastery Learning: Theory and Practice*. Holt, Rinehart, and Winston, New York.

K.C. Bloom and T.J. Shuell. 1981. Effects of massed and distributed practice on the learning and retention of second language vocabulary. *Journal of Educational Psychology*, 74:245–248.

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.

Council of Europe. 2001. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment*. Cambridge University Press.

R. DeKeyser. 2008. Implicit and explicit learning. In *The Handbook of Second Language Acquisition*, chapter 11, pages 313–348. John Wiley & Sons.

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

J.J. Donovan and D.J. Radosevich. 1999. A meta-analytic review of the distribution of practice effect: Now you see it, now you don't. *Journal of Applied Psychology*, 84(5):795–805.

J. Duchi, E. Hazan, and Y. Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

H. Ebbinghaus. 1885. *Memory: A Contribution to Experimental Psychology*. Teachers College, Columbia University, New York, NY, USA.

P.J. Fahy. 2004. Media characteristics and online learning technology. In T. Anderson and F. Elloumi, editors, *Theory and Practice of Online Learning*, pages 137–171. Athabasca University.

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

M.L. Forcada, M. Ginestí-Rosell, J. Nordfalk, J. O'Regan, S. Ortiz-Rojas, J.A. Pérez-Ortiz, F. Sánchez-Martínez, G. Ramírez-Sánchez, and F.M. Tyers. 2011. Apertium: A free/open-source platform for rule-based machine translation. *Machine Translation*, 25(2):127–144. http://wiki.apertium.org/wiki/Main_Page.

ITU and UNESCO. 2015. The state of broadband 2015. Technical report, September.

S. Leitner. 1972. *So lernt man lernen. Angewandte Lernpsychologie – ein Weg zum Erfolg*. Verlag Herder, Freiburg im Breisgau, Germany.

A.W. Melton. 1970. The situation with respect to the spacing of repetitions and memory. *Journal of Verbal Learning and Verbal Behavior*, 9:596–606.

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

[9] Engineering-wise, each lexeme tag (e.g., *être*.V.GER) is represented by an ID in the system. We used indicator variables in this work since the IDs are readily available; the overhead of retreiving all lexeme components would be inefficient in the production system. Of course, we could optimize for this if there were evidence of a significant improvement.

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

P. Pimsleur. 1967. A memory schedule. *Modern Language Journal*, 51(2):73–75.

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.

J.G.W. Raaijmakers. 2003. Spacing and repetition effects in human memory: Application of the sam model. *Cognitive Science*, 27(3):431–452.

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

J.E.R. Staddon, I.M. Chelaru, and J.J. Higa. 2002. Habituation, memory and the brain: The dynamics of interval timing. *Behavioural Processes*, 57(2):71–88.

M. Streeter. 2015. Mixture modeling of individual learning curves. In *Proceedings of the International Conference on Educational Data Mining (EDM)*.

M.T. Ullman. 2005. A cognitive neuroscience perspective on second language acquisition: The declarative/procedural model. In C. Sanz, editor, *Mind and Context in Adult Second Language Acquisition: Methods, Theory, and Practice*, pages 141–178. Georgetown University Press.

R. Vesselinov and J. Grego. 2012. Duolingo effectiveness study. Technical report, Queens College, City University of New York.

Wikimedia Foundation. 2002. Wiktionary: A wiki-based open content dictionary, retrieved 2012–2015. https://www.wiktionary.org.

P.A. Woźniak. 1990. Optimization of learning. Master's thesis, University of Technology in Poznań.

A Appendix

A.1 Lexeme Tagger Details

We use a lexeme tagger, introduced in §2, to analyze and index the learning corpus and student responses. Since Duolingo courses teach a moderate set of words and concepts, we do not necessarily need a complete, general-purpose, multi-lingual NLP stack. Instead, for each language we use a finite state transducer (FST) to efficiently parse candidate lexeme tags[10] for each word. We then use a hidden Markov model (HMM) to determine which tag is correct in a given context.

Abbreviation	Meaning
ACC	accusative case
ADJ	adjective
ADV	adverb
DAT	dative case
DEF	definite
DET	determiner
FORM	formal register
F	feminine
GEN	genitive case
GER	gerund
IMPERF	imperfective aspect
INDF	indefinite
INF	infinitive
M	masculine
N	noun
NT	neuter
P1/P2/P3	1st/2nd/3rd person
PL	plural
PN	pronoun
PP	past participle
PRESP	present participle
PRES	present tense
PST	past tense
SG	singular
V	verb

Table 5: Lexeme tag component abbreviations.

Consider the following two Spanish sentences: '*Yo como manzanas*' ('I eat apples') and '*Corro como el viento*' ('I run like the wind'). For both sentences, the FST parses the word *como* into the lexeme tag candidates *comer*.V.PRES.P1.SG ([I] eat) and *como*.ADV.CNJ (like/as). The HMM then disambiguates between the respective tags for each sentence. Table 5 contains a reference of the abbreviations used in this paper for lexeme tags.

A.2 Pimsleur and Leitner Models

As mentioned in §3.3, the Pimsleur and Leitner algorithms are special cases of HLR using fixed, hand-picked weights. To see this, consider the original practice interval schedule used by Pimsleur (1967): 5 sec, 25 sec, 2 min, 10 min, 1 hr, 5 hr, 1 day, 5 days, 25 days, 4 months, and 2 years. If we interpret this as a sequence of $\hat{h}_\Theta$ half-lifes (i.e., students should practice when $\hat{p}_\Theta = 0.5$), we can rewrite (2) and solve for $\log_2(\hat{h}_\Theta)$ as a linear

[10] The lexeme tag set is based on a large morphology dictionary created by the Apertium project (Forcada et al., 2011), which we supplemented with entries from Wiktionary (Wikimedia Foundation, 2002) and other sources. Each Duolingo course teaches about 3,000–5,000 lexeme tags.

equation. This yields $\Theta = \{x_n : \text{2.4}, x_b : \text{-16.5}\}$, where x_n and x_b are the number of practices and a bias weight (intercept), respectively. This model perfectly reconstructs Pimsleur's original schedule in days ($r^2 = $ 0.999, $p \ll$ 0.001). Analyzing the Leitner variant from Figure 3 is even simpler: this corresponds to $\Theta = \{x_\oplus : 1, x_\ominus : \text{-1}\}$, where $x_\oplus$ is the number of past correct responses (i.e., doubling the interval), and $x_\ominus$ is the number of incorrect responses (i.e., halving the interval).

A.3 Training and Optimization Details

The complete objective function given in §3.3 for half-life regression is:

$$\begin{aligned}\ell(\langle p, \Delta, \mathbf{x}\rangle; \Theta) &= (p - \hat{p}_\Theta)^2 \\ &\quad + \alpha(h - \hat{h}_\Theta)^2 + \lambda \|\Theta\|_2^2 \, .\end{aligned}$$

Substituting (1) and (2) into this equation produces the following more explicit formulation:

$$\begin{aligned}\ell(\langle p, \Delta, \mathbf{x}\rangle; \Theta) &= \left(p - 2^{-\frac{\Delta}{2^{\Theta \cdot \mathbf{x}}}}\right)^2 \\ &\quad + \alpha \left(\frac{-\Delta}{\log_2(p)} - 2^{\Theta \cdot \mathbf{x}}\right)^2 \\ &\quad + \lambda \|\Theta\|_2^2 \, .\end{aligned}$$

In general, the search for Θ^* weights to minimize ℓ cannot be solved in closed form, but since it is a smooth function, it can be optimized using gradient methods. The partial gradient of ℓ with respect to each θ_k weight is given by:

$$\begin{aligned}\frac{\partial \ell}{\partial \theta_k} &= 2(\hat{p}_\Theta - p) \ln^2(2) \hat{p}_\Theta \left(\frac{\Delta}{\hat{h}_\Theta}\right) x_k \\ &\quad + 2\alpha \left(\hat{h}_\Theta + \frac{\Delta}{\log_2(p)}\right) \ln(2) \hat{h}_\Theta x_k \\ &\quad + 2\lambda \theta_k \, .\end{aligned}$$

In order to fit Θ to a large amount of student log data, we use AdaGrad (Duchi et al., 2011), an online algorithm for stochastic gradient descent (SGD). AdaGrad is typically less sensitive to the learning rate parameter η than standard SGD, by dynamically scaling each weight update as a function of how often the corresponding feature appears in the training data:

$$\theta_k^{(+1)} := \theta_k - \eta \left[\mathrm{c}(x_k)^{-\frac{1}{2}}\right] \frac{\partial \ell}{\partial \theta_k} \, .$$

Here $\mathrm{c}(x_k)$ denotes the number of times feature x_k has had a nonzero value so far in the SGD pass through the training data. This is useful for training stability when using large, sparse feature sets (e.g., the lexeme tag features in this study). Note that to prevent computational overflow and underflow errors, we bound $\hat{p}_\Theta \in$ [0.0001, 0.9999] and $\hat{h}_\Theta \in$ [15 min, 9 months] in practice.