Linguistic Skill Modeling for Second Language Acquisition

Brian Zylich
bzylich@umass.edu
University of Massachusetts Amherst
Amherst, Massachusetts

Andrew Lan
andrewlan@cs.umass.edu
University of Massachusetts Amherst
Amherst, Massachusetts

ABSTRACT

To adapt materials for an individual learner, intelligent tutoring systems must estimate their knowledge or abilities. Depending on the content taught by the tutor, there have historically been different approaches to student modeling. Unlike common skill-based models used by math and science tutors, second language acquisition (SLA) tutors use memory-based models since there are many tasks involving memorization and retrieval, such as learning the meaning of a word in a second language. Based on estimated memory strengths provided by these memory-based models, SLA tutors are able to identify the optimal timing and content of retrieval practices for each learner to improve retention. In this work, we seek to determine whether skill-based models can be combined with memory-based models to improve student modeling and especially retrieval practice performance for SLA. In order to define skills in the context of SLA, we develop methods that can automatically extract multiple types of linguistic features from words. Using these features as skills, we apply skill-based models to a real-world SLA dataset. Our main findings are as follows. First, incorporating lexical features to represent individual words as skills in skill-based models outperforms existing memory-based models in terms of recall probability prediction. Second, incorporating additional morphological and syntactic features of each word via multiple-skill tagging of each word further improves the skill-based models. Third, incorporating semantic features, like word embeddings, to model similarities between words in a learner's practice history and their effects on memory also improves the models and appears to be a promising direction for future research.

CCS CONCEPTS

• **Applied computing** → **Computer-assisted instruction**; *Distance learning*.

KEYWORDS

student modeling, second language acquisition, memory decay

ACM Reference Format:
Brian Zylich and Andrew Lan. 2021. Linguistic Skill Modeling for Second Language Acquisition. In *LAK21: 11th International Learning Analytics and Knowledge Conference (LAK21), April 12–16, 2021, Irvine, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3448139.3448153

LAK21, April 12–16, 2021, Irvine, CA, USA

ACM ISBN 978-1-4503-8935-8/21/04…$15.00
https://doi.org/10.1145/3448139.3448153

1 INTRODUCTION

Intelligent tutors and computer-assisted learning are seeing an uptick in usage recently due to advancements in technology and recent shifts to remote learning [39]. These systems have the ability to customize their interactions with each individual learner, which enables them to provide personalized interventions at a much larger scale than one-on-one tutoring. For example, an intelligent math tutor might identify key skills that a learner has not mastered based on items they have attempted and address these skill deficiencies through targeted explanations and additional practice opportunities. Similarly, an intelligent second language acquisition (SLA) tutor might identify the words (and their contexts) with which a learner tends to struggle and provide more practice opportunities, especially retrieval practice [15]. In both settings, an intelligent tutor needs some way to reason about a learner's knowledge state given their past performance in order to make informed decisions regarding what interventions and practice material to provide during their next learning session.

For intelligent SLA tutors, the problem of modeling a learner's knowledge state has historically been addressed using pure memory models. Many memory-based models draw inspiration from the Ebbinghaus forgetting curve [10], which theorizes that memory decays exponentially over time. This theory supports the idea of the *spacing effect* and the *lag effect*. The *spacing effect* observes that people learn better through spaced practice over time rather than practicing a lot within a short duration, and the *lag effect* observes that learning is further improved if the spacing between practice sessions increases gradually over time. The Pimsleur method [31] was one of the first to consider the spacing and lag effects. In the Pimsleur method, vocabulary items are introduced and tested at a set of exponentially increasing time intervals that applies to every learner. The Leitner system [21] expands upon this method, adapting the spacing based on whether the learner recalls the item correctly or not. More recently, the authors of [36] introduced half-life regression (HLR), which models knowledge of each item as an exponentially decaying function of time, where each item has a different "half-life" or decay rate that adapts to a learner's practice history. The use of memory models, such as these, is supported by multiple studies that demonstrate the benefits of retrieval practice on memory retention [3, 20].

As an alternative to pure memory models, skill-based models are widely used for modeling learner knowledge in intelligent tutors for math or science, where skills are well established and easy to associate with each practice item. These models, often developed in the knowledge tracing (KT) framework [8], have been widely applied to science, technology, engineering, and mathematics (STEM) learning contexts; typical variants include Bayesian knowledge tracing [28], factor models [4, 6, 29, 42], and deep learning-based models [12, 27, 30, 44, 47]. These skill-based models use past performance

to estimate learners' current knowledge level of each skill and track the evolution over time as they practice more items. The factor-based KT models incorporate constant parameters that correspond to learner abilities and item difficulties, inspired by classic item response theory models [24, 32, 33], while other KT models do not have these parameters. In particular, the recent difficulty, student ability, skill, and student skill practice history (DAS3H) model [6] recognizes that skills can also have varying degrees of difficulty; it combines the elements of skill, factor, and memory models by capturing learner ability, item and skill difficulty, and practice history on varying time scales and excels at modeling learner skill knowledge evolution and predicting learner performance.

Perhaps due to their different origins in cognitive theories, memory-based models and skill-based models are almost always applied to their own domains. Despite a relatively large amount of SLA data [35] being made available recently, more modern, deep learning-based KT models have not been extensively applied to memory data. Similarly, very few recent models use ideas in memory-based models such as exponential forgetting curves [12]. Nevertheless, these models are useful in various application contexts. Student models have been used to improve intelligent tutors by allowing them to select practice items to review before they are forgotten [23, 34]. As an alternative to constructing instructional policy manually, student models can be used to directly optimize instructional policies to boost learning [22]. Recent works have also incorporated student models into reinforcement learning in order to simulate a learner's performance over time and compare different methods of personalizing review schedules [38, 40, 45].

1.1 Contributions

In this paper, we investigate whether there are "skill practice" aspects of SLA and in particular word memorization tasks. To this end, we compare the performance of memory-based models and skill-based models on the task of student modeling and predicting recall probability using a large-scale real world SLA dataset. The novelty of our approach comes from how we define skills for SLA. In the absence of a clear skill hierarchy, we use linguistic features of words to capture skills related to vocabulary and grammar knowledge that are shared between words. Specifically, we explore lexical, morphological/syntactic, and semantic features as proxies for skills in these skill and factor-based models. The most basic of these are lexical features, which include using the word itself or the word's lemmatized form as proxies for skills. We find that simply using these lexical features as skills improves performance over commonly used memory-based models. Next, we add multiple skills to each lexical item using morphological and syntactic features such as part-of-speech, tense, gender, case, number, etc. Multiple-skill tagging in this manner further boosts student modeling performance. Finally, we explore semantic features by augmenting our skill-based model with a neural network to learn the similarity between semantic word embeddings that correspond to each item reviewed by the learner in past practices. These learned semantic features outperform versions of the model using lexical and morphological/syntactic features and result in a 21% relative improvement over the prior state-of-the-art for the dataset we use.

2 BACKGROUND

2.1 Half-Life Regression

Half-life regression (HLR) [36] is a model for SLA spaced repetition practice developed by Duolingo. The key component of this model is the "half-life" of an item in the learner's long-term memory, which dictates how long it will be retained in their memory. This "half-life" is parameterized as a function of the difficulty of the item itself and the learner's personal learning history. Based on the previously mentioned exponential forgetting curve, HLR models $p \in [0, 1]$, the probability of successfully recalling an item after some amount of time since last practicing that item, Δ, as

$$p = 2^{-\Delta/h}, \tag{1}$$

where h refers to the half-life. Given a set of features, x, that relate to a learner's practice history, the half-life is parameterized as

$$\hat{h} = 2^{\boldsymbol{\theta}^T \mathbf{x}},$$

where $\boldsymbol{\theta}$ refers to the regression coefficient vector learned through regression. Specifically, the features used by this system include: practice history with each item (number of times seen, number of correct recalls, number of incorrect recalls) and lexical features (sparse indicator variables for each lexeme identifier in the system). Each item pertains to a word and its associated syntactic and morphological tags. The lexeme identifier is a unique identifier for the word combined with these syntactic and morphological tags. Intuitively, the indicator variables for each unique lexeme identifer enable the model to learn which items are easier or more difficult to memorize, which is independent of the amount of practice with the item. Then, the practice history variables allow the model to make more personalized predictions. Using the observed recall probability p, The optimization of HLR over a dataset, $\mathcal{D} = \{\langle p, \Delta, \mathbf{x}\rangle_i\}_{i=1}^{D}$ where D denotes the total number of recall attempts across all learners and i indexes each recall attempt, is done as

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \Sigma_{i=1}^{D} l(\langle p, \Delta, \mathbf{x}\rangle_i; \boldsymbol{\theta}) + \lambda ||\boldsymbol{\theta}||_2^2,$$

where the loss function l is given by

$$l(\langle p, \Delta, \mathbf{x}\rangle_i; \boldsymbol{\theta}) = (p_i - \hat{p}_i)^2 + \alpha(h - \hat{h}_i)^2, \tag{2}$$

where $\hat{p}_i$ denotes the estimated recall probability. The loss function uses the standard squared loss to minimize the difference between predicted and true probability of recall, and the loss includes standard ℓ_2-regularization to prevent overfitting. In addition to these standard components, they find that it is useful to include a squared loss that minimizes the difference between the predicted and true half-life. Intuitively, this additional term helps because in some cases there may be a relatively small difference in the recall probabilities between instances since their values are bounded between 0 and 1, but the half-life associated with each may provide a stronger signal that can be used to learn to differentiate between these different instances more precisely. Since the actual half-life is unobserved, they approximate it from the observed probability of successful recall as

$$h = -\frac{\Delta}{\log_2(p)}.$$

Building off of HLR, the authors of [46] augment the HLR model to include more features such as learner identifiers, word complexity (predicted using a pre-trained model [13]), word concreteness, percent of a population that knew the word, and word frequency in an external corpus. Word concreteness and percent known are taken from [2], and word frequencies are taken from [41]. They also incorporate complexity into the modeling of the forgetting curve, following the intuition that it is easier to remember a word that is less complex and likewise more difficult to remember a word that is more complex. This intuition is captured by introducing a multiplicative complexity score, $C_i > 0$, associated with each word, into Equation 1 as

$$p = 2^{-\frac{\Delta \cdot C_i}{h}}.$$

HLR does not take into account any advanced linguistic features with respect to the item (or how that item is related to other items within the system and their associated practice histories). In fact, they find that the lexeme tag features failed to improve the predictive accuracy of recall probability and suggest that it might be better to decompose lexeme tags into denser features like part-of-speech, tense, gender, case, corpus frequency, word length, etc. The experiments conducted by the authors of [46] found that added features on word complexity, concreteness, percent known, and word frequency do not significantly improve model performance. Moreover, these features were derived using human judgment surveys, external corpora, and pre-trained models that were specific to the English language. Therefore, they are not necessarily transferable to the many other languages that SLA systems such as Duolingo offer. In contrast, we explore linguistic features that can be automatically applied to virtually any new language, and we show that incorporating them leads to an increase in performance over HLR.

2.2 DAS3H

While half-life regression is primarily a memory-based model, more recent models, such as DAS3H [6] and DASH [25], have incorporated both factor analysis and ideas from memory-based models. The DAS3H [6] model, detailed below, takes item difficulty (δ_j), student ability (α_s), skill, and student skill practice history into account for performance prediction. In SLA, an item refers to a vocabulary word and skills relate to the knowledge components of the secondary language that are required to successfully recall a word. We define the skills that we use in more depth in Section 4. DAS3H predicts the probability that a learner s answers an item j correctly at time t as

$$P(y_{s,j,t} = 1) = \sigma(\alpha_s - \delta_j + \sum_{k \in S(j)} \beta_k + h(H_{s,j,t})),$$

where $y_{s,j,t} \in \{0, 1\}$ denotes the binary-valued correctness of a learner's attempt to respond to an item, where 1 indicates a correct attempt and $\sigma(\cdot)$ denotes the sigmoid function. β_k is the easiness parameter associated with each skill, k, that is covered by item j. h is a function that synthesizes the learner's history, $H_{s,j,t}$, of past attempts and past correct (and their temporal distribution) for all skills associated with item j. Specifically, we use

$$h(H_{s,j,t}) = \sum_{k \in S(j)} \left(\sum_{w \in W} \theta_{k,w} \log(1 + c_{s,k,w}) - \sum_{w \in W} \theta'_{k,w} \log(1 + a_{s,k,w}) \right),$$

where $S(j)$ denotes the set of skills (or knowledge components) associated with item j. This formulation encodes the number of past attempts, $a_{s,k,w}$, and the number of correct past attempts, $c_{s,k,w}$, for each time window $w \in W$; and $\theta_{k,w}$ and $\theta'_{k,w}$ correspond to the regression coefficients. By capturing past attempts and correct past attempts, DAS3H models learners' abilities related to each skill and keeps track of changes over time. We use log counts to model the diminishing returns of repeated practice. Formally, $a_{s,k,w}$ and $c_{s,k,w}$ are defined as follows:

$$a_{s,k,w} = \sum_{\tau \in [t-w,t)} \mathbf{1}_{k \in S(j(\tau))}$$

$$c_{s,k,w} = \sum_{\tau \in [t-w,t)} \mathbf{1}_{k \in S(j(\tau)), y_{s,j(\tau),\tau}=1},$$

where $\mathbf{1}$ denotes the indicator function. This means that if a past attempt was within a certain window of time from the current example, it is included in the computation for that window, and otherwise it is ignored. DAS3H [6] and DASH [25] both use the same five time windows: $W = \{1 \text{ hour}, 1 \text{ day}, 1 \text{ week}, 1 \text{ month}, \infty\}$.

DAS3H builds off of DASH [25]. DASH accounts for the difficulty of items, ability of the student, and the student history. DASH also incorporates the learning and forgetting processes by using a set of overlapping time windows, where log counts are used to represent diminishing returns of repeated practice within a time window. DAS3H extends DASH, giving the model the ability to handle items tagged with multiple skills and enabling the impact of past practice on present performance to vary between skills.

We note that DAS3H and many other approaches, including the family of deep learning-based methods below, have not frequently been applied to data generated in SLA settings. Instead, they have been primarily applied to data generated in STEM learning contexts, such as math [5, 11, 43], algebra [37], and electrical engineering [19]. Only one dataset contains data generated in language learning contexts (preparation for an English as a secondary language test) [7], although it is unclear whether the tasks learners worked on in that dataset involved memorization tasks.

2.3 Deep Knowledge Tracing

Recently, there has been a wave of new models applying deep learning and neural network techniques to the knowledge tracing problem to improve the predictive accuracy on future learner performance [12, 26, 30]. Deep Knowledge Tracing (DKT) [30] is the simplest and most widely referenced method, applying a long short-term memory (LSTM) network to the KT problem. DKT considers a learner s's practice history as a sequence of $(q_{s,t}, y_{s,t})$ pairs for the item, $q_{s,t}$, and binary-valued answer correctness, $y_{s,t}$, at each time step t. The $(q_{s,t}, y_{s,t})$ pair is one-hot encoded and projected into a latent vector embedding space where the embeddings of each item and answer are learnable. Then, the LSTM recurrent unit uses

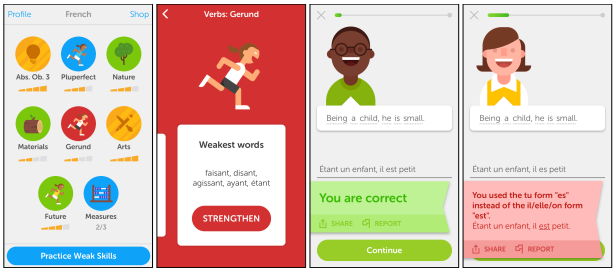


Figure 1: From left to right, this figure from [36] shows examples of different Duolingo modules, the skills contained within one such module, a correct response to a translation task, and an incorrect response to a translation task.

these item-answer embeddings to update the representation of the learner's knowledge state for the current time step, t. Finally, a feed-forward fully-connected neural network uses the learner's latent knowledge state to predict that learner's recall at time step t for each possible item. We will explore the performance of DKT on SLA data.

3 DATA

Duolingo[1] is a widely used, free online language learning tool that offers lessons in dozens of languages and has millions of active learners. In Duolingo, as seen in Figure 1, there are different modules (e.g. greetings, numbers, family, etc.) that a learner can select to learn or review. When a learner selects a module, they enter a *session* where they complete some number of *tasks* until a mastery requirement is achieved. During a session, a number of vocabulary items will be presented to the learner in the forms of fill-in-the-blank, translation, matching, or other similar tasks. We note that the same item may appear more than once within a session since they can be involved in multiple tasks.

The dataset from [35], which we use in this work, splits a session into separate data points for each unique vocabulary item, a setup that matches that for some skill-based models such as DAS3H and other factor analysis-based models. Since an item may appear multiple times in the same session, we follow [36] and use the number of correct recalls and number of attempted recalls, i.e., tasks involving the item, to approximate the probability of recalling the item in a session. We note that the dataset does not include the context surrounding the practice of each word, e.g., the sentence or phrase being translated. The dataset also contains the following features: time since the item was last seen, learner identifier, language being learned, language of the user interface, lemma of the item, the associated part of speech (POS) and other morphological tags, the number of past attempts on this item, and the number of past correct attempts on this item. In total, the dataset contains 12.8 million data points from 115,222 different learners, collected over the course of two weeks (although their learning history contains information on the past beyond these two weeks). The dataset contains examples from learners studying English, French, German, Italian, Spanish, and Portuguese. The user interface languages were English, Italian, Spanish, and Portuguese. We use the user interface language as a proxy for a learner's native language.

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

4 METHODOLOGY

In this section, we detail our extension of the DAS3H model [6] to incorporate rich linguistic features for SLA tasks. Our formulation has the basic form of

$$p_{s,j,t} = \sigma(\alpha_s - \delta_j + \lambda_{s,j} + \sum_{k \in S(j)} \beta_k + h(H_{s,j,t})),$$

where we add a language similarity parameter, $\lambda_{s,j}$, associated with a native language-target language pair, and our main contribution is to incorporate different types of linguistic features into the history feature vector $h_\theta(H_{s,j,t})$. We also note that in our case the successful recall probability $p_{s,j,t} \in [0, 1]$ has a different range than that in the DAS3H model ($\{0, 1\}$).

For the purposes of this work, we consider an SLA *item* to relate to a specific lexeme identifier (19,279 unique) in the language the learner is studying. The lexeme identifier is a unique identifier for a word along with its specific combination of POS and morphological tags. Thus, various items may refer to the same word if the word is used differently in multiple contexts. In the following subsections, we introduce methods for extracting *skills* for each word based on lexical (Section 4.1), morphological/syntactic (Section 4.2), and semantic (Section 4.3) features.

4.1 Lexical Skills

There are two ways that we can choose to label skills using lexical features: by vocabulary word (13,488 unique) and by lemmatized vocabulary word (9,913 unique). Lemmatization is a common technique in natural language processing used to group together inflected forms of a word, e.g. the lemma "hermano" ("brother" in Spanish) is associated with the inflected forms "hermano", "hermana", "hermanos", and "hermanas" that vary in gender or number. Whether we use the word itself or the word's lemma, there will only be a single skill associated with each item. In the following subsections, we use other ways to automatically tag each item so that they cover multiple underlying skills.

4.2 Morphological and Syntactic Skills

So far, we have treated each item as though the successful recall of that item depends only upon a learner's memory of the associated word. In reality, there are often multiple skill components required to successfully recall an item. Consider the task of translating the English phrase "we ate" into the Spanish word "comimos". This task requires knowledge of the infinitive form of the verb "to eat" ("comer") in addition to knowledge of conjugating regular "-er" verbs in the past tense for the first person plural form ("-imos").

In this group of models, we strive to increase coverage of these extra skills by using morphological and syntactic features as additional skills for each item, in conjunction with the lexical skills outlined in Section 4.1. We explore two methods of generating these features: Section 4.2.1 details an unsupervised subword tokenization method that identifies common subwords that comprise a word and treats each of these subwords as an additional skill, and Section 4.2.2 details a a tag extraction method that uses pretrained

models to label words with various morphological and syntactic tags, which can then be used as skills.

4.2.1 Subword Tokenization as Skill Extraction. Based on the previous motivating example, a subword tokenizer [18] might be useful since it is trained not only to tokenize text into whole words but also to tokenize less common words into more common subwords. In the example, these subwords might be "com-" and "-imos" that respectively relate to the base form of the verb and the knowledge of conjugating that verb for a particular tense and person.

Subwords can be viewed as morphological features of an original word. We apply a subword tokenizer to automatically identify such subwords for each vocabulary word, and we treat each subword as an additional morphological skill. Subwords can be as simple as individual characters or arbitrarily long. In natural language processing, this tokenization process mitigates the out-of-vocabulary problem, enabling a model trained on a corpus with a limited vocabulary to generalize to previously unseen words. For our purposes, the subword tokenizer will be used to find the n-best subword decompositions of a particular word. Thus, for a common verb such as "comer" (to eat), it is likely that the whole word would be in the vocabulary as the most likely tokenization. If we then take the second best tokenization, the decomposition might be "com-er" since the subword "com-" is used in all of the verb forms ("como", "comes", "come", etc.) and the subword "-er" is one of the three main suffixes of infinitive verbs in Spanish. These different subwords might then help us model skills related to verb conjugation, gender agreement, prefixes, suffixes, etc. Together, these subword skills also have the potential to improve performance of the model on tracking learner knowledge for words whose subwords are included in the training data but the words themselves are not. However, subwords are noisy approximations of morphological features and are prone to making erroneous connections between words. Nevertheless, they add an easy-to-obtain source of information to our models since they can be automatically extracted.

To apply the subword skills to the DAS3H model, we formulate a Q-matrix [1]; an example is shown in Table 1. We train a separate subword tokenizer with a vocabulary size of 5000 for each language in the dataset, using both the vocabulary words and their lemmatized versions as the training data for the tokenizers. To create the skills for a given word, we choose the top 2 segmentations of the word and use each of the unique words or subwords as individual skills related to that item. This results in a total of 10,431 unique skills, resulting in an average number of 4.137 skills associated with each item.

4.2.2 Syntactic and Morphological Tagging. As an alternative to skills corresponding to unsupervised subword tokenization, we use pre-trained models to label words with syntactic and morphological tags. In practice, our dataset already contains these tags for each word (477 unique) that describe syntactic features like POS and morphological features such as tense, gender, person, case, and number. However, these tags can also be generated using libraries such as SpaCy[2], which have pre-trained models available for a wide variety of languages.

[2]https://spacy.io/

Intuitively, these tags are other candidates to serve as additional skills for a given item. Specifically, we experiment with two versions: one that includes the word and these tags as skills (13,488 words combined with 477 tags for a total of 13,965 skills, an average of 4.154 skills per item), and another one that includes the lemma of the word and tags as skills (9,913 lemmas which results in a total of 10,390 skills, also an average of 4.154 skills per item).

4.3 Semantic Skills

While DAS3H, as we previously formulated, is able to model difficulty for each lexical skill as well as additional morphological and syntactic skills, it does not directly allow for interaction between the practice of distinct yet related words. However, whether practicing semantically similar words helps learners remember a word is not clear. On one hand, practicing one semantically similar word might reinforce the knowledge of similar words (i.e. constructive interference) and enhance memory strength. On the other hand, practicing words that are semantically similar may cause confusion between the different words (i.e. destructive interference) and reduce memory strength. These possible cases motivate the following approaches for incorporating skill-skill interactions within the DAS3H framework using semantic features. In this work, we consider using semantic word embeddings. We start by detailing our method using cosine similarity between semantic word embeddings to model constructive interference in Section 4.3.1. We then detail our method using a neural network to predict the skill similarity based on semantic word embeddings in Section 4.3.2, since a learnable similarity measure gives the model extra flexibility to capture both constructive and destructive interference.

4.3.1 Cosine Similarity of Word Embedding Skills. Since each *item* in the SLA setting is linked to a word, it seems plausible that we can define a similarity metric between words to express how semantically similar two words are to each other. One way to accomplish this is by introducing word embeddings to convert each word into a real-valued vector. Then, we can apply cosine similarity between each pair of words to get a matrix of values that express how similar each word is to another. Because these are real-valued vectors, the resulting cosine similarities serve as a continuous measure of item similarities, which is different from our previous characterization of discrete item similarities using lexical/morphological/syntactic skill tags. Since the resulting cosine similarity can be negative, we scale all similarity scores to the range $[0, 1]$ to ensure the log counts in DAS3H can still be used. Using this similarity matrix, we redefine how the past attempts, $a_{s,k,w}$, and past correct, $c_{s,k,w}$, for each time window $w \in W$, are calculated as

$$a_{s,k,w} = \sum_{\tau \in [t-w,t)} \psi(k, k_\tau) \tag{3}$$

$$c_{s,k,w} = \sum_{\tau \in [t-w,t)} \psi(k, k_\tau) y_{s,j(\tau),\tau} \tag{4}$$

Thus, for each other vocabulary word, k_τ, reviewed within the time window w, we retrieve a similarity score, $\psi(k, k_\tau)$, between the word at time t and the word at time τ. Specifically,

$$\psi(k, k_\tau) = \frac{\mathbf{e}^T \mathbf{e}_\tau}{||\mathbf{e}||_2 ||\mathbf{e}_\tau||_2},$$

	Skills									
Items	comimos	comiste	aprendimos	com-	aprend-	-imos	-iste	-os	-s	...
comimos	X			X		X		X	X	...
comiste		X		X			X			...
aprendimos			X		X	X		X	X	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

Table 1: An example Q-matrix with skills corresponding to subwords of each item generated from a trained subword tokenizer. "X" means that an item (the corresponding row) is tagged with a skill (the corresponding column).

where $\mathbf{e}$ and $\mathbf{e}_\tau$ are the word embeddings associated with words k and k_τ, respectively. This formulation means that all words in the learner's past practice history that have nonzero cosine similarity with the current word will be included in these counts and used to predict the current recall probability.

As specified in Equations 3 and 4, if two words k and k_τ have a high similarity score (close to 1), then practicing k_τ would have a similar effect on the learner's memory of k to practicing k itself. However, in practice, reviewing a similar word may not help quite as much as reviewing the word itself, even if the two words share a high degree of semantic similarity. Therefore, we apply a temperature parameter, γ, to the similarity matrix which effectively discounts correct and total attempt counts when contributed by semantically similar but different words. For a given temperature setting, $\gamma \in [0, 1]$, the resulting similarity matrix is given by

$$\psi_\gamma = \begin{cases} \psi(k, k_\tau) \cdot \gamma & \text{if } k \neq k_\tau \\ 1 & \text{if } k = k_\tau. \end{cases}$$

Lower temperature values discount the similarities between different words, and at $\gamma = 0$, this model reduces back to the basic word-skill model that treats each word as its own skill.

4.3.2 Neural Similarity of Word Embedding Skills. The cosine similarity metric only models constructive interference and assumes that the semantic similarity between two words, as given by their embeddings, are equivalent to the way they are related in memory, which may not be an accurate assumption. Therefore, we formulate another more flexible approach for incorporating word embeddings and capturing the semantic-memory interactions between words. In this approach, each word is associated with an embedding. Then, when we want to predict a learner's recall of a particular vocabulary word, v_t, we look at the embedding that corresponds to this word, $\mathbf{e}_t$, and the embeddings of words practiced within the past time window, $\mathbf{e}_\tau, \forall \tau \in [t - w, t)$. For each combination $(\mathbf{e}_t, \mathbf{e}_\tau)$, we concatenate these two embeddings along with a categorical representation (same session, hours, days, weeks) of the time difference $\Delta_\tau = t - \tau$ and feed the resulting vector into a simple feed-forward fully-connected neural network, $\phi(\cdot)$, to produce a learned similarity score between the two words. We apply the hyperbolic tangent (tanh) activation function to the final layer of this neural network since the interference between two different skills might be constructive (i.e., a positive similarity score between $(0, 1]$), destructive (i.e., a negative similarity score between $[-1, 0)$), or neutral (i.e., a similarity score of 0). Note that log counts are not used in this case because the sums of correct and attempt features can be negative in the case of destructive interference. Additionally, the neural network can use Δ_τ to learn to discount practice sessions that are in the distant past, so we only need to use one time window, i.e. $W = \{\infty\}$. The corresponding formulation of h_θ is given by

$$h(H_{s,j,t}) = \sum_{w \in W} \theta_{v_t, w} \sum_{\tau \in [t-w, t)} y_{s,j,t} \cdot \tanh(\phi(\mathbf{e}_t, \mathbf{e}_\tau, \Delta_\tau)) - \sum_{w \in W} \theta'_{v_t, w} \sum_{\tau \in [t-w, t)} \tanh(\phi(\mathbf{e}_t, \mathbf{e}_\tau, \Delta_\tau)).$$

5 EXPERIMENTS

5.1 Experimental Setup

For our experiments, we use the same 90%-10% (11.5M-1.3M) chronological train-test split that is used to evaluate HLR in [36]. We further split the 90% of data in the train set such that the first 80% (10.2M) is used for training and the last 10% (1.3M) is used for model development, i.e., as a validation set for parameter tuning.

We train all of our models using the Adam optimizer [17] with the cross-entropy loss. Because DAS3H and other skill-based models are normally used for binary classification of response correctness, we add a term to the loss to adapt it to our task of predicting the probability of successful recall. Similar to [36], we add the following logit loss term to our cross-entropy loss objective:

$$l(p, \hat{p}, z) = z \cdot \left(\log\left(\frac{p}{1-p}\right) - \log\left(\frac{\hat{p}}{1-\hat{p}}\right) \right)^2.$$

Here, z is a scaling factor, and p and $\hat{p}$ are the actual and predicted probabilities, respectively. Empirically, we find that this additional term significantly improves the learned model's ability to precisely predict probabilities that are between 0 and 1 rather than skewing toward either extreme. The intuition is that this term provides a stronger signal for training to predict probabilities since the probabilities themselves may be closer but they will exhibit more separation under this logit loss.

During training, we do early stopping by checking whether the loss on the validation set fails to decrease for some number of consecutive epochs. We select hyperparameters by performing grid search over a small number of hand-picked values for each parameter and selecting the hyperparameters that leads to the best area under the receiver operating characteristic curve (AUC) on the validation set.

We find that a learning rate of 0.001, no ℓ_2 regularization, and (when applicable) a logit loss weight $z = 0.01$ perform best for

most models. For subword tokenization, we use a vocabulary size of 5,000 and an n-best subword splits of $n = 2$. For the cosine similarity method we use a temperature of 1.0. For the neural similarity method, we use a learning rate of 0.005, dropout probability of 0.1, a batch size of 2,000, a logit loss weight $z = 0.01$, and $\phi(\cdot)$ is a two-layer, fully-connected neural network with 512 and 64 neurons in the first and second hidden layers, respectively. Finally, for DKT, we use a learning rate of 0.0005, dropout probability of 0.2, a batch size of 256, an RNN hidden layer size of 2,048, and a two-layer, fully-connected neural network with 1,024 and 128 neurons in the first and second hidden layers, respectively. We also report limited results where hyperparameters are optimized for the mean absolute error (MAE) metric. We find that a learning rate of 0.001, an ℓ_2 regularizaton weight of 0.05, and a logit loss weight of $z = 0.1$ works best for the MAE for the subset of models included in these results. In all experiments that use word embeddings, we use 300-dimensional fastText word embeddings [16] obtained from [14]. We choose these embeddings because we need word embeddings for all six languages in the dataset, and no context is provided for the words in our dataset, so contextual word embeddings (such as BERT [9]) are not beneficial. For all models we limit the practice history of a learner to their last 200 recall attempts so that models with precomputed practice history features and models that compute these features while training can be fairly compared.

5.2 Evaluation Metrics and Baselines

Following [36], we report AUC as our primary metric and MAE as a secondary metric. AUC measures the model's ability to distinguish between cases when a learner is more likely to successfully recall a word and cases when a learner is less likely to successfully recall a word. Since our dataset contains real-valued probabilities of successful recall, MAE measures the average absolute error between the model's predictions and the actual recall probabilities, which is a mismatch with the cross-entropy objective we are using.

Results for HLR and other baselines are from [36] (and confirmed using their code[3]). After examining the distribution of recall probabilities, seen in Figure 2, we observe that a large majority of the recall rates are 1.0, i.e., the learner successfully recalled the word in its every appearance during that session. Thus, we add a baseline for MAE, predicting the median from the train set (median = 1.0). We note that this baseline outperforms HLR in terms of MAE.

Since DKT already uses embeddings to represent items in a latent vector space, we substitute pretrained word embeddings as the item embeddings. To include the recall probability as part of the input (similar to DKT's item-answer embeddings), we include two additional features that are the number correct recalls and the number of recall attemps in the current session for each word. Additionally, we use a categorical representation (same session, $<$ 10 minutes, $<$ 1 hour, $<$ 1 day, $\geq$ 1 day) of the time elapsed since the last training session as input to the LSTM. For the feed-forward neural network used for prediction, we concatenate the current item embedding with the hidden knowledge state computed by the LSTM.

[3]https://github.com/duolingo/halflife-regression

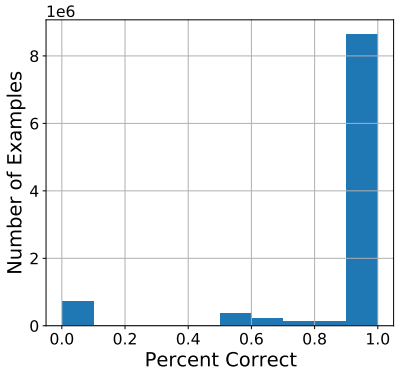


Figure 2: Histogram of recall rates for all examples.

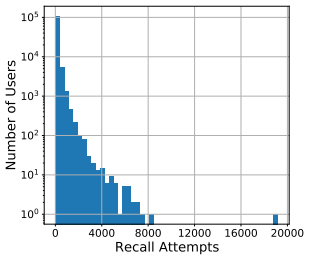
5.3 Results and Discussion

Tables 2 and 3 show the results of our experiments when optimizing for AUC and MAE respectively during grid search. Our results are grouped into six different "blocks". **Baselines** shows results from the HLR paper [36] and the constant median baseline that we add for MAE. **Learner/Item** shows results for logistic regression using just the one-hot encoded learner and item (LI) identifiers. We also show results when we add the one-hot encoded native-target language pair (+ Lang). **Lexical** shows results for variants of DAS3H that use only words (DAS3H) or lemmatized words (DAS3H Lemma) as skills. **Morphological/Syntactic** shows results for models that build on either the word or lemma variant of DAS3H with additional skills derived using subword tokenization (Subword) or syntactic/morphological tagging (Tags). **Semantic** shows results for models using the cosine similarity and neural approaches to calculate word similarities from their embeddings. As the neural approach uses only one time window, we include DAS3H with one time window (DAS3H Infinity) in this section for comparison. Finally, **DKT** shows results for DKT with pretrained word embeddings as item embeddings; we observed no performance gains by making these embeddings learnable after initializing them either randomly or with pretrained word embeddings.

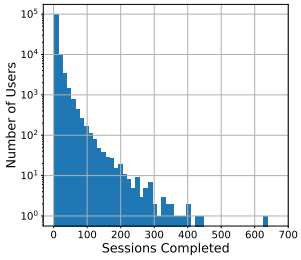
Starting with the block of lexical features, we find that the use of either words or lemmas as skills within the DAS3H framework outperforms each of the memory model baselines on both AUC and MAE. As we see in the Learner/Item block, much of this performance can be captured using only the learner and item factors of DAS3H, perhaps because the dataset spans just two weeks of learner interactions. In fact, because of this limited data collection window, the average number of recall attempts and sessions completed per learner is very low, as shown in Figures 3a and 3b. Still, the lexical skills and the associated learner skill practice history components of DAS3H provide a boost in performance in addition to information attached to learner and item identifiers. Between the different types of lexical features, the use of the lemma slightly outperforms the use of the inflected form of the word. This observation can be explained by the use of lemmas resulting in more items that share skills with each other, enabling DAS3H to capture more interactions between different items.

Block	Model	AUC↑	MAE↓
Baselines	HLR	0.538	0.122
	Leitner	0.542	0.235
	Pimsleur	0.510	0.445
	Constant $\bar{p} = 0.859$	n/a	0.175
	Constant $p_{median} = 1.0$	n/a	0.105
Learner/Item	LI - logit loss	0.619	0.180
	LI	0.627	0.114
	LI + Lang	0.631	0.112
Lexical	DAS3H - logit loss	0.629	0.174
	DAS3H - Learner	0.617	0.111
	DAS3H	0.635	0.112
	DAS3H + Lang	0.637	0.112
	DAS3H Lemma	0.636	0.112
	DAS3H Lemma + Lang	0.638	0.111
Morphological/Syntactic	DAS3H Subword	0.642	0.111
	DAS3H Subword + Lang	0.643	0.111
	DAS3H Word + Tags	0.649	0.110
	DAS3H Word + Tags + Lang	0.649	0.110
	DAS3H Lemma + Tags	0.649	0.110
	DAS3H Lemma + Tags + Lang	0.650	0.110
Semantic	DAS3H + Lang + Cosine Similarity ($\gamma = 1.0$) Word Embeddings	0.630	0.111
	DAS3H Infinity + Lang	0.636	0.111
	DAS3H Infinity + Lang + Neural Similarity Word Embeddings	**0.651**	0.109
DKT	DKT with word embeddings	0.592	0.110

Table 2: Experimental results on the Duolingo dataset, optimized for AUC. Best result is in bold.



(a) Distribution of recall attempts per learner



(b) Distribution of sessions per learner

Figure 3: The y-axis in both histograms is on a log scale. (a) Out of the 115222 learners in the dataset, the mean number of recall attempts per learner was 111.6 and the median was 34. (b) The mean number of sessions completed was 7.8 and the median was 3.

Building on this idea of items sharing skills, we move to the Morphological/Syntactic block, where we introduce additional skills along with word or lemma-based lexical features. Both the subword and tag-based approaches appear to provide useful additional features, as both lead to better AUC and MAE than DAS3H with lexical features alone. Morphological and Syntactic tagging, perhaps unsurprisingly, outperforms morphological skills derived from subwords. This result can be explained by the fact that syntactic and morphological tags provide key linguistic information based on expert knowledge, while subwords generated from a trained tokenizer provide much of the same information but are less reliable. However, in the case where automated tools for identifying POS and morphological information in words are unavailable, as in low-resource languages, subwords can still be helpful as they are automatically generated.

The use of lexical and morphological/syntactic skills, while effective, remains unsatifying because these features do not capture the inherent semantic relationships that exist between words. In the semantic block, we see that our first approach to incorporating semantic features using cosine similarity between word embeddings is ineffective. Figure 4 shows that varying the temperature parameter, γ, on a log scale, does not provide any benefits. These results show that the relationships between semantic word embeddings do not necessarily directly align with the relationships that practicing these words have on memorizing other words. One possible explanation is that destructive interference may happen, i.e., learners may get confused by similar words so that practicing similar words can be detrimental to their ability in memorizing the meaning of a word. However, we see that using a neural network to map semantic word embeddings to the context of memory significantly improves the model. This method not only outperforms the DAS3H methods with an infinite time window but also slightly outperforms the best morphological/syntactic model. This observation suggests that with a more flexible framework, the model can learn to associate semantic similarity between different practiced skills with either

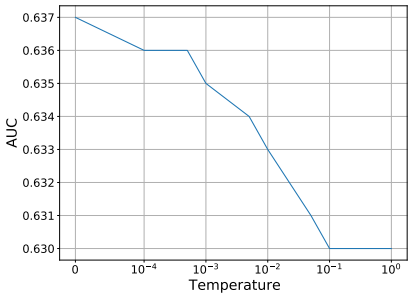


Figure 4: Performance of DAS3H with word embedding skill interactions via cosine similarity. Temperatures (γ) are varied on a log scale.

Block	Model	AUC↑	MAE↓
Baselines	HLR	0.538	0.122
	Leitner	0.542	0.235
	Pimsleur	0.510	0.445
	Constant $\bar{p} = 0.859$	n/a	0.175
	Constant $p_{median} = 1.0$	n/a	**0.105**
Learner/Item	LI - logit loss	0.619	0.180
	LI	0.556	**0.105**
	LI + Lang	0.516	**0.105**
Lexical	DAS3H - logit loss	0.629	0.174
	DAS3H	0.548	**0.105**
	DAS3H + Lang	0.513	**0.105**

Table 3: Limited experimental results on the Duolingo dataset, optimized for MAE. Best results are in bold.

constructive or destructive interference and automatically learn how to discount practices from the distant past.

We observe that the best version of the model using morphological/syntactic skills and the best version of the model using semantic word embeddings as skills perform similarly. One possible explanation is that word embeddings also capture morphological and syntactic information about the words they represent. The use of word embeddings is preferable because it does not require tagging words according to their morphological attributes. Moreover, it is possible to develop more advanced models to better leverage these semantic embeddings and more interpretable models to explain the impact of practicing semantically related words on one's ability to memorize a word, which we leave for future work.

There are a few general trends that we observe throughout Table 2. Using the logit loss in addition to the cross-entropy objective substantially improves performance on our secondary metric, MAE, while also further improving performance on the primary metric, AUC. This is likely due to the stronger signal provided by the logit loss for discriminating between cases where the true recall rates are similar. Additionally, the inclusion of the Lang feature tends to provides a slight benefit in most cases. One reason these features are useful is that they can capture how similar the learner's native language is to the language they are learning, hence making the learning process easier or more difficult. Finally, we note that no models can surpass the MAE that is achieved by simply predicting the constant median recall rate. This is in part due to the imbalance in the distribution of recall rates, as shown in Figure 2. However, we do decrease the MAE from that of HLR while simultaneously increasing the AUC. Furthermore, in Table 3, we see that if we do optimize for MAE in our grid search, we can match this baseline in terms of MAE. However, in practice this means that the model is just learning to always predict a recall probability of 1.0, due to the aforementioned skewed recall distribution. This conclusion is reinforced by the drop in AUC observed as MAE approaches that of the median baseline.

Lastly, DKT does not perform well on this dataset relative to DAS3H. This may be due to a combination of factors, such as the skewed recall distribution and the relatively short practice histories of most learners (Figures 3a and 3b). Moreover, in their paper, Piech et al. [30] indicate that performance decreases when item and answer are embeddings separated as opposed to using combined item-answer embeddings. However, this was not easily transferable to our setting since recall probabilities are continuous rather than binary. These observations suggest that using methods such as DAS3H with elements inspired by cognitive theory such as exponential forgetting curves is more beneficial than simply applying deep learning without these elements.

6 CONCLUSIONS

In this paper, we have explored the use of skill-based models for the memory-based task of predicting probability of recalling a word given a learner's practice history. We proposed a series of methods, most based on the DAS3H model, that incorporate rich linguistic features for each word. Experimental results on a large real-world second language acquisition task showed that incorporating lexical, morphological/syntactic, and semantic features improves our ability to break down each word into a series of "skills", and thus improve predictive accuracy. There are many avenues for future work. First, we need to develop methods that explicitly capture the constructive and destructive interference effect that semantically similar words have on memory. Second, we need to understand the relationship between word similarity and time difference in a learner's practice history.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under grants DGE-1938059 and IIS-1917713.

REFERENCES

[1] T. Barnes. 2005. The Q-matrix Method: Mining Student Response Data for Knowledge. In *Proc. AAAI Workshop Educ. Data Min.* 1–8.
[2] Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known English word lemmas. *Behavior research methods* 46, 3 (2014), 904–911.
[3] M. Carrier and H. Pashler. 1992. The influence of retrieval on retention. *Memory & Cognition* 20, 6 (Nov. 1992), 633–642.
[4] Hao Cen, Kenneth Koedinger, and Brian Junker. 2006. Learning factors analysis–A general method for cognitive model evaluation and improvement. In *Proc. International Conference on Intelligent Tutoring Systems*. 164–175.
[5] Haw-Shiuan Chang, Hwai-Jung Hsu, and Kuan-Ta Chen. 2015. Modeling Exercise Relationships in E-Learning: A Unified Approach.. In *EDM*. 532–535.

[6] Benoît Choffin, Fabrice Popineau, Yolaine Bourda, and Jill-Jênn Vie. 2019. DAS3H: Modeling Student Learning and Forgetting for Optimally Scheduling Distributed Practice of Skills. *arXiv preprint arXiv:1905.06873* (2019).

[7] Youngduck Choi, Youngnam Lee, Dongmin Shin, Junghyun Cho, Seoyon Park, Seewoo Lee, Jineon Baek, Chan Bae, Byungsoo Kim, and Jaewe Heo. 2020. Ednet: A large-scale hierarchical dataset in education. In *International Conference on Artificial Intelligence in Education*. Springer, 69–73.

[8] Albert Corbett and John Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-adapted Interaction* 4, 4 (Dec. 1994), 253–278.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[10] Hermann Ebbinghaus. 2013. Memory: A contribution to experimental psychology. *Annals of neurosciences* 20, 4 (2013), 155.

[11] Mingyu Feng, Neil Heffernan, and Kenneth Koedinger. 2009. Addressing the assessment challenge with an online system that tutors as it assesses. *User Modeling and User-Adapted Interaction* 19, 3 (2009), 243–266.

[12] Aritra Ghosh, Neil Heffernan, and Andrew S. Lan. 2020. Context-Aware Attentive Knowledge Tracing. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2330–2339.

[13] Sian Gooding and Ekaterina Kochmar. 2019. Complex Word Identification as a Sequence Labelling Task. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 1148–1153.

[14] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning Word Vectors for 157 Languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

[15] Phillip J Grimaldi and Jeffrey D Karpicke. 2014. Guided retrieval practice of educational materials using automated scoring. *Journal of Educational Psychology* 106, 1 (2014), 58.

[16] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759* (2016).

[17] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[18] Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. *arXiv preprint arXiv:1804.10959* (2018).

[19] Andrew Lan, Christoph Studer, and Richard Baraniuk. 2014. Time-varying learning and content analytics via sparse factor analysis. In *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 452–461.

[20] D. P. Larsen, A. C. Butler, and H. L. Roediger III. 2009. Repeated testing improves long-term retention relative to repeated study: a randomised controlled trial. *Medical education* 43, 12 (Dec. 2009), 1174–1181.

[21] Sebastian Leitner. 1991. *So lernt man lernen:[angewandte Lernpsychologie-ein Weg zum Erfolg]*. Herder.

[22] R. Lindsey, M. Mozer, W. Huggins, and H. Pashler. 2013. Optimizing instructional policies. In *Proc. Adv. Neur. In. Process. Syst.* 2778–2786.

[23] Robert Lindsey, Jeffery Shroyer, Harold Pashler, and Michael Mozer. 2014. Improving students' long-term knowledge retention through personalized review. *Psychological Science* 25, 3 (Jan. 2014), 639–647.

[24] Frederick Lord. 1980. *Applications of Item Response Theory to Practical Testing Problems*. Erlbaum Associates.

[25] Michael C Mozer and Robert V Lindsey. 2016. Predicting and improving memory retention: Psychological theory matters in the big data era. In *Big data in cognitive science*. Psychology Press, 43–73.

[26] Shalini Pandey and George Karypis. 2019. A Self-Attentive model for Knowledge Tracing. *arXiv preprint arXiv:1907.06837* (2019).

[27] Shalini Pandey and Jaideep Srivastava. 2020. RKT: Relation-Aware Self-Attention for Knowledge Tracing. *arXiv preprint arXiv:2008.12736* (2020).

[28] Z. A. Pardos and N. T. Heffernan. 2010. Modeling individualization in a Bayesian networks implementation of knowledge tracing. In *Proc. 18th Intl. Conf. on User Modeling, Adaptation, and Personalization*. 255–266.

[29] Philip Pavlik Jr, Hao Cen, and Kenneth Koedinger. 2009. Performance factors analysis–A new alternative to knowledge tracing. In *Proc. International Conference on Artificial Intelligence in Education*. 531–538.

[30] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. In *Advances in neural information processing systems*. 505–513.

[31] Paul Pimsleur. 1967. A memory schedule. *The Modern Language Journal* 51, 2 (1967), 73–75.

[32] Georg Rasch. 1993. *Probabilistic Models for Some Intelligence and Attainment Tests*. MESA Press.

[33] M. D. Reckase. 2009. *Multidimensional Item Response Theory*. Springer.

[34] Siddharth Reddy, Igor Labutov, Siddhartha Banerjee, and Thorsten Joachims. 2016. Unbounded human learning: Optimal scheduling for spaced repetition. In *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1815–1824.

[35] Burr Settles. 2017. Replication Data for: A Trainable Spaced Repetition Model for Language Learning. https://doi.org/10.7910/DVN/N8XJME

[36] 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)*. Association for Computational Linguistics, Berlin, Germany, 1848–1858. https://doi.org/10.18653/v1/P16-1174

[37] J. Stamper, A. Niculescu-Mizil, S. Ritter, G. Gordon, and K. Koedinger. 2016. Algebra I 2005-2006 and Bridge to Algebra 2006-2007: Development data sets from KDDCup 2010 Educational Data Mining Challenge. online: http://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp.

[38] Behzad Tabibian, Utkarsh Upadhyay, Abir De, Ali Zarezade, Bernhard Schölkopf, and Manuel Gomez-Rodriguez. 2019. Enhancing human learning via spaced repetition optimization. *Proceedings of the National Academy of Sciences* 116, 10 (2019), 3988–3993. https://doi.org/10.1073/pnas.1815156116 arXiv:https://www.pnas.org/content/116/10/3988.full.pdf

[39] UNESCO. 2020. Education: From disruption to recovery. online: https://en.unesco.org/covid19/educationresponse.

[40] Utkarsh Upadhyay, Abir De, and Manuel Gomez-Rodrizuez. 2018. Deep Reinforcement Learning of Marked Temporal Point Processes. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18)*. Curran Associates Inc., Red Hook, NY, USA, 3172–3182.

[41] Walter JB Van Heuven, Pawel Mandera, Emmanuel Keuleers, and Marc Brysbaert. 2014. SUBTLEX-UK: A new and improved word frequency database for British English. *Quarterly journal of experimental psychology* 67, 6 (2014), 1176–1190.

[42] Jill-Jênn Vie and Hisashi Kashima. 2019. Knowledge tracing machines: Factorization machines for knowledge tracing. In *Proc. AAAI Conference on Artificial Intelligence*, Vol. 33. 750–757.

[43] Zichao Wang, Angus Lamb, Evgeny Saveliev, Pashmina Cameron, Yordan Zaykov, José Miguel Hernández-Lobato, Richard E Turner, Richard G Baraniuk, Craig Barton, Simon Peyton Jones, et al. 2020. Diagnostic Questions: The NeurIPS 2020 Education Challenge. *arXiv preprint arXiv:2007.12061* (2020).

[44] Yang Yang, Jian Shen, Yanru Qu, Yunfei Liu, Kerong Wang, Yaoming Zhu, Weinan Zhang, and Yong Yu. 2020. GIKT: A Graph-based Interaction Model for Knowledge Tracing. In *Proc. Joint European Conference on Machine Learning and Knowledge Discovery in Databases*.

[45] Zhengyu Yang, Jian Shen, Yunfei Liu, Yang Yang, Weinan Zhang, and Yong Yu. 2020. TADS: Learning Time-Aware Scheduling Policy with Dyna-Style Planning for Spaced Repetition. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1917–1920.

[46] Ahmed Zaidi, Andrew Caines, Russell Moore, Paula Buttery, and Andrew Rice. 2020. Adaptive Forgetting Curves for Spaced Repetition Language Learning. *arXiv preprint arXiv:2004.11327* (2020).

[47] Jiani Zhang, Xingjian Shi, Irwin King, and Dit-Yan Yeung. 2017. Dynamic key-value memory networks for knowledge tracing. In *Proc. International Conference on World Wide Web*. 765–774.