EBERHARD KARLS UNIVERSITÄT
TÜBINGEN

BACHELOR'S THESIS

Application of Personalized Adaptive Learner Modeling to Vocabulary Learning Software

Author:
Christopher DILLEY

Supervisor:
Prof. Dr. Detmar MEURERS

A thesis submitted in fulfillment of the requirements for the degree of B.A. Computational Linguistics

in the

Department of Computational Linguistics

Eberhard Karls Universität Tübingen

Abstract

Seminar für Sprachwissenschaft

Department of Computational Linguistics

B.A. Computational Linguistics

Application of Personalized Adaptive Learner Modeling to Vocabulary Learning Software

by Christopher DILLEY

Learning a language presents a unique challenge for many individuals. There are multiple aspects to this language learning process, but vocabulary acquisition in particular can be directly tied to discrete fact learning. Numerous studies over the last century have demonstrated several effects that greatly improve a person's acquisition and retention of discrete facts, such as the spacing, recency, and testing effects. Further studies in recent years have attempted to accurately model how these effects can be best used to adaptively optimize an individual's rate of learning facts, often applying these models to the learning of vocabulary pairs. However, these studies do not employ other known methodologies for improving vocabulary learning in individuals. In this thesis, an Android application is described which combines these fact learning models with other proven techniques for language learning, and how it may be scaled up for real-world usage.

Contents

Abstract i

1 Background 1
- 1.1 Second language acquisition . . . 1
- 1.2 The role of technology . . . 3
- 1.3 Discrete fact learning . . . 4
- 1.4 Context-based vocabulary training . . . 7
- 1.5 Summary . . . 8

2 Summary of Goals 9

3 The Learner Model 12
- 3.1 Description of the system . . . 12
- 3.2 Extensions to the model . . . 15
- 3.3 Modeling across sessions . . . 19
- 3.4 Current implementation . . . 20
- 3.5 Summary . . . 22

4 Presentation of Material 23
- 4.1 Training exercises . . . 23
- 4.2 Testing exercises . . . 24
- 4.3 Integration with the model . . . 25

5 The Android Application and Backend 26
- 5.1 The learning interface . . . 26
- 5.2 The local datastore . . . 27
- 5.3 The remote database . . . 28
- 5.4 Coordinating the components . . . 30

6 Evaluation 32
- 6.1 The simulations . . . 32
- 6.2 Simulation results . . . 33

7 Prospective Larger-scale Studies **36**
7.1 Testing the model 36
7.2 Testing the contextual presentations 38
7.3 Testing the entire system 39

8 Conclusion **40**
8.1 Discussion 40
8.2 Optimization through statistical learning 41
8.3 Final thoughts 43

A Results from Simulations **44**

B Relevant slices of code **46**
B.1 Activation calculation 46
B.2 Selecting next word 47

C Mathematical Formulas **49**

Bibliography **51**

Chapter 1

Background

1.1 Second language acquisition

In a world of increasing globalization, a greater number people are learning second languages every day. It is estimated that there will be 2 billion people learning English as a second language alone by the year 2020 (British Council, 2013). Despite this high demand, there is little consensus on how best to approach the teaching and learning of languages for optimal outcomes. Research into this and related fields is ongoing, and although it is clear that no definitive solution has been found, recent advancements have made progress towards this end.

When speaking of language learning, distinctions are often made between a person's **L1** and **L2** languages. An individual's **L1** refers to their native language, of which they are assumed to have essentially 'perfect' knowledge. **L2** may then refer to a 'second' language of the individual, but in principle represents any language being learned that is not their native language. Native languages are learned while very young, and coincide with learning about the world in general. In contrast, second languages are acquired later on, and typically learned by mapping it onto the individual's native language. Second language learning is a challenging endeavor, and despite best efforts, they are rarely learned to the same proficiency as native languages.

A key component of language learning is the acquisition of vocabulary items. In order to be able to functionally communicate in a language, a considerably large number of words must be known, along with their particular meanings and usages. However, vocabulary is often not well taught in most learning contexts, despite being crucial to the process (Oxford and Scarcella,

1994). This signals that room for improvement may be found, and doing so will increase a learner's ability to acquire a language.

According to Knoop (2012), learning a vocabulary item in a second language requires several different aspects:

- Knowledge of the form, both written and spoken.
- Knowledge of correct grammatical variations.
- The word's use in collocations or phrases, where they may not be interchangeable with synonyms.
- Knowledge of the frequency of the word, and in which situations it is appropriate to use.
- The meaning of the word, including particular nuances in meaning in various contexts.
- Ability to understand the word when heard/read, and ability to produce through speech/writing.

Mastery of a vocabulary item is dependent on mastery of each of these aspects. Accomplishing this may be approached in many different ways. A study by Gu and Johnson (1996) explored the reported study techniques of 850 language-learning students, and investigated which activities correlated most positively or negatively with test results over time. Strategies that involved extensive reading, guessing words from context combined with smart usage of dictionaries and oral repetition proved to be effective, whereas strategies involving rote memorization of vocabulary and visual repetition showed negative links.

One of the greatest positive predictors of success, however, was 'self-initiation', or the student's motivation to learn and improve on their own. Thus, providing more opportunities for well-motivated students to continue their studying beyond the classroom setting ought to be essential. Furthermore, developing systems that are accessible and engaging may encourage learners to continue improving themselves and increase their self-initiative.

1.2 The role of technology

Computer-Assisted Language Learning (CALL) systems offer a unique opportunity to supplement language learning. Students who are more heavily motivated to learn are able to seek out such systems to further engage with the language in a productive manner. These systems may also be deployed in classrooms to aid the reinforcement of language material (Nerbonne, 2002). Intelligent CALL systems (ICALL) can be made to adapt to each individual student's learning ability and goals (Segler, Pain, and Sorace, 2002), and may also employ natural language processing tools to guide particular activities and to detect systematic errors (Amaral and Meurers, 2011).

ICALL systems can pinpoint areas that need focus on a student-by-student basis, and tailor training and exercises to suit. Adapting to individual differences in students is key to optimizing the learning of material, and doing so greatly improves learning outcomes (Atkinson and Paulson, 1972). For this, systems may build an internal representation, or model, of the student. This model would attempt to describe the state of the student's learning progress based on their interactions with the ICALL system. Self (1990) argues that building such a model is essential for effective learning systems.

To apply ICALL systems more narrowly, they may be used to assist in the vocabulary learning process (Grace, 1998; Segler, Pain, and Sorace, 2002). Given a general lack of focus on vocabulary and the large amount of time required to study it, students may find a significant benefit in using ICALL systems to shore up any potential shortcomings of classroom learning. Words that a student struggles with can be automatically identified, and exercises can be designed to specifically target these words. Similarly, words that a student knows well may also be identified, and these may be presented less frequently, in favor of focusing on the student's weaknesses.

There are very many examples of ICALL systems, both in academia (C.-M. Chen, Lee, and Y.-H. Chen, 2005; Graesser et al., 2001; Amaral and Meurers, 2011; Knoop and Wilske, 2013) and in the private sector (Duolingo, Phase6, Babbel). These systems have shown promising results, and with the rise of language learners worldwide, the demand for them is only increasing. It then follows that rigorous research into the optimal strategies to employ for language learning would be tremendously beneficial.

1.3 Discrete fact learning

In psychology, memory is often described as having two distinct components: declarative and procedural memory (Ullman, 2004). Procedural memory is associated with the learning of skills and habits, such as playing the guitar or riding a bicycle. On the other hand, declarative memory is associated with the storing and recall of discrete facts and events, such as memorizing the capital of a country or a friend's birthday. The latter, which is most relevant for acquisition of vocabulary, benefits from the *spacing effect*.

The *spacing effect* (also referred to as the *distributed practice effect*) is a well-established learning phenomenon, first described over a century ago by Ebbinghaus (1885). Ebbinghaus observed that retention of discrete facts was greater if the studying of these facts was spread out over a longer period of time, with more time between studying sessions. These findings have been replicated many times since then, being observed in many different fields, including mathematics (Rohrer and Taylor, 2006), marketing (Janiszewski, Noel, and Sawyer, 2003), and, of course, vocabulary learning (H. P. Bahrick, L. E. Bahrick, et al., 1993; Bloom and Shuell, 1981).

Specifically, the spacing of learning sessions appears to increase retention of the facts considerably, even if it takes longer for the facts to be initially acquired (H. P. Bahrick, L. E. Bahrick, et al., 1993). This means that individuals who mass study material over a period of a few days will have a short-term advantage in recalling the material, but what was learned is quickly forgotten afterwards. If the same amount of time is spent studying, but the sessions take place further apart, long-term retention of the facts will improve significantly.

Not only does this seem to hold across time spans of days, weeks or months, it holds within individual studying sessions as well (R. V. Lindsey et al., 2014). This means that when presenting particular items through the course of a studying session, it benefits the learner in both the short- and long-term to space the encounters throughout the session.

Psychologists disagree about the cause of this effect. One theory attributes the effect to *diminished processing* (Hintzman et al., 1975; Greeno, 1970), claiming that the amount of information gained decreases the longer an individual studies. Another theory suggests that memory benefits from *variable encoding* (Glenberg, 1979; Melton, 1970), claiming that encountering a fact in different

contexts improves the encoding of the fact into memory. Bjork and Allen (1970) compared these effects side by side, and came to the latter conclusion, with a more recent study by H. P. Bahrick and Hall (2005) agreeing as well.

This comports with the theory of discrimination learning, which describes facts as being learned by associating features (context cues) with labels (facts) (Ramscar et al., 2010). When a fact is encountered in a particular context, all of the different components of that context serve as predictors of the fact. If any of these components later occur without the corresponding fact, their relationship in memory becomes weakened. This means that experiencing a fact in differing environments over time allows the elimination of redundant or irrelevant cues, allowing a strengthening of the correct cue-fact relationship.

In addition to the spacing effect, another learning phenomenon, the *testing effect*, can also be explained by this theory. The *testing effect* shows that testing an individual on their knowledge and recollection of a fact is more beneficial to the learning and retention of that fact than simply studying it (Carrier and Pashler, 1992). This seems sensible, as forcing an individual to make predictions given a set of cues allows for the opportunity that an incorrect answer may be given, and this failure triggers significant cue discrimination and a heightened degree of learning (Ramscar et al., 2010).

Competing with the desire to space repetitions is the need to re-study material before it is completely forgotten; too much time between study sessions has a significant negative impact on fact retention (Cepeda et al., 2009). Thus, it becomes important to strike a balance between maximizing the spacing effect through having a greater amount of time between studying, with the *recency effect* (Anderson and Lebiere, 2014), which demands less time between studying to increase likelihood of the item being recalled. Such a balance has been demonstrated successfully in several studies (R. Lindsey et al., 2009; Van Rijn, Van Maanen, and Van Woudenberg, 2009; Pavlik and Anderson, 2005).

Key to this is having an understanding of an individual's rate of forgetting, so that learning material may be presented at the most optimal time. Every individual has their own forgetting rates which can also vary depending on particular subject matter, which means many parameters have to be learned to fit the individual. However, these parameters have been shown to be remarkably constant over time (Sense et al., 2016), and so repeated trials should allow a system to converge on the correct values for those parameters.

In order to do this, a model of an individual's learning and forgetting over time must be built. Ebbinghaus (1885) proposed the idea of the *forgetting curve*, which suggests that the memory decays as an exponential function over time; the likelihood of recall decreases rapidly immediately following a learning encounter, but tapers off over time. This concept was expanded on by Settles and Meeder (2016), who describe the model being deployed as part of Duolingo's learner modeling. Another model which cites inspiration from Glenberg (1979) and the components-level (variable encoding) theory, the SAM model, is described by Raaijmakers (2003).

The model used by this paper is built upon the declarative fact model described in Pavlik and Anderson (2005), which is based on earlier work by Anderson and Schooler (1991) and derived from the ACT-R (Adaptive Character of Thought - Rational) cognitive modeling architecture (Anderson, Bothell, et al., 2004). This model describes the *activation* of a particular declarative fact item in memory, which is affected by a variable rate of decay.

The *'activation'* of an item is the measure of how well the item is encoded in memory; a greater activation value of an item for an individual indicates a greater likelihood of recalling the item. This value is determined as the sum total of all previous encounters with the item modified by the time since each encounter, and the item's rate of activation decay. The decay rate can vary depending on the spacing between encounters, in order to appropriately model the spacing effect.

This model has been used rather successfully (Pavlik and Anderson, 2005; Pavlik and Anderson, 2008), and several studies have expanded upon the model further (Van Rijn, Van Maanen, and Van Woudenberg, 2009; Van Thiel, 2010; Koelewijn, 2010; Sense et al., 2015). One of the major improvements added involves measuring the reaction time of learners in their responses to the test stimuli, and using this measurement as an indication of the learner's level of activation. This allows the model to more quickly and accurately adapt to the learner, resulting in an improved learning experience.

More details on the ACT-R model can be found in chapter 3.

1.4 Context-based vocabulary training

These discrete fact learning models have been successful in optimizing the learning of vocabulary pairs; however, they fail to incorporate other aspects that positively influence and reinforce language learning. Notably, the aforementioned studies using the ACT-R system simply present the L2 words paired with their L1 translations. Although this well reflects an individual's ability to learn fact items, language and vocabulary exist in a much richer ecosystem than this structure would suggest. The context of a word plays an important role in the meaning and usage of many words. Providing the words in context benefits learners, greatly improving vocabulary acquisition (Oxford and Scarcella, 1994; Nation and Waring, 1997; Nerbonne, 2002).

Not only is this true for native speakers of a language, it has also been shown to be very important in acquiring vocabulary in a second language (Nagy, 1995; Prince, 1996). Experiencing words in their context trains learners on their usage, but also helps encode the meaning of words in the mind of the learner. This again aligns with the variable encoding model, as the surrounding words can later act as cues which condition the word in memory (Ramscar et al., 2010).

As an example of this effect, Arnon and Ramscar (2012) show that gender agreement in a language can be used to improve acquisition and retention of nouns. The study describes training subjects to learn nouns in an artificial language that uses a gender system. Their results showed that subjects who first encountered the nouns along with their gender-inflected article performed better at recalling the meaning of the nouns than subjects who first encountered them isolated from context. This suggests that the varying inflected forms served to delineate several separate contexts for the words, and with less cue competition, it became easier to recall the words based on their associated gender cues. This phenomenon is described further by Dye et al. (2016).

Although English lacks a gender system to take advantage of in this way, evidence suggests that its use of adjectives may perform a similar role (Dye et al., 2017, under review). English has a considerably high rate of adjective usage preceding nouns, especially pairs of adjectives. In addition, the distribution of many adjectives in terms of the nouns they modify appears to follow a distribution comparable to the usage of gender in gendered languages. Therefore, presenting English nouns along with their adjectival context may

similarly improve their acquisition. It is also likely that the power of context learning in general may be attributed to this idea.

There are limitations to this approach, however. Often, it is challenging to derive the meaning of a word entirely from context (Nagy, 1995), and so it can be quite beneficial to combine the presentation of words in context along with L1 translations (Grace, 1998). Furthermore, it is also important that enough of the words in the context be understandable. Nerbonne (2002) suggests that around 95% of words presented should be already familiar to the learner.

1.5 Summary

Despite the inherent challenges language learning presents, ICALL systems provide a unique opportunity to improve and optimize the learning process. Through comprehensive modeling, presentation of material can be modified in real time to maximize the benefits of the spacing, recency, and testing effects based on an individual's specific needs. Furthermore, this learning can be reinforced through contextual presentations of words that allow the learner to better understand their usage and improve the likelihood of recalling them in the future. This thesis is focused on exactly these goals, and demonstrates an Android application utilizing these features.

Chapter 2

Summary of Goals

For this thesis, all of the aspects described in the previous chapter will be incorporated into an application for Android devices which will optimize the teaching of vocabulary items for individual users. This application is developed in collaboration with Kathrin Adlung, who provided the base system upon which the components of this thesis are built. This application contains the following features:

English vocabulary with German translations: The scope of the application will be narrow, teaching a particular set of English words targeted towards German 7th-9th grade students. The chosen vocabulary will be from the students' word lists for their respective grade levels, and will incorporate sentence contexts that use only words learned in their previous years of English instruction. This should ensure that the material to be learned is at the appropriate level, and that the context they are presented in is understandable. The material used to build this data set was provided by English teachers of this grade level.

Smart contexts: Example sentences to use as context for certain words will be chosen based on how relevant the context is for a particular word; words that appear in the context of the target word more frequently are more likely to appear as part of examples. These example sentences will be generated by the Sketch Engine (Kilgarriff et al., 2014), using their GDEX system.

All sentences will be pulled from genuine sources which represent real usage of the words. It has been shown that such sentences do a better job of providing the benefit of context than do artificially constructed sentences (Nerbonne, 2002). Each word will have multiple example sentences, in order

to vary the word's context and lead to stronger discrimination learning.

The following features are added to this application for the purpose of this thesis:

Separate training and testing exercises: Although the bulk of the word presentations will test the user on their knowledge by asking them to provide the correct word, it is also necessary to initially present the words in a training exercise. These training exercises may also be used as a fall-back in case the user consistently fails to provide the correct answer for a testing exercise.

Training exercises will contain presentations of the words in sentence context, with the word to learn being highlighted, and the German translation appearing after a delay. Testing exercises will consist of sentence contexts with a blank space in the position of the word to test, along with the German translation for the word, which appears after a short delay. The user will be required to type in their response. If incorrect, the correct answer will be displayed before moving onto the next word.

Adaptive user modeling: Every interaction with a vocabulary item will be tracked by the application, which will be used to form a detailed model of the user's activation for all items. This model will make predictions about the user's response to an exercise, and will adjust itself based on how well this prediction aligns with reality. Parameters will be modified in order to best fit the data and make more accurate predictions. This information will be used to determine which word to present at a given moment for the greatest amount of benefit to the user.

Remote data storage: Each user will have data about their word interactions entered into a remote backend database. This larger collection of data will allow for opportunities to identify patterns, and make stronger claims about non-user-specific parameters. Before users start a study session, information from this database may be queried in order to prepare the local model based on any improvements derived server-side. Interactions with this database will be kept to a minimum, and all session modeling will take place locally on the device. Word interactions will only be reported after a

session has been completed.

Successful implementation of this combination of features will produce an application well-suited to the task of vocabulary learning for the target group, and will allow a great amount of flexibility for future modifications.

Chapter 3

The Learner Model

3.1 Description of the system

One of the most important aspects of this project is to develop the system responsible for presenting vocabulary items to learn when it is most optimal to do so. The model chosen to make this determination is one first proposed by Pavlik and Anderson (2005), and expanded several times since (Pavlik and Anderson, 2008; Van Rijn, Van Maanen, and Van Woudenberg, 2009; Koelewijn, 2010; Nijboer, 2011; Sense et al., 2015). In this model, each learning item has a measure of *activation* for the user. A greater level of activation corresponds with a greater likelihood that the user recalls the item, and is at its highest immediately following an encounter with the item.

To conduct the modeling, information about each word presented to the user is recorded, along with information about the user's interaction with the presentation. This includes the item presented, the time of presentation, how many attempts were necessary for the user to get the correct answer, and the reaction time for each attempt.

For each item i, the activation m of the item at time t is calculated as follows:

$$m_i(t) = \ln\left(\sum_{j=1}^{n;t_j<t} (t-t_j)^{-d_{i,j}}\right) \tag{3.1}$$

This formula finds the difference between the current time t and the times of each of the previous encounters t_j (where $t_j < t$), scaling them by a decay rate $-d_{i,j}$, and summing these values together. Doing so creates a complete

summation of all encounters with the word, with the most recent encounters having the greatest effect on the item's activation.

The decay rate d for item i at encounter j is calculated as follows:

$$d_{i,j} = ce^{m_i(t_j)} + \alpha_i \tag{3.2}$$

The decay rate is the power of the activation of the item at encounter j, multiplied by a scaling parameter c. This is then adjusted by the item's decay intercept α_i, which represents the minimum decay value for the item. These formulas recursively call each other until there is no previous encounter j, at which point $m_i(t) = -\infty$, and α_i is used as the default decay value for the first encounter.

The α_i parameter is adjusted by the model to better fit the learner with each successive item encounter. For example, if the user answers correctly when presented with an item, that then reflects positively on their ability to recall the item. In this case, the item's α will decrease, which will cause the model to predict a lower amount of activation decay for the item, leading to fewer future presentations of the item. The reverse is also true; incorrect responses will increase the item's α, allowing the item to be presented again sooner.

What these formulas represent is the improved level of activation of an item over time with each successive encounter. Activation decays rapidly at first, but this activation loss tapers off over time. Additional encounters with the item will continue to increase activation. As the rate of decay after an encounter is greater when the item's activation is already high, spacing items further apart results in a lower rate of decay. This attribute represents the improvement in retention resulting from the spacing effect. Figure 3.1, taken from Van Woudenberg (2008), depicts how the activation of an item tracks with time over multiple encounters. The graph on the left depicts activation when the model uses this variable decay, and the graph on the right when it does not.

In this model, the optimum strategy must be to space encounters of objects as much as possible, without the activation level of the item falling below a certain threshold that indicates that the item is most likely forgotten. As the activation for the item increases with each encounter, the amount of time before decaying past this threshold becomes longer, allowing for more and

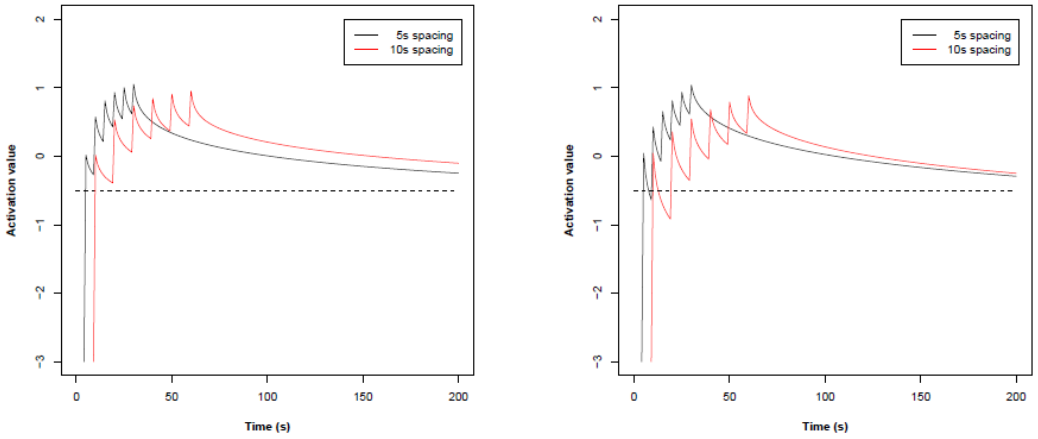


FIGURE 3.1: Activation of an item over multiple encounters, with and without variable decay. From Van Woudenberg (2008)

more spacing. Ultimately this leads to a maximal level of retention. The value of this retrieval threshold is referred to as τ.

After calculating the level of activation for all items, a decision must then be made about the order in which to present the items. Figure 3.2 depicts this decision flow. Activation for items is calculated for a time 15 seconds in the future, in order to catch items before they fall below the retrieval threshold. Items that the user has previously interacted with, but will soon fall below the activation threshold, have the highest priority to present. If no such items exist, a new item may instead be presented. If all items are above the activation threshold and there are no new items, the item with the lowest level of activation will be defaulted to. After an item is presented, the activation for all items is then recalculated. This process is repeated until the learning session is over.

The activation level of items may also be used to determine the likelihood that the user recalls the item when presented. This probability of recall p_r given a particular level of activation m_i is calculated as follows:

$$p_r(m_i) = \frac{1}{1 + e^{\frac{\tau - m_i}{s}}} \tag{3.3}$$

In this formula, the parameter s serves to smooth out any noise in the level of activation. A smaller value for s will cause the transition from 0% to 100% likelihood of recall to be very steep. Conversely, a larger value for s will distribute the probabilities more smoothly across a sigmoidal curve.

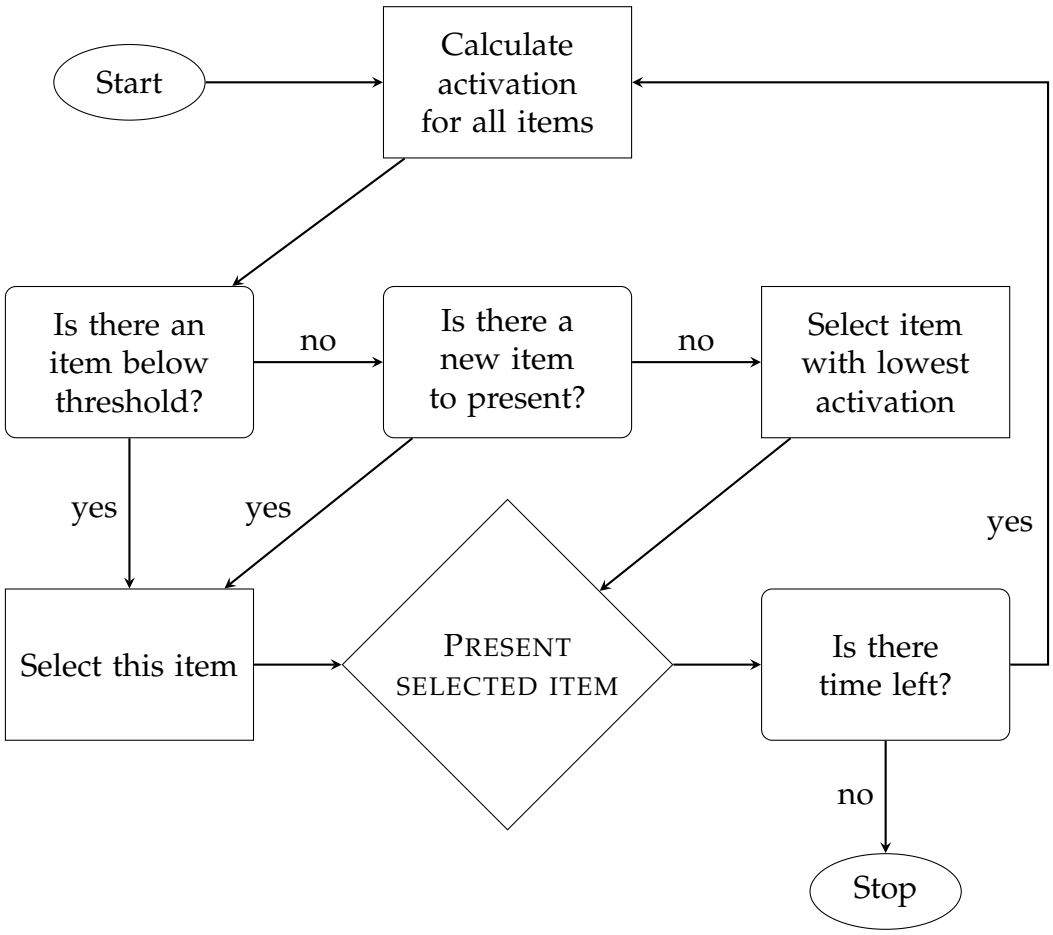


FIGURE 3.2: Decision flow for presenting the most optimal items.

This probability value may be used as a factor in adjusting the α parameter with each encounter. If the model predicts a 90% likelihood of recall, and the user answers correctly, then the adjustment of α may be small. If the model predicts 10% instead, then the adjustment of α may be larger in order to compensate for what is more likely to be an inaccuracy in prediction based on the current α value. For the purposes of this thesis, this possibility is ignored under the assumption that the probability of recall will be roughly the same for every word presentation due to the nature of the model. However, this serves as a potential avenue for future exploration.

3.2 Extensions to the model

Since the initial publication of this model, several modifications have been made to improve upon some of its shortcomings. An extension to the model was proposed by Pavlik (2007), which modifies the activation formula:

$$m_i(t) = \beta_s + \beta_i + \beta_{s,i} + \ln\left(\sum_{j=1}^{n;t_j<t} b_j(t-t_j)^{-d_{i,j}}\right) \tag{3.4}$$

This improved formula adds several parameters which can be adjusted to better fit the data and produce better predictions. The β parameters are values unique to the individual and the item: β_s represents the the individual's learning ability, β_i represents the difficulty of the item, and $\beta_{s,i}$ represents the relative difficulty of the item for the individual. These account for the fact that some items are more difficult to learn than others (β_i), some learners are more proficient than others (β_s), and that some items may be particularly more or less suited to an individual ($\beta_{s,i}$).

In addition, the scaling parameter b_j is added, which represents the effectiveness of the word presentation itself. For example, as it has been shown that testing promotes learning more effectively than simply studying (Carrier and Pashler, 1992), study-only trials may be given a smaller value of b_j when computing their resultant increases in activation.

Another significant innovation involves factoring the reaction time of the learner into the recomputation of activation values (Van Rijn, Van Maanen, and Van Woudenberg, 2009). The idea behind this being that a higher reaction latency indicates that the item is more difficult to retrieve from memory, and thus the item likely has a lower level of activation than an item that is recalled more quickly. The reaction latency is measured as the time between the item being presented and the moment the user first interacts with the application to begin inputting a response. The predicted reaction time for an item RT_i for a given encounter j, utilizing the activation of the item i at encounter j, is computed as follows:

$$RT_{i,j} = Fe^{-m_{i,j}} + f \tag{3.5}$$

In this formula, F represents a scaling parameter, and f is the base-level reaction time for an individual which can be attributed to mental and physical processes associated with providing a response beyond what is necessary to recall the item. Most studies that utilize this extension use a fixed value for f (Pavlik and Anderson, 2008; Van Rijn, Van Maanen, and Van Woudenberg, 2009; Van Thiel, 2010; Sense et al., 2015), but Koelewijn (2010) suggests obtaining a baseline reaction time measurement before conducting word trials

in order to accommodate individual differences.

In order to account for situations in which a user takes a very long time to respond, the reaction time is capped at a maximum value, preventing the α from being distorted too severely. This value is also used as the reaction time value when an incorrect response is given, regardless of how long the actual response time was. The capped reaction time $RT_{capped_{i,j}}$ is calculated as follows:

$$RT_{capped_{i,j}} = \min(RT_{max}, RT_{i,j}) \tag{3.6}$$

$$RT_{max} = Fe^{-1.5\tau} + f \tag{3.7}$$

The calculation of the maximum value is determined by the activation threshold (τ), which is scaled arbitrarily by 1.5 to accommodate some error allowance.

In its most basic form, the discrepancy between predicted and observed reaction times can be used to adjust alpha values in a particular direction by a varying amount. A formula for accomplishing this, proposed by Van Rijn, Van Maanen, and Van Woudenberg (2009), describes the change in the α value based on these reaction times:

$$\Delta\alpha = \max\left(0.01, \frac{RT_{observed} - RT_{predicted}}{1000}\right) \tag{3.8}$$

This method produced good results, but such a model has the problem of converging very slowly due to the incremental nature of its α adjustments. To improve upon this, Van Thiel (2010) proposed reverse-engineering the reaction time formula in equation 3.5 to obtain an estimated activation value for the last encounter based on the observed reaction time.

$$m_{obs}(t) = -\ln\left(\frac{RT - f}{F}\right) \tag{3.9}$$

After determining this observed activation value, we then need to find the contribution of the last encounter to the total activation for the item. We do this by separating out the last encounter from the original activation formula in equation 3.1:

$$m_{obs}(t) = \ln\left(\left(\sum_{j=1}^{n-1;t_j<t}(t-t_j)^{-d_{i,j}}\right) + (t-t_{j=n})^{-d_{i,j=n}}\right) \tag{3.10}$$

This can be rewritten in order to get the decay value for the last encounter:

$$d_{i,j=n} = -\log_{(t-t_{j=n})}\left(e^{m_{obs}(t)} - \left(\sum_{j=1}^{n-1;t_j<t}(t-t_j)^{-d_{i,j}}\right)\right) \tag{3.11}$$

Finally, this decay value can be inserted into equation 3.2, and rearranged to get the α for the last encounter:

$$\alpha_{i,j=n} = d_{i,j=n} - ce^{m_i(t_j)} \tag{3.12}$$

Now that this α_{new} for the last encounter is calculated, it serves as one bound of range to search for a new optimal α value; the value α_{old} from the previous item encounters serves as the other bound. A binary search is conducted between these two values to obtain an α that minimizes the total error of all predicted reaction times when compared to all observed reaction times for the item. In this way, the observed reaction times can allow the value of α to quickly converge on its true value that best fits the collected data and most accurately models the learner.

One complication that arises when implementing this system lies in situations where the observed activation is lower than what the model expects to be the minimum possible activation based on the decay of previous encounters. In these situations, the subtraction inside the logarithm of equation 3.11 results in a negative value, leading to complex numbers and undefined behavior. In order to prevent such circumstances, the values making up this subtraction operation are checked before implementing this formula. If they were to lead to this situation, a maximum α value of 0.5 is instead used as the α_{new} bound for the following binary search. This bypasses the problem, and leads to results that still adjust the α_i upwards as intended.

Another workaround involves forgoing this problematic equation entirely. In the SlimStampen system developed by Van Rijn (2010), α_{new} is instead calculated as $\alpha_{old} + 0.05$ if the reaction time was slower than expected, and $\alpha_{old} - 0.05$ if faster than expected. A binary search is then conducted between α_{old} and α_{new} in the same manner as previously described. This is similar

to the simple system in equation 3.8, but allowing larger adjustments and ensuring that adjustments are made to fit all existing data rather than only considering the last encounter. This system is also less susceptible to noise in the first few encounters, and does not set an arbitrary maximum value for α_i. However, this method will still converge more slowly on the correct α_i. For this reason, the Van Thiel (2010) method is used in this thesis instead.

3.3 Modeling across sessions

The model, as described so far, functions well inside the span of a single study session, but begins to fall apart when modeling the rate of forgetting between study sessions. A likely cause of this is a change in the rate of forgetting, which has been shown to slow considerably between study sessions (Anderson, Fincham, and Douglass, 1999). This phenomenon was also observed by Pavlik and Anderson (2005) when constructing this model.

In order to compensate for this, Pavlik and Anderson introduced the concept of *psychological time*. This concept suggests that for the purposes of modeling the rate of forgetting, the time spent outside study sessions can be multiplied by a *psychological time constant* (δ) and added to the previous session time:

$$t_{current} = t_{eos} + \delta t_{out} \tag{3.13}$$

In this equation, $t_{current}$ represents the total study duration to use when calculating activation decay, t_{eos} is the study time at the end of the previous sessions, and t_{out} is the amount of time spent outside of studying. This method is clearly inelegant and likely doesn't accurately represent actual mental processes. However, it has been used to fit data effectively and may serve as a useful simplification.

Applying this methodology requires an accurate value for δ, which can be difficult to determine precisely. For this reason, Nijboer (2011) suggests an alternative method for determining the level of activation at the start of a study session. For this, an activation offset value m_{os} is calculated for an item i, based on the number of days that have passed since the end of the last session t_{days}:

$$m_{os_i} = m_{f_i} - \alpha_{last_i} \log(t_{days}) \tag{3.14}$$

In this equation, m_{f_i} represents the 'future' activation of item i, and α_{last_i} represents the α value of item i at the end of the previous study session. This new activation offset can be used in the activation equation (3.1) as part of the summation of encounters:

$$m_i(t) = \ln\left(e^{m_{os_i}} + \sum_{j=1}^{n;t_j<t} (t - t_j)^{-d_{i,j}}\right) \tag{3.15}$$

This methodology is effective at capturing the effects of intersession slowing of activation decay. It also allows for the simplification of calculating activation, as the activation for previous sessions no longer needs to be calculated every time. However, this does also mean that the adjusting of α throughout a session no longer attempts to fit encounters from previous sessions.

For the purposes of this paper, the former solution will be used. There is no one value of δ that will perfectly capture all of the potential variability in the rate of forgetting for all individuals in all circumstances, but it seems to best estimate the psychological phenomenon in the simplest manner. It also seems sensible to not fuse all word encounters in a session into one value, as some nuance in the modeling becomes lost in future sessions. Furthermore, any future modifications made to parameter values would no longer be able to retroactively apply using the latter method. To this end, a baseline value of 0.025 has been chosen for δ, as suggested by Pavlik and Anderson (2005).

3.4 Current implementation

For the purposes of this application, some modifications needed to be made to the model in order to handle some of the differences in presentation.

Having a fixed reaction time value (f), such as in equation 3.5, makes less sense when the words are being presented in sentential context. As some context sentences may be longer than others, it it crucial to take into account the length of the sentence when determining this value. To accommodate

this, a means of calculating a scaled reaction time f_s was taken from Nijboer (2011):

$$f_s = \max\left(-157.9 + 19.5C_{count}, 300\right) \tag{3.16}$$

This results in a value (in milliseconds) that is scaled based on the number of characters (C_{count}) in the sentence, including spaces, with a minimum reaction time of 300ms. The constant values in this equation were obtained experimentally by Nijboer (2011). These values represent averages, as each individual reader will have different levels of reading competency and speed. This is a place where optimization for a learner could be improved, but as this is outside the scope of this thesis, this generalized formula is used.

In addition, values for all constant parameters were selected from Van Thiel (2010), Pavlik and Anderson (2005), and Nijboer (2011). These values are shown in table 3.1.

TABLE 3.1: The chosen parameter values.

Param	Value	Description
α_d	0.3	The default initial α value for all items.
α_{min}	0.1	The minimum α value.
α_{max}	0.5	The maximum α value.
c	0.21	The decay scaling factor.
τ	-0.8	The activation threshold.
F	1	The reaction time scaling factor.
f	n/a	The base reaction time is variable.
s	0.255	The recall probability noise reduction factor.
$\beta_s, \beta_i, \beta_{s,i}$	0	Item/individual parameters are not used.
b_j^{train}	0.8	Activation scaling factor for training exercises.
b_j^{test}	1	Activation scaling factor for testing exercises.
t_{LA}	15s	Look-ahead time for activation calculations.
δ	0.025	Psychological time scaling constant.

3.5 Summary

The model utilized in this thesis aims to, as accurately as possible, track the activation of all word items for a user. These activation values represent the sum total of the activation gained from all encounters, with the gain from each individual encounter decaying as a function of time. How quickly a word's activation decays varies from individual to individual, and each word has its own rate of decay. The model must make estimates of these decay rates, and will adjust its estimates based on the reaction time measured when the learner provides an answer, as well as whether or not the given answer is correct. With each encounter, the model will improve this estimate in an attempt to best fit the collected data, and will continue to successively select the word it deems most optimal to present. In this way, the model will attempt to create the best possible learning environment for the individual, maximizing all potential learning gains.

Chapter 4

Presentation of Material

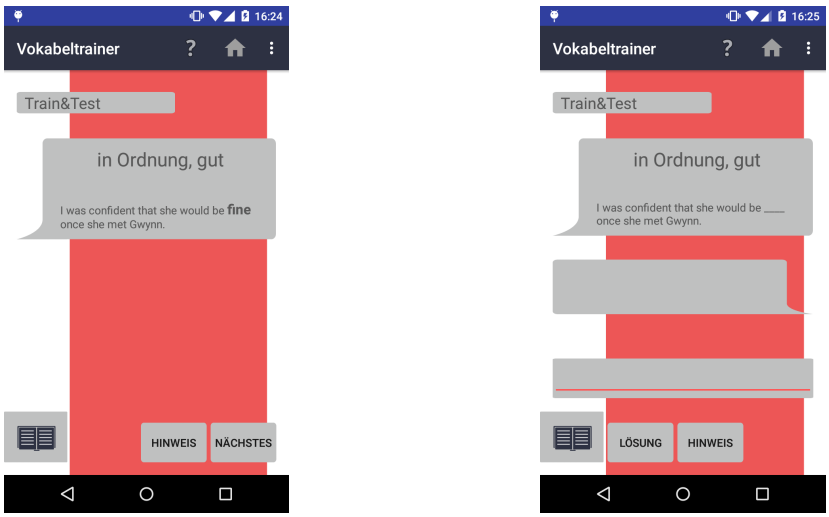
Within the framework of the application, a separate exercise activity was created for the purposes of applying the model system. This activity operates independently from other activities in the application, and so all updating and usage of the underlying model occurs only in this context. The book, chapter, and unit to draw words from may be changed at any time by pressing a button in the bottom-left of the screen. When restrictions are set in this manner, only words which fall within the defined parameters may be presented.

This activity consists of two types of exercises: **training** and **testing**.

4.1 Training exercises

Training exercises consist of a single text region in which the context sentence for a particular word is presented, with the appropriate word highlighted in bold face. After a delay, the German translation for the word appears above. The length of the delay is scaled based on the length of the context sentence. Once the German translation has appeared, the learner must press the button labeled "Nächstes" to proceed. See figure 4.1a.

This type of exercise is designed to introduce a new word to a learner in order to prepare them for future testing encounters. The word to learn is clearly emphasized, and the example sentence encourages the learner to understand the context the word finds itself in. The German translation for the word may help the learner solidify any connection they have inferred from the context, or help them establish the connection if not inferred. The translation appears only after a delay so that the context draws attention first, following proper feature-label ordering (Ramscar et al., 2010). This allows



(A) A training exercise (B) A testing exercise

FIGURE 4.1: The two activity types

the context to serve as an effective priming cue for the word in future recall attempts.

4.2 Testing exercises

Testing exercises consist of multiple text regions, including one designed to accept user input. In a presentation for the word, the context sentence is shown in a manner similar to the training exercise, except that the appropriate word is replaced with a blank rather than highlighted. The German translation again appears after delay. Feedback relating to the correctness or incorrectness of the given answer is provided in a separate text field. See figure 4.1b.

The learner must type a guess into the input field in order to continue. After submitting an answer, the activity will continue to the next word presentation after 2 seconds if a correct answer was given, or 4 seconds if not. This additional delay for incorrect responses is designed to allow the learner more time to read the feedback, helping them better understand their mistake.

Using this design, this exercise aims to encourage successful recall of a previously-encountered word. The context serves as a priming cue for the word in the user's memory, along with the German translation. The translation appears after a delay in order to give the learner an opportunity to intuit

the word purely based on context before incorporating the translation into their attempts at word recall. The translation also serves to remove any ambiguity that may arise in a context sentence where multiple words could fit the blank.

4.3 Integration with the model

Upon starting this activity, the model is initialized by loading all relevant information from the locally-stored database. The activation of all words is calculated, which allows the model to select the next word to present. After each word presentation, the measured reaction time and the result of the encounter is stored. The reaction time for an encounter is calculated as the difference in time between the moment the context sentence was presented and the moment the user first taps the input field to begin typing.

After this, all activation values are recalculated, and the next word is then selected. This continues either until all words have a sufficiently high activation (> 0), or the user exits the activity.

For the purposes of factoring in psychological time (see section 3.3), a *learning session* is started when the activity starts, and ends when the activity is closed. Any time between these sessions is considered psychological time, and the activation decay of all items will be scaled accordingly. These session time spans are stored locally and used for all future activation calculations.

Chapter 5

The Android Application and Backend

The Android application developed for this thesis consists of three primary components: the *learning interface*, the *local datastore*, and the *remote database*. These components combine to create a learning experience dictated by the model (chapter 3) and the exercise activity (chapter 4), while ensuring model calculations are performed efficiently and can be improved with time.

5.1 The learning interface

The *learning interface* refers to all of the front-facing aspects of the application that the user interacts with. This includes the menus and other UI elements designed to help the user navigate the app, as well as the exercise activities themselves. As it is a priority to keep the user engaged, it is important to have an interface that is appealing and intuitive to operate. To this end, a consistent style and theme was chosen for the application which keeps elements as clear and simple as possible.

Another important consideration lies in keeping all interactions with the application smooth and responsive. This is especially important in the exercise activity which utilizes the learning model, as some of the model's operations take a non-trivial amount of time to complete on a mobile device. For this reason, it was necessary to utilize Android's multi-threading capabilities for all calculations, database accessing, and network communication. What this means is that processing and drawing of UI components on the screen is done in parallel with any of these model operations, ensuring that the screen

rendering does not wait on the calculation to finish. This keeps the usage of the app feeling smooth, even when it is doing processor-intensive work.

For the exercise activity, the material needed to be presented clearly and unambiguously, allowing the user to focus on learning rather than the operation of the app. The design of this activity is described in more detail in chapter 4.

5.2 The local datastore

In order to perform the exercise activities, a significant amount of data needs to be stored on the device. This includes all of the vocabulary words and their associated data, all of the example sentences, and records of all word presentations for the user. All of this data must be loaded whenever the model is initialized, and then saved once the activity is over.

This data storage was accomplished using SQLite, a database system that Android supports natively. In this database, 4 separate tables of data were stored:

- **Word data:** Information about all words in the app's dictionary, including the word itself, any lemmas, references to all associated example sentences, the word's translation, and the book, chapter, and unit of the source book that the word belongs to. In addition, this includes information about the word as it relates to the model for the user, such as the word's β, α, and activation values.
- **Sentence data:** Information about all sentences used by the app, including the sentence itself and a tagged list of all words in the sentence, allowing for quick access and replacement of words.
- **Session data:** Information about all study sessions for the user, indicating the starting and ending times for all sessions (see section 4.3).
- **Interaction data:** Information about all interactions with word presentations, including the word and sentence used, the time of the presentation, the reaction time and result of the interaction, and the session during which the interaction took place.

In addition, other information needed to be stored in files outside of the database for recording and recalling. Any time the user selects a new book,

chapter, or unit, the selection is stored so that this choice may persist between different components of the application, as well as between sessions of usage. In addition, all parameters used by the learning model (table 3.1) are stored in the same manner rather than being hard-coded into the app, allowing these values to be updated easily. Finally, the application also stores information that has been generated to identify a particular user when communicating with the remote server.

With all of this data stored locally, the application and the underlying model can operate without any need to communicate with the server. This information can be accessed and updated easily, and allows for the application to operate in a robust manner.

5.3 The remote database

A key feature of this application is its connection to a remote database which stores all model and user information for all users on a single server. This means that any user with an internet connection can communicate with this server to store any interaction data they generate, as well as receive any modifications to the dictionary and the model that have been generated server-side. All server communication happens entirely in the background without any need for user interaction. However, this server communication is entirely optional; the app will continue to function without any issue if the device never has connection to the internet, simply forgoing any benefits that the server may provide.

The first major benefit of this feature is the ability to consolidate all data points from all users into a single database. This allows one to perform large-scale data analysis that could lead to incremental improvements in the design of the exercises and the models. More specifically, the individual parameters used by the model can be adjusted to fit the observed results collected from all users. A larger dataset enables more complex computations, fitting more parameters to the data with a greater degree of accuracy and confidence than would be possible with a single user's data. This idea is explored further in section 8.2.

Secondly, this server communication allows the server to update certain information and push that data to users without any need for the app itself to be updated or changed. This provides a great deal of flexibility, both when

it comes to keeping dictionary data up-to-date, as well as in distributing updates to the model parameters when necessary. Furthermore, more challenging computations can be performed on the server, with just the results being sent to the users. For example, calculating values for β_s and $\beta_{s,i}$ (equation 3.4) may be difficult to do, and the means for doing so may change. The ability to do this work from a central location simplifies the role of the application as a user tool.

To fulfill this server role, some server space was provided which hosted a database, as well as code that allowed the application to communicate with the server. The database was designed to store information similar to the application's local database, but with additional modifications for keeping track of individual users.

The tracking of individual users is an important aspect of the database system. Each instance of the application is designed to create a new, anonymous user profile the first time it communicates with the server. All generated user profiles are given a unique identifier as well as a unique session token to use for authentication, and these values are communicated to the app for storage and usage. This provides a layer of security that prevents users from accessing any information that does not belong to them.

Interactions and study sessions from all users are stored in single table that can be sorted and queried by the user that submitted them. In addition, data that relates to words for specific users is stored in a similar manner, and kept separate from the general dictionary data. All dictionary information and values for model parameters are stored in publicly accessible areas of the database, and do not require user credentials to access.

The communication between the application and the server is kept to a minimum, and maintains a simple flow of operations. Whenever the app's exercise activity is started, the app will attempt to acquire all database IDs for data elements that are missing in the local database, in order to remain synchronized with the remote database. If this is completed successfully, any changes to dictionary and parameter data that have been made since the last such query are sent to the app. Then, when the activity closes, any newly generated session and interaction data is pushed to the server. If any of these operations fail due to the volatility of internet communication, the app will continue on unabated, and will wait to try them again whenever a connection is reestablished.

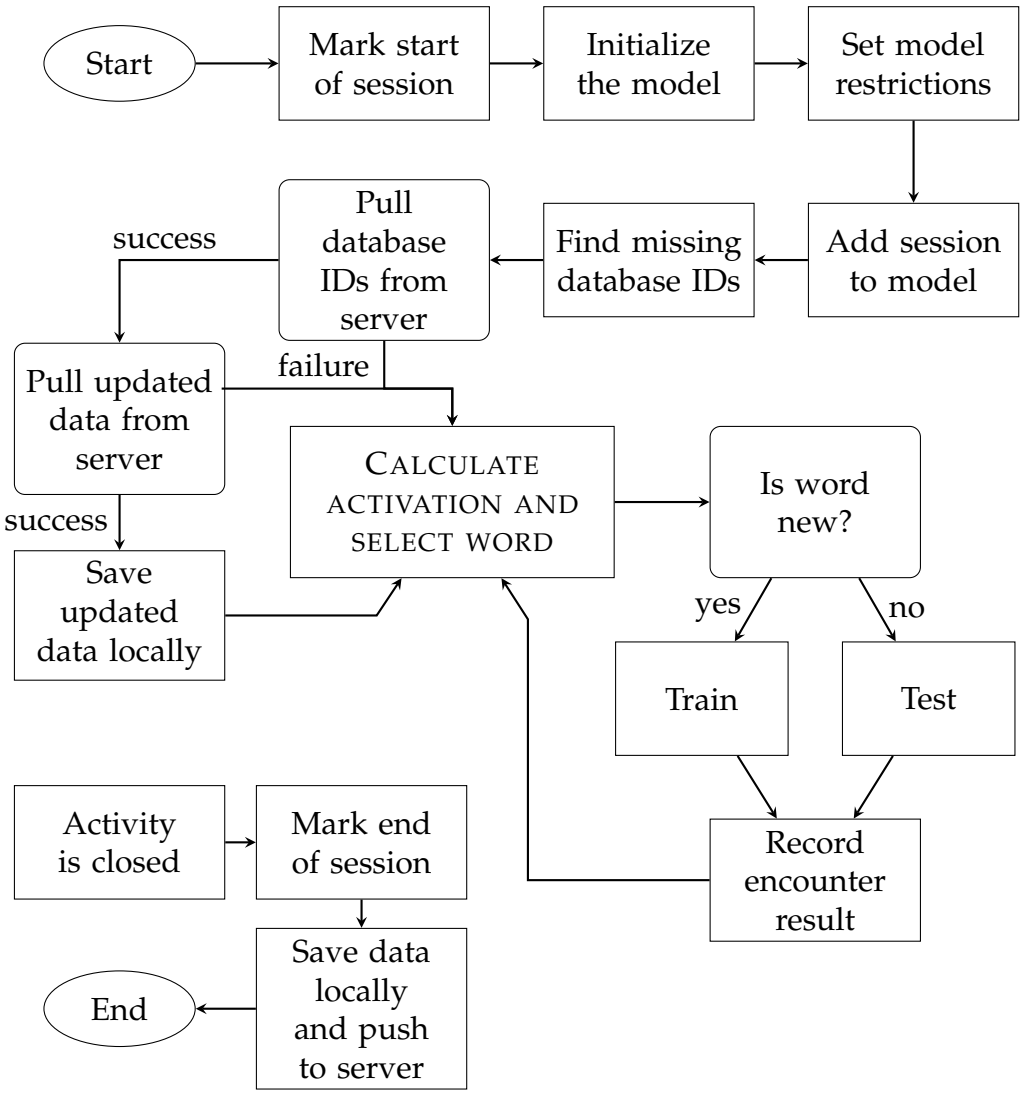


FIGURE 5.1: Process flow for the operation of the activity.

Through this system, data can be anonymously collected from users with minimal intrusion, and the user's learning experience can be improved without any user action necessary.

5.4 Coordinating the components

Putting these components together, the application creates an optimal learning experience by storing all word interactions both locally and remotely in order to best fit a model to the user. Whenever the exercise activity is started, the application follows the procedure outlined in figure 5.1. This flowchart describes the actions taken step-by-step for initializing and preparing the

model by reading the local data as well as updating data from the server, if able, before moving on to execute the training and testing exercises.

This system is designed to operate as a self-contained, independent unit without the need to communicate with the server. However, if given internet access, the user is able to provide data to the server and ultimately obtain incremental improvements to the model derived server-side. This flexibility creates an environment that suits an individual user's preferences, making it suitable for general use even outside of research conditions.

Chapter 6

Evaluation

In order to demonstrate the functionality of the system, simulations were constructed that emulate a user of the application. These simulations produced results that reflect how the system expects real users to perform.

6.1 The simulations

Within the system, a separate module was developed which served to simulate user interactions and the passage of time, which then interacted with the model as if the generated data was genuine user input. For each word in the system, a random value was selected for the decay intercept (α_i), which is the value that the model attempts to adjust and fit for a user. This is done in order to see how well the system converges on a particular α_i value over time. The random α_i values followed a Gaussian distribution, with a mean of 0.3 and a standard deviation of 0.08. With each successive word presentation, the simulation advanced an internal clock ahead by an amount which reflected the amount of time that it would take to complete a word interaction.

Whenever a new word was selected by the model, the simulation computed a 'real' value of activation using the randomly selected α_i for the word. This allowed it to pick a reaction time to simulate (equation 3.5) as well as whether or not the simulated response would be correct based on the likelihood of correctness (equation 3.3). The reaction time was also varied using a Gaussian distribution with a standard deviation of 0.3 seconds. In addition, the simulation was also set up to be able to simulate multiple sessions spread out over the course of days, in order to also model the scaled rate of forgetting in inter-session time. The simulation is only capable of communicating with the application's learning model through these simulated interactions,

in the same way that a user would, leaving the learning model to interpret the input in the same manner.

6.2 Simulation results

Two study sessions were simulated, with each session lasting one half hour. The starting times of the sessions were separated by 24 hours. Through these sessions, 37 unique words were presented over 441 total word presentations. Time between these sessions was scaled according to the psychological time constant (δ), as described in section 3.3. The model's estimate of activation and alpha over time for several selected items is depicted in figure 6.1. In these graphs, time is represented along the x-axis for both sub-graphs, with the shaded area representing scaled inter-session time.

The upper graph plots the activation of the item at each point in time, according to the model. Each spike in this line represents the moment of an encounter with the word. All testing exercises are also labeled with either an 'X' or an 'O', depending on whether the answer given in this encounter was incorrect or correct, respectively. The dashed line indicates the forgetting threshold (τ). The model aims to present words before their activation falls below this line, but they may do so nonetheless if multiple items fall below the threshold at around the same time.

The lower graph plots the estimate of the α_i value for the item at each point in time. After each encounter, the α_i value is adjusted by the model to fit the new data. This value will increase any time an incorrect answer is given or the simulation gives a slower reaction time than predicted, and it will decrease whenever the simulation gives a faster reaction time than predicted. The dashed line represents the randomized 'real' α_i chosen by the simulation. The model is designed to converge towards this line over time based on the answers and reaction times given by the simulation.

As can be seen in figure 6.1, the estimated activation for the simulations follows this pattern well. Any time an incorrect answer is simulated, this corresponds with an increase in α_i and a greater decay, resulting in the next encounter occurring sooner. Occasionally this also occurs when correct answers are given, which corresponds to answers with a slower reaction time than anticipated. Adjustments to α_i are much larger early on while the model tries to best fit the first few encounters, but these adjustments generally become

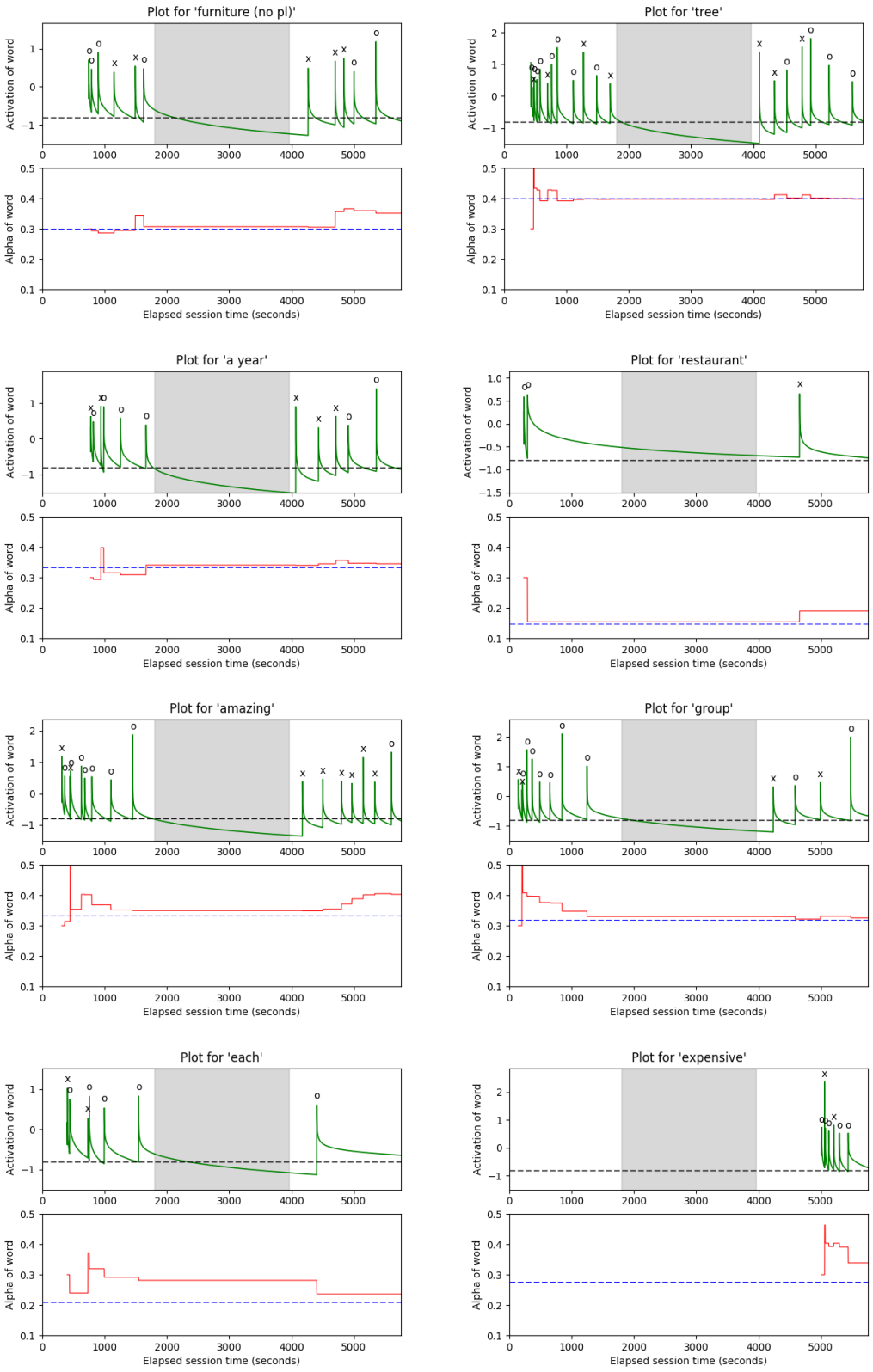


FIGURE 6.1: Simulation results

more fine with successive encounters. This suggests that although this adjustment is very susceptible to random noise in the data early on, the model seems to pierce through the noise and converge on the correct value over time, as intended. As can be seen, the model does a good job of converging on the hidden α_i values, with few exceptions (notably, for the word *'amazing'*). Due to the random nature of the simulations, such exceptions are not unexpected.

Also notable is the amount of decay that occurs during intersession time. Despite this time frame being scaled significantly ($\delta = 0.025$), most words fall well below the forgetting threshold. Only one word did not (*'restaurant'*), which was one that had a very low α_i and thus very low decay. However, falling below the threshold does not necessarily mean that the word is certainly forgotten; the threshold represents the level of activation in which the learner has a 50% chance of recall. For this reason, so many words falling below the threshold is not alarming, but the value chosen for δ is certainly a parameter to keep an eye on and adjust once real data is obtained. As a result of this phenomenon, most of the first half of the second session involves reviewing words learned in the first session in order to get them all above the threshold. Only later are new words introduced again (eg. *'expensive'*).

These simulated results line up well with anticipated results. The model behaves as expected, properly adjusting its internal representation of the learner through the limited window offered by the word presentations. In doing so, it successfully converges on the real α_i for each item, and thus accurately models their activation and decay, leading to an optimized presentation order.

A full list of words presented in the simulation and their corresponding α_i values can be found in appendix A.

Chapter 7

Prospective Larger-scale Studies

After developing a system such as the one in the application, it is important to determine whether the system outperforms control conditions, as well as how it compares to existing solutions in the same space. For this reason, three experiment designs are outlined below. Two of these pertain to the main features present in the system, namely the learning model and the contextual presentations. The final experiment then seeks to ascertain where the system improves upon, or falls short of, other vocabulary learning applications.

7.1 Testing the model

The modeling of the user to produce optimal timings of presentations is a key component of the application. This system aims to, for each presentation, select the word that would provide the greatest learning benefit at that moment. Doing this successively for all words should yield the most efficient usage of time, maximizing the knowledge gained over a fixed duration.

To show this to be the case, the system must be compared against a control condition which utilizes the same words and same interface for presentation, but does not dynamically optimize the order in which words are presented. Such a set-up is typically referred to as a "flashcard" condition, as it emulates how one might study word pairs using flashcards. In this system, words are arranged in a fixed, predetermined queue and presented in order. After a word has been presented, it is placed at the end of the queue, where it is presented again after all other words have been presented.

An alternative system to use for comparison would be the Leitner flashcard system (Leitner, 1974). This system dynamically adjusts the flashcard queue based on whether correct or incorrect answers have been given. The

queue of words is split into several 'boxes' of words. The learner starts by working through the words in the first box. Any time a correct answer is given, the word is moved into the next box. Incorrect answers are put back into the first box again. The learner continues picking flashcards from the first box until all flashcards are exhausted, at which point they move onto the next box, and the process repeats. This serves as an improvement over the standard flashcard system, as it adjusts itself based on the user's proficiency with particular words.

The experiment testing this would require 3 groups of study participants who are native speakers of German, and are of roughly equal age and English proficiency. The groups are defined as follows:

- **Group 1:** Experimental group, using the application with the full learning model.
- **Group 2:** Flashcard (control) group, using the basic flashcard algorithm.
- **Group 3:** Leitner flashcard (control) group, using the Leitner flashcard algorithm.

Each group would use the same application, with the same set of words and style of presentation. The only variance between each group would be in how the order of words is chosen.

Participants would initially be presented a written test, where they would be asked to write in the German translations for all English words to be presented. This would serve as a baseline for comparisons later. Following this, participants would conduct half hour study sessions each day for 3 consecutive days. On the 4th day, participants would be asked to complete the same written test. It may also be insightful to present the test again after one or two weeks, to test retention over time.

With this set-up, the relative effectiveness of each of the systems would be evident. Using the initial test as a baseline, the improvement of each user could be measured, and the average knowledge gain could be calculated for each group as a means of direct comparison. It may also be possible to identify if any system performed better at helping users grasp words that they initially struggled with.

7.2 Testing the contextual presentations

The other major component of the application is the presenting of words in context, with the goal of improving the encoding of words in memory and thus increasing the likelihood of future recall. Presenting words in this manner should improve the user's ability to learn the words, and result in greatest gains in knowledge for the time committed.

Testing this would be similar to the arrangement for testing the model, but in this case the control condition would remove all context sentences from the presentation. Instead, only the German translation of the word would be shown, and the user would have to enter the English translation. In the initial training exercise for a word, only the word and its German translation would be displayed.

This experiment would comprise of just two simple groups:

- **Group 1:** Experimental group, using the application with all context sentences.
- **Group 2:** Control group, using the application devoid of any context sentences.

Again, both groups would use the same application with the same words and presentation style, only varying in their usage of context sentences.

The experimental procedure would follow the same structure as outlined in section 7.1, performing an initial test, and then taking place across 3 consecutive study days followed by a testing day. Results would also be similarly analyzed, looking for relative improvement compared to the baselines obtained in the initial testing.

It would also be relevant to compare how learners perform with less context or different kinds of context. For example, instead of presenting the full context sentence along with the word, only the preceding or succeeding word may be shown. Alternatively, visual or aural cues may be used as context. Such images or sounds may be presented along with the word, relating directly to the word being presented. It also may not even be necessary for the context to be related, as simply having any sort of cue may lead to better encoding and recall of words. These are conditions which may also be added as additional groups in the experiment outlined in this section, serving as alternative control conditions.

7.3 Testing the entire system

In the event that encouraging results are obtained through the experiments comparing the system's features against control conditions, it would then become necessary to compare it against existing language learning systems. Conducting such a study would reveal the advantages and disadvantages that this application yields over similar systems. It would be especially important to consider the long-term retention of words, and so testing multiple times over a longer timespan would be beneficial to understanding the implications of each system style.

To conduct this study, a number of academic and commercial vocabulary learning tools would need to be identified. This application would need to be modified to present the same collection of words as a the compared system, and each comparison would need to be made individually. This way, the only variance between the two applications would lie in how the words are presented, rather than which words are presented. In every comparison, the experiment would likely have to be modified to accommodate various aspects of the other system to ensure they are compared fairly.

In general, the structure of each experiment would be similar to those described in sections 7.1 and 7.2, with multiple learning sessions and written tests. A greater emphasis would be put on spreading tests out over multiple weeks to monitor retention decay over time.

In this way, such experiments would show how well this application performs, revealing where it outperforms established methods and also illuminating potential areas for improvement.

Chapter 8

Conclusion

Given the increasing need for effective language learning tools, it is more important than ever to develop systems to aid language learners using the best possible methods. Intelligent language learning software provides a unique opportunity to supplement traditional classroom learning, and offers motivated learners a means of accelerating their progress. To this end, this thesis has explored possibilities in improving the acquisition of vocabulary items by optimizing the scheduling of word presentations and by providing contextual content. From this, an application was developed to demonstrate the effectiveness of the system.

8.1 Discussion

The learner model utilized by the application is designed to capture the effect of memory decay in a manner that best reflects real cognitive processes. By tracking the estimated activation of individual items for a learner, words can be presented in an order that maximizes the potential learning gain over a fixed period of time. This is done by spacing the presentation of words enough to take advantage of the spacing effect, while still presenting the words before they are most likely forgotten. In order to accommodate differences between individual learners, the model adjusts the predicted rate at which items are forgotten based on the learner's reaction time and whether or not a correct answer was given. In this way, the model becomes personalized for every learner, and will continue to improve itself over time.

In the simulations that were conducted, the model has performed this function well; the model's estimates for decay parameters quickly converged

on the random individual parameters that were generated by the simulations. These simulations interacted with the model in the same way a user would, which suggests that it would perform similarly well with real test subjects.

This evaluation assumes the underlying theory of activation accumulation and decay is correct. However, given the positive results found in other papers utilizing this model's methodology, this is not an unfair assumption. In any case, more exploration is required to confirm the effectiveness of the application's modeling system.

The application also incorporated the presentation of context sentences alongside the words to learn, and used these sentences as cues when testing the learner's recall of the words. This is done to improve understanding of the word's usage, as well as to improve the likelihood of the word being recalled in the future. The inclusion of these contexts serves as an improvement over many previous systems, as providing context has been consistently shown to improve vocabulary acquisition considerably.

8.2 Optimization through statistical learning

One area for improvement lies in the parameters chosen by the model. In the learning model, there are many parameter values that have simply been set and treated as given. However, in reality, these values are simply estimates, and may actually differ significantly from their optimal values. Due to the large number of parameters, it is exceptionally difficult to separate out the influence of each parameter from the rest. There will always be a lot of noise in the collected data as well, which confounds the problem further. For these reasons, it is a sizable challenge trying to determine the accuracy of the chosen values.

This application is designed to provide a means of tackling this problem, by providing a central source for collecting large amounts of data. This data can be collected from any individuals using the application from anywhere, without requiring a laboratory setting and all the overhead and complications that that entails. Because of this, a significantly larger amount of data may be easily collected, especially if such a system is publicly released for general use as a fully-developed language learning tool. With a sufficiently

large collection of data, statistical learning techniques could be applied to attempt to find parameters that best fit all collected data.

Statistical learning is the process of understanding data; utilizing particular inputs to predict or estimate a particular output (James et al., 2013). In particular, *supervised* learning methods allow one to take known inputs and outputs, and use them to 'train' a statistical model to identify any patterns in these input-output pairs. These patterns can be used make predictions on further data, and are continuously refined based on the error found between predicted and actual outputs.

A statistical model could therefore be constructed which incorporates all of the mathematical formulas involved with the learning system, allowing the parameters to be varied to find the best fit for all data. The process would be similar to the one used to adjust α_i parameters after word encounters (section 3.2), where the activation estimated by the measured reaction time is compared against activation predicted by the learning model. However, in this case, more parameters than simply the α_i value would be adjusted.

The most important fundamental parameters to adjust would include those involved in predicting reaction times based on activation levels (equations 3.5 and 3.16) and the decay scaling parameter c (equation 3.2). In addition, the β parameters associated with particular users and items (equation 3.4) could also be estimated by identifying patterns in how their performance differs from other users and items. By doing so, the learning model would have a more accurate representation of activation decay for all users, which would improve its ability to optimize the timing of presentations further.

The system of interaction between the application and the server is specifically built with this sort of centralized optimization in mind. As performing such calculations is time consuming, and because the data for a single individual is quite limiting, it is infeasible to perform them inside the application. Instead, these calculations could be performed on the server at regular intervals with a much larger collection of data at its disposal. Once better parameters have been fit to the data, the server is able to communicate their new values to all users of the application. This allows dynamic updating of the model for all users, including all user-specific parameters, ensuring an optimized experience.

Incorporating this statistical learning methodology would greatly improve the efficacy of the learning model over time, and is well suited for broad,

scaled-up usage of the application.

8.3 Final thoughts

Altogether, the application provides a powerful learning environment that adapts to any learner using it. This thesis has demonstrated how ideas of discrete fact learning optimization can be integrated into an effective language learning application. It is hoped that these systems might be expanded on further and developed into software usable by learners everywhere.

Appendix A

Results from Simulations

This appendix lists all words that were presented in the simulation, along with the randomly chosen α_i that was hidden from the model, the final α_i determined by the model, the number of times the word was encountered during the simulation, and the number of incorrect responses given. Words are listed in the order they were first encountered.

TABLE A.1: Words used in simulation.

Word	Real α_i	Model α_i	# of encounters	# incorrect
noodles	0.285	0.277	10	2
where	0.356	0.407	24	9
many	0.268	0.279	9	2
way	0.240	0.228	7	2
market	0.419	0.497	36	16
castle	0.350	0.388	19	6
group	0.318	0.326	14	4
restaurant	0.148	0.190	4	1
amazing	0.334	0.403	17	8
to visit	0.237	0.251	9	1
each	0.211	0.236	8	2
tree	0.400	0.398	20	7
British	0.249	0.281	9	1
adult	0.300	0.379	16	6
a day	0.282	0.276	10	3
open (from ... to ...)	0.286	0.304	10	4

TABLE A.2: Words used in simulation, continued.

Word	Real α_i	Model α_i	# of encounters	# incorrect
furniture	0.300	0.351	12	5
a year	0.335	0.345	12	5
open	0.273	0.303	9	3
free time	0.284	0.289	10	2
canal	0.450	0.496	31	14
Chinese	0.361	0.418	22	8
stall	0.302	0.324	10	4
playing field	0.193	0.239	6	1
fancy	0.391	0.389	14	5
a week	0.248	0.251	6	2
to enjoy	0.420	0.500	22	13
best	0.329	0.374	13	6
wonderful	0.325	0.400	12	5
expensive	0.277	0.339	8	2
to add (sth to sth)	0.313	0.322	6	1
boat	0.354	0.375	8	1
to join in	0.230	0.289	5	1
view	0.327	0.318	4	1
canoeing	0.340	0.347	5	1
flower	0.245	0.300	2	0
area	0.246	0.300	2	0

Appendix B

Relevant slices of code

B.1 Activation calculation

```
/**
 * Calculates the activation for the given word at the given time, based on the
 * given interactions.  Must recursively calculate activation for all previous
 * encounters.
 * The 'doLast' parameter is for when calculating the sum up to interaction n-1.
 * When false, the last interaction will be entirely ignored, while still using
 * the given time.  This is used as part of the
 * {@link ModelMath#latestAlpha(InterxObject, List, List)} calculations.  This
 * being false also signals that the extra activation parameters (betas) should
 * not be counted.
 *
 * @param time         The time to calculate activation in reference to.
 * @param word         The word to calculate activation for.
 * @param interactions List of all interactions with the word, in chronological
 *                     order.
 * @param sessions     List of all study sessions, in order to scale the time
 *                     between them.
 * @param doLast       Whether or not the last interaction should be counted
 * @param alpha        The alpha value of the word to use for calculations.
 *                     This must be passed as an argument rather than simply
 *                     reading the word object in order to allow calculating
 *                     with different alphas.
 * @return The activation of the item at the given time
 */
private static float activationRecursion(Date time, VocObject word,
                                         List<InterxObject> interactions,
                                         List<SessionObject> sessions,
                                         boolean doLast, float alpha)
{
    // Base case for recursion: if there are no previous encounters,
    // activation is -infinity.
    // Used !before() here instead of after(), so that it acts like >=
    if (interactions.size() == 0 ||
        !interactions.get(0).getTimestamp().before(time))
        return Float.NEGATIVE_INFINITY;

    // Initialize the running sum
    float runningSum = 0;
```

```
    int limit = interactions.size();

    // Iterate through all interactions before the current time
    for (int i = 0;
         i < limit && interactions.get(i).getTimestamp().before(time);
         i++)
    {
        if (!doLast && (i >= limit - 1 ||
                        !interactions.get(i + 1).getTimestamp().before(time)))
            break;

        InterxObject interx = interactions.get(i);

        // Get the values for b_j and (t - t_j)
        float exerciseScalar = getExerciseScalar(interx.getExerciseType());
        double timeDifference = effectiveTimeDifference(interx.getTimestamp(),
                                                        time, sessions);

        // If this interaction does not have a cached activation value,
        // calculate and store it
        if (!interx.hasActivation())
            interx.storeActivationValue(
                    activationRecursion(interx.getTimestamp(),
                                        word, interactions, sessions,
                                        true, alpha));

        // b_j( t - t_j )^(-d_i,j)
        float decay = decay(interx.retrieveActivation(), alpha);
        runningSum += exerciseScalar *
                      (float) Math.pow(timeDifference, -decay);
    }

    // m_i(t) = beta_s + beta_s,i + beta_i + ln( sum(...) )
    return doLast
           ? activationModifiers(word) + (float) Math.log(runningSum)
           : runningSum;
}
```

B.2 Selecting next word

```
/**
 * Allows the model to choose the next word to present based on the
 * current state of the model.
 * The given words are ignored and will not be selected.  This is to
 * allow moving onto the next word without having to wait for the
 * post-interaction computation to complete.  This argument is
 * entirely optional and may be left null.
 * This does NOT force a recalculation of activation values,
 * and should only be used after doing so for the selection to be
 * meaningful.
 *
 * @param ignoreWords Words to ignore (not select), even if they
 *                                          are optimal.  Optional.
 * @return The word most optimal to present, according to the model.
```

```
 */


@Override
public VocObject getNextWord(Collection<VocObject> ignoreWords)
{
    // Check that the model has been initialized, otherwise throw
    // runtime exception
    if (!initialized)
        throw new ModelNotInitializedException();

    VocObject min = null;
    VocObject newWord = null;

    for (VocObject word : interactions.keySet())
    {
        // If the word does not meet the requirements for
        // presentation, ignore it
        if (!meetsRestrictions(word) ||
                (ignoreWords != null &&
             ignoreWords.contains(word)))
            continue;
        // If the word has an activation greater than 0,
        // ignore it
        if (word.getActivation() != null &&
                word.getActivation() > 0)
            continue;

        // Note the first new word encountered, in case a
        // new word needs to be presented.
        if (word.isNew())
        {
            if (newWord == null)
                newWord = word;
            continue;
        }

        if (min == null ||
                word.getActivation() < min.getActivation())
            min = word;
    }

    // Selection criteria:
    // 1. Lowest word below activation threshold
    // 2. If no words below threshold, then a new word
    // 3. If there are no new words, the word with the lowest
    //      activation

    if (min == null) // Only happens if no words meet reqs
        return newWord;
    else if (min.getActivation() < ModelMath.THRESHOLD ||
            newWord == null)
        return min;
    else
        return newWord;

}
```

Appendix C

Mathematical Formulas

A list of all mathematical formulas described in chapter 3 and used in the application:

Basic activation:

$$m_i(t) = \ln\left(\sum_{j=1}^{n;t_j<t}(t-t_j)^{-d_{i,j}}\right) \tag{C.1}$$

Decay:

$$d_{i,j} = ce^{m_i(t_j)} + \alpha_i \tag{C.2}$$

Correctness probability:

$$p_r(m_i) = \frac{1}{1+e^{\frac{\tau-m_i}{s}}} \tag{C.3}$$

Activation, extended:

$$m_i(t) = \beta_s + \beta_i + \beta_{s,i} + \ln\left(\sum_{j=1}^{n;t_j<t} b_j(t-t_j)^{-d_{i,j}}\right) \tag{C.4}$$

Predicted reaction time:

$$RT_{i,j} = Fe^{-m_{i,j}} + f \tag{C.5}$$

Reaction time capping:

$$RT_{capped_{i,j}} = \min(RT_{max}, RT_{i,j}) \tag{C.6}$$

$$RT_{max} = Fe^{-1.5\tau} + f \tag{C.7}$$

Observed activation:

$$m_{obs}(t) = -\ln\left(\frac{RT - f}{F}\right) \tag{C.8}$$

Decay for last encounter:

$$d_{i,j=n} = -\log_{(t-t_{j=n})}\left(e^{m_{obs}(t)} - \left(\sum_{j=1}^{n-1;t_j<t}(t-t_j)^{-d_{i,j}}\right)\right) \tag{C.9}$$

Alpha for last encounter:

$$\alpha_{i,j=n} = d_{i,j=n} - ce^{m_i(t_j)} \tag{C.10}$$

Psychological time:

$$t_{current} = t_{eos} + \delta t_{out} \tag{C.11}$$

Reading time for sentences:

$$f_s = \max\left(-157.9 + 19.5C_{count}, 300\right) \tag{C.12}$$

Bibliography

Amaral, Luiz A and Detmar Meurers (2011). "On using intelligent computer-assisted language learning in real-life foreign language teaching and learning". In: *ReCALL* 23.01, pp. 4–24.

Anderson, John R, Daniel Bothell, et al. (2004). "An integrated theory of the mind." In: *Psychological review* 111.4, p. 1036.

Anderson, John R, Jon M Fincham, and Scott Douglass (1999). "Practice and retention: A unifying analysis". In: *JOURNAL OF EXPERIMENTAL PSYCHOLOGY LEARNING MEMORY AND COGNITION* 25, pp. 1120–1136.

Anderson, John R and Christian J Lebiere (2014). *The atomic components of thought*. Psychology Press.

Anderson, John R and Lael J Schooler (1991). "Reflections of the environment in memory". In: *Psychological science* 2.6, pp. 396–408.

Arnon, Inbal and Michael Ramscar (2012). "Granularity and the acquisition of grammatical gender: How order-of-acquisition affects what gets learned". In: *Cognition* 122.3, pp. 292–305.

Atkinson, Richard C and John A Paulson (1972). "An approach to the psychology of instruction." In: *Psychological Bulletin* 78.1, p. 49.

Bahrick, Harry P, Lorraine E Bahrick, et al. (1993). "Maintenance of foreign language vocabulary and the spacing effect". In: *Psychological Science* 4.5, pp. 316–321.

Bahrick, Harry P and Lynda K Hall (2005). "The importance of retrieval failures to long-term retention: A metacognitive explanation of the spacing effect". In: *Journal of Memory and Language* 52.4, pp. 566–577.

Bjork, Robert A and Ted W Allen (1970). "The spacing effect: Consolidation or differential encoding?" In: *Journal of Verbal Learning and Verbal Behavior* 9.5, pp. 567–572.

Bloom, Kristine C and Thomas J Shuell (1981). "Effects of massed and distributed practice on the learning and retention of second-language vocabulary". In: *The Journal of Educational Research* 74.4, pp. 245–248.

British Council (2013). "The English Effect: The impact of English, what it's worth to the UK and why it matters to the world". British Council, Manchester, England.

Carrier, Mark and Harold Pashler (1992). "The influence of retrieval on retention". In: *Memory & Cognition* 20.6, pp. 633–642.

Cepeda, Nicholas J et al. (2009). "Optimizing distributed practice: Theoretical analysis and practical implications". In: *Experimental psychology* 56.4, pp. 236–246.

Chen, Chih-Ming, Hahn-Ming Lee, and Ya-Hui Chen (2005). "Personalized e-learning system using item response theory". In: *Computers & Education* 44.3, pp. 237–255.

Dye, Melody et al. (2016). "A functional theory of gender paradigms". In: *Morphological paradigms and functions. Leiden: Brill*.

– (2017, under review). "Cute Little Puppies and Nice Cold Beers: An Information Theoretic Analysis of Prenominal Adjectives". Submitted for inclusion in the Proceedings of the 39th Annual Conference of the Cognitive Science Society. URL: http://mypage.iu.edu/~meldye/papers/2017_dyeetal_adj.pdf.

Ebbinghaus, Hermann (1885). *Über das gedächtnis: untersuchungen zur experimentellen psychologie*. Duncker & Humblot.

Glenberg, Arthur M (1979). "Component-levels theory of the effects of spacing of repetitions on recall and recognition". In: *Memory & Cognition* 7.2, pp. 95–112.

Grace, Caroline A (1998). "Retention of Word Meanings Inferred from Context and Sentence-Level Translations: Implications for the Design of Beginning-Level CALL Software". In: *The Modern Language Journal* 82.4, pp. 533–544.

Graesser, Arthur C et al. (2001). "Intelligent tutoring systems with conversational dialogue". In: *AI magazine* 22.4, p. 39.

Greeno, James G (1970). "Conservation of information-processing capacity in paired-associate memorizing". In: *Journal of Verbal Learning and Verbal Behavior* 9.5, pp. 581–586.

Gu, Yongqi and Robert Keith Johnson (1996). "Vocabulary learning strategies and language learning outcomes". In: *Language learning* 46.4, pp. 643–679.

Hintzman, Douglas L et al. (1975). "Voluntary attention and the spacing effect". In: *Memory & Cognition* 3.5, pp. 576–580.

James, Gareth et al. (2013). *An introduction to statistical learning*. Vol. 6. Springer.

Janiszewski, Chris, Hayden Noel, and Alan G Sawyer (2003). "A meta-analysis of the spacing effect in verbal learning: Implications for research on advertising repetition and consumer memory". In: *Journal of consumer research* 30.1, pp. 138–149.

Kilgarriff, Adam et al. (2014). "The Sketch Engine: ten years on". In: *Lexicography* 1.1, pp. 7–36. URL: http://www.sketchengine.co.uk/.

Knoop, Susanne (2012). "Automatic Generation of Multiple-Choice Cloze Exercises for the Android Smartphone". Bachelor Thesis. Universität Bremen.

Knoop, Susanne and Sabrina Wilske (2013). "WordGap-Automatic generation of gap-filling vocabulary exercises for mobile learning". In: *Proceedings of the second workshop on NLP for computer-assisted language learning at NODALIDA 2013; May 22-24; Oslo; Norway. NEALT Proceedings Series 17*. 086. Linköping University Electronic Press, pp. 39–47.

Koelewijn, Laurens (2010). "Optimizing fact learning gains: Using personal parameter settings to improve the learning schedule." Master's Thesis. University of Groningen.

Leitner, Sebastian (1974). *So lernt man lernen*. Herder.

Lindsey, Robert V et al. (2014). "Improving students' long-term knowledge retention through personalized review". In: *Psychological science* 25.3, pp. 639–647.

Lindsey, Robert et al. (2009). "Optimizing memory retention with cognitive models". In: *Proceedings of the ninth international conference on cognitive modeling (ICCM 2009)*. ICCM Manchester, UK, pp. 74–79.

Melton, Arthur W (1970). "The situation with respect to the spacing of repetitions and memory". In: *Journal of Verbal Learning and Verbal Behavior* 9.5, pp. 596–606.

Nagy, William E (1995). *On the role of context in first-and second-language vocabulary learning*. Tech. rep. Champaign, Ill.: University of Illinois at Urbana-Champaign, Center for the Study of Reading.

Nation, Paul and Robert Waring (1997). "Vocabulary size, text coverage and word lists". In: *Vocabulary: Description, acquisition and pedagogy* 14, pp. 6–19.

Nerbonne, John (2002). "Computer-assisted language learning and natural language processing". In: *Handbook of computational linguistics*. Citeseer.

Nijboer, Menno (2011). "Optimal fact learning: Applying presentation scheduling to realistic conditions". In: *Groningen, The Netherlands: Unpublished master's thesis, University of Groningen*.

Oxford, Rebecca L and Robin C Scarcella (1994). "Second language vocabulary learning among adults: State of the art in vocabulary instruction". In: *System* 22.2, pp. 231–243.

Pavlik, Philip I (2007). "Understanding and applying the dynamics of test practice and study practice". In: *Instructional Science* 35.5, pp. 407–441.

Pavlik, Philip I and John R Anderson (2005). "Practice and forgetting effects on vocabulary memory: An activation-based model of the spacing effect". In: *Cognitive Science* 29.4, pp. 559–586.

– (2008). "Using a model to compute the optimal schedule of practice." In: *Journal of Experimental Psychology: Applied* 14.2, p. 101.

Prince, Peter (1996). "Second language vocabulary learning: The role of context versus translations as a function of proficiency". In: *The modern language journal* 80.4, pp. 478–493.

Raaijmakers, Jeroen GW (2003). "Spacing and repetition effects in human memory: Application of the SAM model". In: *Cognitive Science* 27.3, pp. 431–452.

Ramscar, Michael et al. (2010). "The effects of feature-label-order and their implications for symbolic learning". In: *Cognitive Science* 34.6, pp. 909–957.

Rohrer, Doug and Kelli Taylor (2006). "The effects of overlearning and distributed practise on the retention of mathematics knowledge". In: *Applied Cognitive Psychology* 20.9, pp. 1209–1224.

Segler, Thomas M, Helen Pain, and Antonella Sorace (2002). "Second language vocabulary acquisition and learning strategies in ICALL environments". In: *Computer Assisted Language Learning* 15.4, pp. 409–422.

Self, John A (1990). "Bypassing the intractable problem of student modelling". In: *Intelligent tutoring systems: At the crossroads of artificial intelligence and education* 41, pp. 1–26.

Sense, Florian et al. (2015). "Stability of Individual Parameters in a Model of Optimal Fact Learning".

– (2016). "An Individual's Rate of Forgetting Is Stable Over Time but Differs Across Materials". In: *Topics in cognitive science* 8.1, pp. 305–321.

Settles, Burr and Brendan Meeder (2016). *A Trainable Spaced Repetition Model for Language Learning*.

Ullman, Michael T (2004). "Contributions of memory circuits to language: The declarative/procedural model". In: *Cognition* 92.1, pp. 231–270.

Van Rijn, H. (2010). *SlimStampen: Optimaal leren door kalibratie op kennis en vaardigheid*. URL: http://onderzoek.kennisnet.nl/onderzoeken-totaal/slimstampen.

Van Rijn, Hedderik, Leendert Van Maanen, and Marnix Van Woudenberg (2009). "Passing the test: Improving learning gains by balancing spacing and testing effects". In: *Proceedings of the 9th International Conference of Cognitive Modeling*. Vol. 2. 1, pp. 7–6.

Van Thiel, Wendy (2010). *Optimize learning with reaction time based spacing*.

Van Woudenberg, Marnix (2008). "Optimal word pair learning in the short term: Using an activation based spacing model". In: *Unpublished master's thesis, University of Groningen*.