



THE UNIVERSITY
OF QUEENSLAND
A U S T R A L I A

**Predicting Responses to Spaced Repetition Flash Cards
with Machine Learning Techniques**

BY

JORDAN J. WEST

The School of Information Technology
& Electrical Engineering

and

Centre for Educational Innovation & Technology
The University of Queensland

Submitted for the degree of Bachelor of Engineering (Honours)
in the division of Mechatronic Engineering

5 November, 2012

Jordan West

41184882

PO Box 6170 St Lucia QLD 4067

5 November 2012

Prof Paul Strooper
Head of School
School of Information Technology and Electrical Engineering
The University of Queensland
St Lucia QLD 4072

Dear Professor Strooper,

In accordance with the requirements of Bachelor of Engineering (Honours) in the School of Information Technology and Electrical Engineering, I hereby submit the following thesis entitled:

Predicting Responses to Spaced Repetition Flash Cards with Machine Learning Techniques

The thesis was performed under the supervision of Dr. Mark Schulz. I declare that the work submitted in this thesis is my own, except as acknowledged in the text and footnotes, and has not been previously submitted for a degree at the University of Queensland or any other institution.

Yours sincerely,

Jordan J. West

Acknowledgments

This thesis would not have been possible without the support of my supervisor Dr. Mark Schulz whose input and guidance has been invaluable for the project.

I would like to thank Dr Yuriko Nagata for her assistance with this project and for allowing me to introduce the software to her students. I would very much like to thank the students of JAPN1023 who participated in the project, without whom none of this would have been possible. I hope that the students found value in using the software.

I wish to acknowledge the help of Dr Michael Harrington of the School of Languages and Comparative Cultural Studies for his advice with additions to the online learning software.

I would also like to express my sincere appreciation to the UQ CEIT team for their assistance with this project – particularly the advice and guidance of Dr. Alan Cody and Prof. Phil Long.

Last but not least I want to thank my friends for their encouragement and for putting up with me throughout the project, and finally Mum and Dad for being there and without whose support, care and encouragement I would never have made it this far in my university career.

Abstract

With the development of personal computers and smart phones, many new learning environments and learning methods are emerging. With this, educational data collection has become trivial and is becoming a powerful tool for improving education. This project created an online flash-card environment for memorising foreign vocabulary, and attempted to predict whether a student will be able to recall a particular word in a target language by tracking, recording and analysing reviews with machine learning techniques. This has many implications in education, including identifying struggling students, identifying difficult vocabulary to focus on, as well as improving methods of memorisation of foreign vocabulary. The web based software was available from desktop and mobile devices and allowed students to study anywhere. Over 13 weeks, 28 students completed over 7,500 reviews. The results were promising, with the system correctly predicting student recall of approximately 70% of flash-card reviews. The rate of forgetting was also plotted based on the review data, however the amount of data and number of reviews was too small for any significant findings. Future work might involve a larger number of more motivated students to gather a larger dataset.

Contents

1	Literature Review	3
1.1	Machine Learning	3
1.1.1	Multilayer Perceptrons	3
1.1.2	Support Vector Machines	4
1.2	The Forgetting Curve	5
1.3	Spaced Repetition	6
1.4	Similar Projects	8
2	Methods and Materials	11
2.1	Goals	11
2.2	Experimental Design and Data Collection	11
2.3	Participants	12
2.3.1	Participant Recruitment	12
2.3.2	Ethical Clearance	12
2.4	Online Learning Software Design	13
2.4.1	Requirements	13
2.4.2	Tools	14
2.4.3	Functionality	16
2.4.4	Vocabulary List	18
2.4.5	Spaced Repetition Algorithm Implementation	18
2.4.6	Data storage, formatting and output	19
2.5	Data Analysis and Prediction	20
2.5.1	Forgetting Curves	20
2.5.2	Prediction of Recall	20
3	Results	23
3.1	Usage Statistics	23
3.2	User Demographics	25
3.3	Forgetting Curves	25
3.4	Prediction of Recall	27
4	Discussion	29
4.1	Evaluation	29
4.1.1	Usage of Software	29
4.1.2	Forgetting Curves	29
4.1.3	Prediction of Recall	30
4.2	Changes to Original Scope	30
4.3	Potential Future Work	31
4.4	Conclusion	31

A	Participant Information Sheet	35
B	Participant Consent Form	37
C	Original Wireframes	39
D	Membit Screenshots	47
E	Data	49

Introduction

With the computing power available today, analysis of very large amounts of data is providing new insights in many areas that were previously impossible. One of these areas is education and the possibilities for analysing learning behaviour, patterns and memory could potentially lead to many improvements and efficiencies in the way education is conducted.

This thesis project focuses on self-motivated rote memorisation of foreign vocabulary in a university setting, with the use of a flash-card like online learning environment to record information about student study. The goal is to analyse the data recorded from the study of the foreign vocabulary and subsequently predict whether a student will correctly recall the foreign pronunciation and the meaning in their native language at a future point in time.

The report outlines some background behind the memory and the effect of spacing reviews, and the subsequent development of spaced repetition algorithms used to track study and improve efficiency of study. The report then outlines the process to build the online learning environment which records usage by university students as they study foreign vocabulary using the SuperMemo 2 spaced repetition algorithm. The spaced repetition algorithm keeps track of parameters for each individual vocabulary item for each user, which are then recorded in review data.

Analysis is performed on the review data including the use of machine learning algorithms attempting to predict the recall of a vocabulary item given only the known spaced repetition parameters of a word for a particular user. The same review data is also grouped to produce curves that illustrate the rate of forgetting a vocabulary item.

Chapter 1

Literature Review

1.1 Machine Learning

When developing a computer algorithm which takes some data as an input and returns an output, we will usually look at the underlying mechanism as to how we come to that conclusion. However it is not always possible to know what is the underlying mechanism that produces an output given certain inputs [8].

What makes machine learning algorithms unique is that they in a sense ‘learn’; given a large enough set of observed inputs and outputs, a machine learning algorithm builds a model which infers certain outputs given new inputs. We do not always need to know the underlying mechanism that translates a set of inputs to a set of outputs, as long as the inputs and outputs are related and a large enough data set is used.

What is ‘large enough’? This of course depends upon the problem domain, the number and type of inputs and outputs, and how the inputs are related to the outputs.

Various types of machine learning algorithms exist, however this report focuses on classification algorithms – that is algorithms which return an output as a ‘class’ rather than a continuous variable (regression). Additionally, we will only be looking at supervised learning since training data will contain outputs alongside inputs.

Complete descriptions of machine learning algorithms is beyond the scope of this report, however the following sections outline information relevant for the machine learning techniques that were used in this report.

1.1.1 Multilayer Perceptrons

Artificial neural networks are a form of machine learning inspired by the neural connections in the brain. Although their similarity to biology essentially ends there, they are a useful tool for modelling connections between inputs and outputs. By connecting the inputs to the outputs via a certain weighting, observing a set of inputs and outputs and adjusting weights the network can essentially be ‘trained’ produce those outputs given the same inputs.

For example if we have three inputs connected directly to one output via weights w_1 , w_2 and w_3 , we achieve a network as shown in figure 1.1.

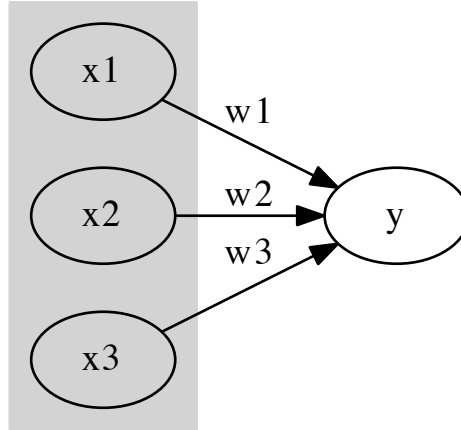


Figure 1.1: Simple network with 3 inputs and a single output

At its simplest, this is essentially a representation of a linear function shown in equation 1.1.

$$y = w_1x_1 + w_2x_2 + w_3x_3 = \sum w_nx_n \quad (1.1)$$

However, such networks are limited in that they can only solve linear problems [8]. Multilayer perceptrons take this model a step further and add ‘hidden layers’ of units between the input and output as shown in figure 1.2, which allows nonlinear regression.

In order for the networks to ‘learn’, the common technique is backpropagation with gradient descent in which the weights are initialised randomly to small values and then adjusted based on input and output values, error, and constants for momentum and learning rate. This process continues until convergence of error is reached. The momentum specifies how much of the previous weight should be incorporated, while the learning rate adjusts the magnitude of change [8].

Overtraining can occur when training the network over too many iterations. A classic symptom of overtraining is when training error remains fixed over many iterations but validation error continues to increase [8].

Often a single hidden layer is used due to the increased complexity introduced by adding several hidden layers. Adjusting the number of units in a hidden layer can affect the performance of the network, so a common approach to finding the best performance is to begin with a large or small number of units and gradually decrease or increase the number of units [8].

1.1.2 Support Vector Machines

A support vector machine (SVM) – also known as a kernel machine – is a method for linear classification and regression. SVMs operate in an n -dimensional space, where n is the number

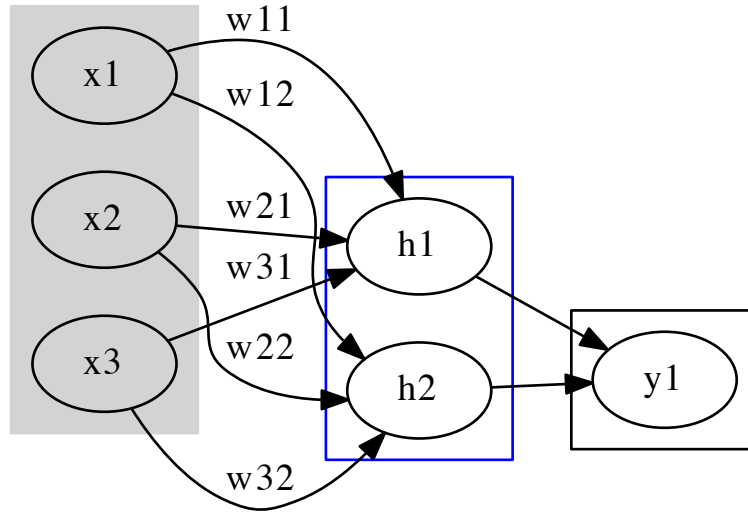


Figure 1.2: Multilayer perceptron

of input variables. By using a hyperplane in this space to separate instances based on their class, it is possible to place new instances in the space in order to predict their class.

However, this approach does not work for nonlinear problems. In nonlinear cases, a ‘kernel trick’ is used – a kernel function is applied to the data which extends the data into an extra dimension. Common kernel functions include:

- None (Linear) - No kernel function is applied.
- Polynomial
- Radial-basis
- Sigmoidal

1.2 The Forgetting Curve

The forgetting curve was first hypothesised by Hermann Ebbinghaus[7] who observed that forgetting tends to happen over time in an exponential fashion.

Ebbinghaus performed experiments on himself by attempting to memorise nonsense syllables. He hypothesised that memory retention follows a curve similar to equation 1.2 where R is the retention of the information, t is time and S is the relative strength of memory. This equation attempts to estimate the rate at which a person forgets newly learned information by capturing the exponential nature of forgetting which Ebbinghaus observed.

$$R = e^{\frac{-t}{S}} \quad (1.2)$$

The equation is not intended to provide quantitative prediction of recall but rather to illustrate

the point that most of the ‘forgetting’ happens soon after learning. Furthermore, the equation illustrates that if the ‘strength of memory’ S can be increased then the decay of the curve can be hampered.

1.3 Spaced Repetition

Spaced repetition is a method for memorising pieces of information by reviewing each piece of information at increasingly longer periods of time. It exploits the spacing effect of memory to improve efficiency in rote memorisation by attempting to have a student review a piece of information *just before* it is forgotten.

Various studies have found that spacing out repetitions over time is more effective than massed repetition or studying in a short space of time [17]. The type of spacing is however a more controversial topic, with some studies suggesting fixed intervals are better [4], while others suggest expanding intervals are more effective [15]. Regardless of this, spaced repetition algorithms usually use expanding intervals in order to make study time more efficient.

Spaced repetition can be used for memorising nearly anything - equations, vocabulary, numbers, phrases, diagrams. A typical application is using standard flash-cards, with a prompt to recall on one side and the correct answer on the other. Depending on how well the student recalls and the history of the flash-card, the flash-card is rescheduled after each review.

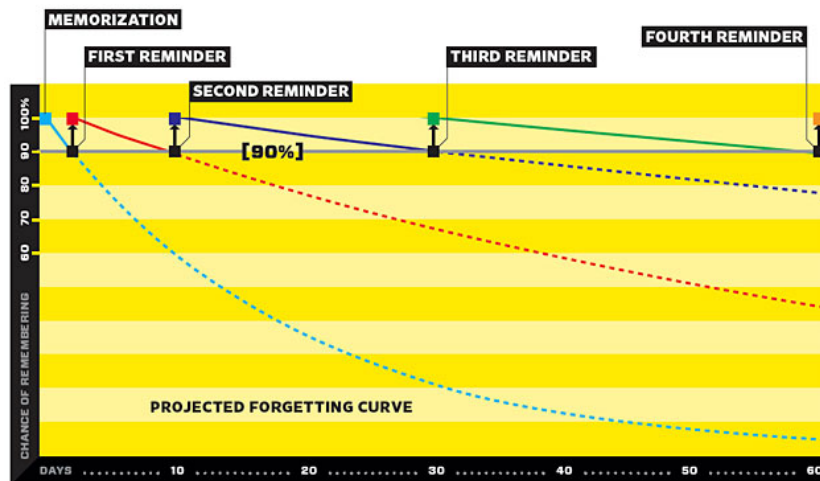


Figure 1.3: Projected Forgetting Curves with Spaced Repetitions [18]

Figure 1.3 shows an example of spaced repetition in action. After initial memorisation, a student might revise the content when there is a 90% chance of recalling the content correctly – which might be the following day. After this first revision, the student is likely to remember the content for a longer period of time and on day 10 will again have a 90% chance of recall. This continues for each subsequent revision, with each revision solidifying the content in the student’s mind and the chance of remembering diminishing at a slower rate over time.

As an added advantage, by prompting the user to recall and then to rate their answer – the user is in a sense being tested and thus actively engaging rather than passively ‘studying’. Testing and the associated active retrieval has been shown to improve retention significantly over passive study [1], [10], [11].

Spaced repetition software automates this process by storing relevant data alongside each flash-card in a database. The type of data stored depends upon the spaced repetition algorithm used.

Table 1.1: Example of reviews for a single flash-card in spaced repetition

Review Date	Recall	New Interval
1 Jan	Incorrect	0 days
1 Jan	Correct, difficult	1 day
2 Jan	Correct, difficult	2 days
4 Jan	Correct, easy	5 days
9 Jan	Correct, easy	12 days
21 Jan	Incorrect	0 days
21 Jan	Correct, easy	1 day
22 Jan	Correct, easy	3 days
...

Most algorithms store the current interval (in days) which represents the spacing, or an estimate of how long a student should be able to remember the word between reviews. On each successful review, the interval is increased and the card rescheduled based on the interval. Of course this is not an exact science, sometimes the student will not be able to recall, and some algorithms take this into account and adjust based on the difficulty of the particular word. An example of how a card might be rescheduled is shown in table 1.1.

Unfortunately, little publicly available research on spaced repetition algorithms has been carried out. Dempster (1988) postulated several potential reasons that spaced repetition itself has failed to catch on in education, which include a lack of demonstration in school-like activities, the relative recency of its development, and a lack of understanding of the effect [6].

Nonetheless, spaced repetition is a valuable tool in this project because of its inherent tracking of several parameters for each vocabulary item for each student.

SuperMemo 2 Algorithm (SM2)

As one of the first spaced repetition applications available for personal computers, SuperMemo and its algorithms paved the way for other applications such as Mnemosyne and Anki (see section 1.4). Developed by Piotr Wozniack[19], the *SuperMemo 2* algorithm was an enhancement over the *SuperMemo* algorithm primarily in that it would differentiate between items based on their difficulty to memorise[19].

The SuperMemo algorithm reschedules flash-cards a number of days into the future, known as the *interval*, where the interval is defined as the function $I(n)$ with n the current repetition number. For the first repetition, the interval is simply one day. For the second repetition, the interval increases to six days.

$$I(1) := 1 \tag{1.3}$$

$$I(2) := 6 \tag{1.4}$$

For all $n \geq 3$, equation 1.5 applies.

$$I(n) := I(n - 1) \times EF \tag{1.5}$$

EF is defined as the *easiness factor* of the flash-card. The easiness factor of the flash-card is adjusted on each review based on the answer given by the user with equation 1.6

$$EF := EF + (0.1 - (5 - q) \times (0.08 + (5 - q) \times 0.02)) \quad (1.6)$$

Where q denotes the user's self-rated accuracy of recall, according to table 1.2.

Table 1.2: User Rated Answers for the SuperMemo 2 Algorithm [19]

5	Perfect response
4	Correct response after a hesitation
3	Correct response recalled with serious difficulty
2	Incorrect response; where the correct one seemed easy to recall
1	Incorrect response; the correct one remembered
0	Complete blackout.

The easiness factor is bounded by the values $1.1 \leq EF \leq 2.5$ where 1.1 indicates the most difficult flash-cards and 2.5 indicates the easiest. Before a user begins studying a flash-card for the very first time, the associated easiness factor is set to 2.5.

1.4 Similar Projects

Memrise

Memrise is a private company which produces web-based flashcard software. Memrise states on their website: *“Our memory experts have spent long, sleepless nights tinkering with exotic algorithms so as to be able precisely to estimate the point at which you’re about to forget it.”* [13]. Their implementation details however have not been made public.

The Mnemosyne Project

Mnemosyne is open source spaced repetition software collecting anonymised data from its many users in order to evaluate the effectiveness of the implemented spaced repetition algorithm [14]. Mnemosyne uses a modified version of the Supermemo algorithm. Mnemosyne chose to avoid SuperMemo 3+ algorithms due to their complexity, however has recently moved to the SuperMemo 11 algorithm [14]. The project does not appear to have produced any papers or research publications at this time.

Anki

Anki is one of the most full-featured open-source spaced repetition applications available. Anki allows users to attach images, sounds, and even embed \LaTeX equations in flash-cards. The software is not designed for any research purposes but rather purely for learning and review of information.

The developer of Anki decided against SuperMemo 3 and later algorithms instead opting for the SuperMemo 2 algorithm because of the complexity that the SM3+ algorithms introduce [5].

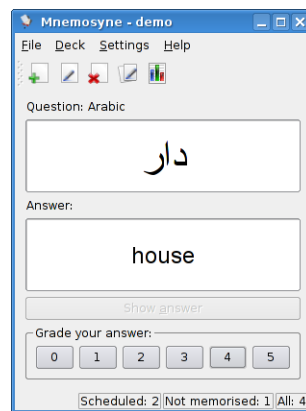


Figure 1.4: Screenshot of Mnemosyne in use



Figure 1.5: Screenshot of Anki in use

Chapter 2

Methods and Materials

2.1 Goals

This project aims to develop an online learning environment to gather data on spaced repetition reviews and to analyse that data in order to build a model of memory in two ways:

- Produce Forgetting Curves
- Predict Student Responses

Forgetting curves will provide an immediately visible overview of how time affects chance of recall, and how this changes after reviews of a piece of information. With these curves, it may be possible to support Ebbinghaus' theory of the exponential decaying nature of memory as well as to find optimal times at which to review information.

Prediction of student responses will evaluate the feasibility of predicting a student's recall of foreign vocabulary that the student has actively been reviewing with a spaced repetition method. With a good prediction accuracy and regular student use of the spaced repetition review software, it might be possible in future work to predict student test scores based on the content contained within. This project however only aims to evaluate the accuracy of predicting reviews within the software.

2.2 Experimental Design and Data Collection

In order to gather data, the online learning environment records each and every review by a participant. A review is defined as a student recalling a Japanese vocabulary item on an electronic flash-card and the subsequent rating of their recall.

The 'front' of the flash-cards will prompt the student with an expression, or the Japanese written form of a Japanese word. On the 'rear' of the flash-card the pronunciation is written in Hiragana – the Japanese alphabet – along with the English meaning of the word.

A student will see the front of the flash-card and attempt to recall both the pronunciation and the meaning of the word. The user then chooses to show the answer and rates their own answer on a scale of 0-5 which is then used to reschedule the card for a future date. Another card is then displayed to the user and the process repeats.

If a user selects a response of 0-2 indicating their failure to recall the pronunciation and meaning of the word, the word is queued for review at the end.

The review process continues until the student has studied all cards due for review that day, all failed flash-cards, and 20 new flash-cards or until the student logs out. A new card is defined as a card that a student has never studied using the online learning environment. Assuming that the student studies the maximum 20 new cards each day they review, a student must use the software for 12 days and complete all reviews in order to have studied each word in the complete vocabulary list at least once. After this point, all flash-cards shown are reviews of already studied cards. Since new cards contain no information about past study, these reviews are impossible to use in predicting a users recall. Therefore it is important that many reviews are captured of previously studied cards.

Students have the freedom to use the software at any time. They are encouraged to use it every day in order to make sure that flash-cards are reviewed on the day that they are due (increasing the chance that they will be recalled correctly) however for this experiment reviews of overdue cards is not necessarily a problem as the overdue time is taken into account as an additional variable when predicting recall, as well as providing more diverse data when generating forgetting curves.

2.3 Participants

2.3.1 Participant Recruitment

Participants were students of the introductory Japanese course JAPN1023. The author introduced the project in the first week of classes and handed out registration cards to students who chose to participate and signed a participant consent form.

2.3.2 Ethical Clearance

As with any project involving humans, the details of the project must be reviewed and approved by the University Human Ethics Committee prior to any student participation.

An application for review was submitted in June and approved with modifications on 25 July 2012 in time for the second week of semester.

The application included details of the methods of data collection, recruitment of participants, and approval by a ‘Gatekeeper’ who provides access to participants - in this case the course coordinator of JAPN1023, Dr Yuriko Nagata.

Data was to be stored anonymously and securely. In order to ensure participation was anonymous, cards containing unique codes were to be handed out randomly to participants to allow them to register online. Student review data was tied to a unique number in the database that could not be traced back to individual students. Email addresses were collected from students in order to allow them to log in and to reset their password if required, however exported review data was stored only against a unique number in the database which could not be traced back to individual students. Participants were also given an information sheet (See Appendix A) and a consent form (See Appendix B) to sign and return before receiving a registration card.

2.4 Online Learning Software Design

The online learning environment for students, named ‘Membit’ was built as the main interface for data collection. This section outlines the design and construction of the software.

2.4.1 Requirements

A number of requirements were set out for the online learning software. These are listed below along with how these requirements are achieved.

Should be easily accessible to students. Removing barriers to usage will encourage students to use the software more often.

By providing access to the learning environment online via a web-based interface, students can access the software anywhere - including from university computers without requiring installation of any software. The application is simply accessed with a web-browser by visiting <http://membit.herokuapp.com/>.

Should be secure and anonymous. For peace of mind for students, and to meet ethical standards data must be collected anonymously and securely. Students may be more reluctant to use the software with the knowledge that their individual progress is being tracked.

This is achieved by allocating each student a unique code with which to register. Students are then recorded in the database using a newly assigned number which is unknown to the users. Downloaded review data should only refer to users with this number, meaning that even if the registration code is known, a user cannot be identified from downloaded review data. Furthermore, all data should be stored on a secure password protected server and all usage of the system via a secure connection to the server.

Should be easy to update and/or fix bugs. It is important that bugs can be fixed quickly and efficiently, as downtime could affect results if users are prevented from performing reviews when they are due. Building the application as a web application with a fast deployment pipeline will ensure that bugs can be fixed and patches published to the live application quickly with users not required to perform any software updates.

Data should be captured and stored immediately. Since the project period is short, users cannot be relied upon to submit their data manually in time for analysis. Therefore reviews should be recorded immediately.

Only nominated students should be able to access the system To ensure some uniformity of the student level of Japanese knowledge, only students in JAPN1023 were allowed to use the system. Student who already have a strong knowledge of the Japanese language could potentially skew results since they may already know a majority of the vocabulary within the software. By requiring users to register with a one-use-only registration code, students cannot share their registration code with somebody outside of the class.

2.4.2 Tools

This section outlines the software tools that were used for the project and reasoning for choosing these tools.

Git and Github (<http://git-scm.com/>), (<http://www.github.com/>)

Git is a distributed version control system (VCS) which tracks changes to source code (often amongst multiple developers) and keeps a complete history of changes. This is invaluable when a change in code occurs that results in a critical bug. Versions can be compared to find the change that introduced the bug, and production code can be reverted if need be [16].

Git repositories can be hosted anywhere, however Github offers free Git repository hosting for open source projects. It also allows users to 'fork' public repositories to create their own version of a project. For this reason it is useful for research projects as the project can be picked up and continued at any time by others.

Git was selected for this project because of its portability (moving repositories between servers is trivial). Github was chosen as it is free, encourages collaboration and is also the tool of choice for the Centre of Educational Innovation in Technology [20], under which this project was completed.

Ruby on Rails (<http://www.rubyonrails.org/>)

Ruby on Rails (aka Rails) is a popular open source framework for developing web applications[2]. Rails was originally extracted from a commercial application (Basecamp by 37Signals) to create a generic web application framework [3] written in the Ruby language. Rails is designed for rapid development and provides many guidelines that the developer is recommended to follow in order to speed up development. Additionally, as an open source project Rails has gained many developers who have contributed back to the community by sharing reusable components (known as Ruby Gems) with the community. This means many pieces of functionality can be used in a project without rewriting, speeding up the development process. Gems used in this project include:

Prawn Provides PDF output - used for generating registration code cards

CanCan User authorisation - Allow and deny access to users based on their role (participant, administrator, teacher)

Highcharts-Rails Adds the Highcharts library to the application (See section below)

Heroku (<http://www.heroku.com/>)

Heroku is a private company offering hosting for Ruby on Rails applications with automated deployment. While deploying a Rails application on a server normally requires system administrator knowledge and a significant amount of time to install, Heroku allows deployment via Git and automatically installs dependencies to get an application up and running in less than a minute.

Heroku was chosen over a private server for this project since it was necessary to be able to push updates to the live application quickly in order to respond to bugs and to reduce time spent finding faults in the server.

Backbone.JS (<http://www.backbonejs.org/>)

Backbone.js is an open source Javascript framework providing a model oriented structure for Javascript heavy web applications. Backbone.js allows data to be easily linked to user interface components and synchronise with the server, so that any changes to the underlying data automatically update the user interface. It was selected because of its integration with Ruby on Rails and because of the author's familiarity with the library.

Highcharts (<http://www.highcharts.com/>)

Highcharts is a commercial Javascript framework that provides graphing capabilities to web sites. Highcharts allows free usage by non-commercial projects. Highcharts was selected for graphing usage statistics on the website because of the features it provides in addition to recommendations on websites such as Stack Overflow [9].

Twitter Bootstrap (<http://twitter.github.com/bootstrap>)

Twitter Bootstrap is a set of default styles for websites and web applications, provided as open-source by Twitter. Using Twitter Bootstrap rapidly speeds up theming of a web application with default looks for navigation, buttons, text and layout.

See figure 2.1 for a comparison of default styling with and without Twitter Bootstrap

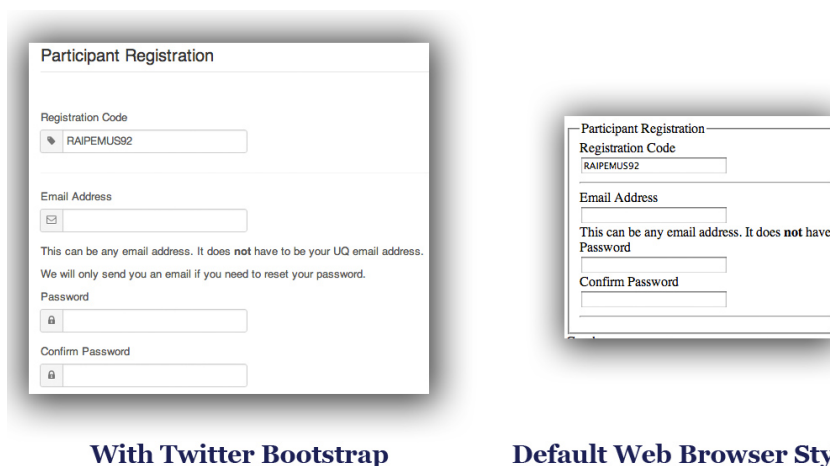


Figure 2.1: Comparison of a page with no styling and Twitter Bootstrap default styling

More significantly, Twitter Bootstrap offers a 'responsive' layout system which provides a reduced screen size (ie. smartphone) layout with little to no extra work on the part of the developer. This means a smartphone version of the web application could be designed at the same time. Twitter Bootstrap was also chosen for this reason.

The R Programming Language R is an open source programming language designed primarily for statistical computing. Many packages are available for R which provide functionality including various machine learning algorithms.

R was selected since it is open source and therefore it is possible for others to recreate the experiment without purchasing expensive software such as MATLAB. Additionally, the libraries available in R (such as e1071, the Support Vector Machine library) generally provide more customisability over those available in MATLAB as standard.

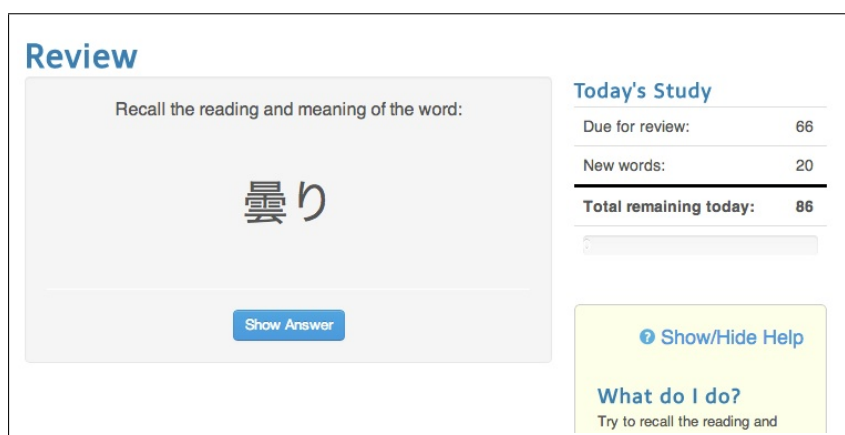


Figure 2.2: The ‘front’ of a flash-card during review

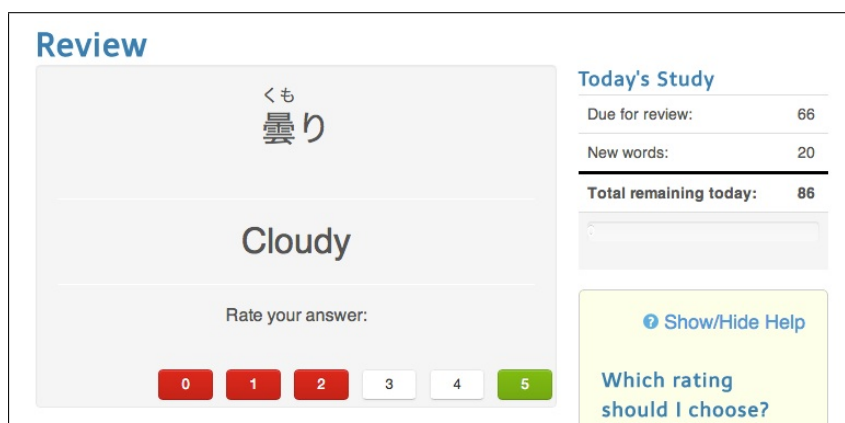


Figure 2.3: The ‘back’ of a flash-card during review. Here the user is asked to rate their response

2.4.3 Functionality

Reviewing

The review interface was designed with a responsive Javascript frontend in order to reduce lag between reviews. While a student is reviewing a flash-card, the following card is pre-loaded and displayed immediately after a response is selected.

Figures 2.2 and 2.3 show the ‘front’ and ‘back’ of the flash-cards respectively. The user is prompted to recall the pronunciation and English meaning of the word on the front. The answer can be revealed either by pressing the ‘Show Answer’ button or pressing the ‘F’ key on the keyboard, allowing the user to compare their recalled answer with the correct answer.

Finally the user rates their recall on the 0-5 scale using either the buttons on screen or the keyboard, and the next card is displayed for the process to repeat.

Withdrawal of Participation

To meet ethical standards, participants are able to withdraw from participation from within the software at any time using the ‘Withdraw Participation’ function which immediately deletes all associated information.

Participant card Your registration code is: KEAPUCEZ84 Register at http://membit.herokuapp.com/	Participant card Your registration code is: CUEPAYUP46 Register at http://membit.herokuapp.com/
Participant card Your registration code is: CEUPOKOX72 Register at http://membit.herokuapp.com/	Participant card Your registration code is: MEOPADOQ49 Register at http://membit.herokuapp.com/
Participant card Your registration code is: YUEPETAR43 Register at http://membit.herokuapp.com/	Participant card Your registration code is: XEUPALAW89 Register at http://membit.herokuapp.com/
Participant card Your registration code is: PUOPIGOY62 Register at http://membit.herokuapp.com/	Participant card Your registration code is: NIUPAQUK92 Register at http://membit.herokuapp.com/
Participant card Your registration code is: SOAPURAV78 Register at http://membit.herokuapp.com/	Participant card Your registration code is: NOAPAKAH32 Register at http://membit.herokuapp.com/

Figure 2.4: Sample printout of registration codes

Password Reset and Change

Students are able to reset their password by entering their email address into a password reset page. An email is sent to the account associated with the email address containing a new random password which can be used to log in. A new password can be set after logging in.

Administration

A secured administration section was built to facilitate the operation of system, including features such as:

- Generating registration codes
- Viewing error logs
- Downloading anonymised formatted review data

Registration Codes

An interface for the administrator allows generation of registration codes. Registration codes are strings of random numbers and letters that can be used to create an account. Each code can be used only once, and is associated with a role in the system which carries over to the account created with the code. For example, the administrator may wish to create 50 participant codes, 5 tester codes and 1 teacher code that can be given to the respective users. This ensures a hard limit on the number of each type of user that can register.

The system also provides an interface for printing sheets of registration codes on cards. This allows the sheets to be printed onto card and cut out to be individually handed to students. With over 100 students in the class, this feature was added to reduce the time in writing out 100 individual cards by hand.



Figure 2.5: Sample registration card for a tester

Teacher Dashboard

A common feature in online learning environments is a dashboard for teachers to track student usage. A basic dashboard was designed to visualise usage of the system over time and to show information about the vocabulary – which words were difficult and which were not so difficult for students to remember based on the average easiness factors for all students.

2.4.4 Vocabulary List

The vocabulary selected was the vocabulary for the JAPN1023 course in which the participants were enrolled. It was hoped that since the vocabulary is part of the course content, students would be encouraged to use the software in order to help them learn the vocabulary. A total of 240 words were entered into the database.

The vocabulary was entered into a spreadsheet and run by the course coordinator for verification. The vocabulary list was then exported to a .csv file and text manipulation software used to convert the list to Ruby code which inserts records directly into the database. A .csv import mechanism was considered to replace this process, however since the import is a one-time process this was decided against.

2.4.5 Spaced Repetition Algorithm Implementation

The SuperMemo 2 algorithm was selected in favour of newer SuperMemo algorithms because of the widespread use of this algorithm over newer algorithms. Both Mnemosyne and Anki developers chose the SuperMemo 2 algorithm due to the complexity of the SuperMemo 3+ algorithms – although the Mnemosyne developer eventually switched to the SuperMemo 11 algorithm [14], [5]. Additionally, the SuperMemo 3 algorithm was decided against as the flash-cards are not considered independently – a review of one card can affect the values for similar cards [5]. For simplicity sake in this project, all flash-cards are considered independent from each other.

The SuperMemo 2 algorithm was implemented in Ruby as part of the application. Data is stored in an SQL database. The variables required for the spaced repetition algorithm are stored in a ‘UserWords’ table which contains a record for each individual user and vocabulary item. This means each time a new user is created, 240 records are added to the ‘UserWords’ table to keep track of each word. Upon review, the variables are pulled from this table, recorded in the

reviews table along with the user's response and a new interval and easiness factor calculated based on the previous values and the user response.

In order to provide more review data at smaller intervals, the interval for the second repetition was halved from the original SuperMemo 2 algorithm from 6 days to 3 days. This also affects subsequent reviews since each new interval is calculated on the previous interval.

The same 0 to 5 rating system was used for the SuperMemo 2 algorithm, with a description of each rating written beside the review interface. As with the original SuperMemo 2 algorithm, if a user fails a review – that is, selects a response from 0 to 2 – the repetition number is reset to zero and thus the interval reset. A flag is set against that item recording it as 'failed' and it is shown again to the user in the same session until they recall the item correctly. The easiness factor however is not reset on failing an item and is calculated as per usual.

The system selects cards for the user to review in the following order:

1. Cards due for review
2. New cards
3. Failed cards

Since failed cards are effectively due immediately, they are cycled so that the failed card with the oldest previous review is displayed first.

2.4.6 Data storage, formatting and output

The application data is stored in an SQL database accessed with the Ruby on Rails framework. The following tables contain the majority of useful data:

Users A listing of all registered users, and a secure one-way hash of their password information to provide login functionality.

Words A listing of all 240 vocabulary items and their associated expression, meaning and readings.

UserWords Records spaced repetition parameters for a particular user and word association. Each user has their own record of progress with each individual word.

Reviews Records of every review completed with the software. A review captures the current state of the associated UserWord record when a review is completed, and the user's selected answer for that review. The review table keeps track of a user's role so that teacher, administrator and tester review data can be removed when the data is exported leaving only participant review data.

UserLogins Records each 'login' by each user. Since users are not automatically logged out, a login is recorded whenever a student uses the software after a minimum 15 minute break.

UserInfos Contains demographic data on the users - their gender and whether English is their first language.

A full listing of tables and fields is available in the `/db/schema.rb` file in the Membit source code.

Review data is made available via an administrator login. The software converts the table of anonymised participant reviews to a .CSV file for download. CSV was chosen because of its portability – almost all data analysis packages support CSV files. Data could also be filtered

before download, to remove unwanted entries. Most importantly, it was made possible to filter out ‘new’ reviews, ie. reviews for which it was the first time for a student to study a word. These reviews contain almost no useful data for prediction since there is no history for that student reviewing that word.

Fields contained in the downloaded review data are described in table 2.1

2.5 Data Analysis and Prediction

2.5.1 Forgetting Curves

Forgetting curves can be generated from the recorded reviews by grouping review data on the following variables:

- Repetition number
- Actual interval

Given these groups, the chance of remembering a fact for a specific repetition number and interval can be estimated with equation 2.1:

$$P(\text{correct}, \text{incorrect}) = \frac{\sum \text{correct}}{\sum \text{correct} + \sum \text{incorrect}} \quad (2.1)$$

However with groups of data for which there is minimal review data, this equation will yield large errors. For example, if a grouping of data is as shown in table 2.2:

Given the sample of the three reviews in table 2.2 – equation 2.1 would yield $P = \frac{1}{1+2} = 0.333$.

However the standard error as calculated with equation 2.2 yields $\sqrt{\frac{0.333(1-0.333)}{3}} = 0.272$

$$\text{Standard Error} = \sqrt{\frac{p(1-p)}{n}} \quad (2.2)$$

We cannot say with any accuracy from this data that the probability of remembering a word given these inputs is 0.333. On the other hand, with a sampling of thousands of reviews for a given repetition number and actual interval the error is reduced and a probability can be considered more accurate.

This issue is one of gathering enough data, and so various thresholds for the minimum number of reviews (n) were arbitrarily selected based on the amount of data available.

Due to the limited amount of data expected, forgetting curves will be constructed on the basis of the entire set of users and not for individual users.

2.5.2 Prediction of Recall

Prediction of recall is carried out using the R programming language and packages which implement machine learning algorithms. Packages used include:

e1071 An implementation of libsvm, a common Support Vector Machine library.

Table 2.1: Fields contained within downloaded data files

Field	Type	Description
User Word ID	ID	A unique ID for a user and vocabulary item pair (a user-word).
Word ID	ID	A unique ID for each vocabulary item.
User ID	ID	A unique ID for each user.
Was New?	Boolean	True if this is the first time the user has studied this vocabulary item. False otherwise.
Overdue Time	Continuous	Number of days after the vocabulary item was due for review that it was studied.
Previous Incorrect Count	Discrete	Number of times this vocabulary item has not been recalled correctly before this review.
Previous Correct Count	Discrete	Number of times this vocabulary item has been recalled correctly before this review.
Previous Easiness Factor	Continuous	The easiness factor that was assigned to this vocabulary item for this user before this review.
User Rated Answer	Discrete	The answer the student selected (0 - 5)
Time to Answer	Continuous	The amount of time between seeing the front of the flash-card and selecting an answer.
Correct?	Boolean	Whether the recall was considered correct. FALSE if answer selected was 0-2, TRUE if 3-5.
Previous Repetition Number	Discrete	The repetition number before this review. NULL = This is the first review, 0 = The previous review was the first. Note that this number is reset after a card is failed.
Previous Interval	Continuous	The interval, or amount of time between the previous review and the due date for this review.
Actual Interval	Continuous	The actual time between the previous review and the current review.
Was Failed?	Boolean	Whether the user-word was failed on the previous review. TRUE if the card was failed before this review, FALSE if the card was not failed.
Previous Attempts	Discrete	Total number of attempts at reviewing this user-word prior to this review
Previous Answer	Discrete	The user rated answer of the previous review
Previous Time to Answer	Continuous	The time the user took to answer on the previous review
Word Average Easiness Factor	Continuous	Average easiness factor for this word for all users

Table 2.2: Example of too few review samples after grouping

Repetition Number	Actual Interval	Correct?
3	18	TRUE
3	18	FALSE
3	18	FALSE

nnet A neural network implementation supports multilayer perceptrons with logistic and linear outputs.

The review data is downloaded as .csv from the administration section of the online learning environment. An R script loads the .csv file into the environment, strips out the irrelevant columns in the data, and trains and validates the selected machine learning algorithms on the data. Table 2.3 shows the variables from the review data that are used for the training and validation. Note that the output ‘Correct’ is calculated based on the ‘User Rated Answer’ and is simply a convenience to refer to whether or not the user correctly recalled a word without taking into account the subjective difficulty of recall.

The inputs shown in table 2.3 are the same variables which are either stored or can be calculated from values stored alongside each user-word in the database by the spaced repetition algorithm. This means that given a well trained algorithm and a set of user-words for a user, it should be possible to calculate which words the user will correctly recall at any given point in time.

Multilayer perceptrons and support vector machines were selected for their handling of nonlinear regression. In the case of support vector machines, both a linear and radial basis kernel function was selected for comparison.

Table 2.3: Inputs and Outputs to Machine Learning Algorithms

Field	Type	Input/Output
Overdue Time	Continuous	Input
Previous Incorrect Count	Discrete	Input
Previous Correct Count	Discrete	Input
Previous Easiness Factor	Continuous	Input
Previous Repetition Number	Discrete	Input
Previous Interval	Continuous	Input
Actual Interval	Continuous	Input
Previous Attempts	Discrete	Input
Previous Answer	Discrete	Input
Previous Time to Answer	Continuous	Input
Word Average Easiness Factor	Continuous	Input
User Rated Answer	Discrete	Output
Correct	Boolean	Output

Chapter 3

Results

3.1 Usage Statistics

A total of 28 students registered to use the software and 7,879 total reviews were recorded by 3 November 2012.

Students showed interest in the software during the in-class introduction, however many students registered and used the software only once. Figure 3.1 shows the initial interest in the software as a spike in the number of visits toward the beginning of semester. This graph tracks all visits including users that have not yet registered.

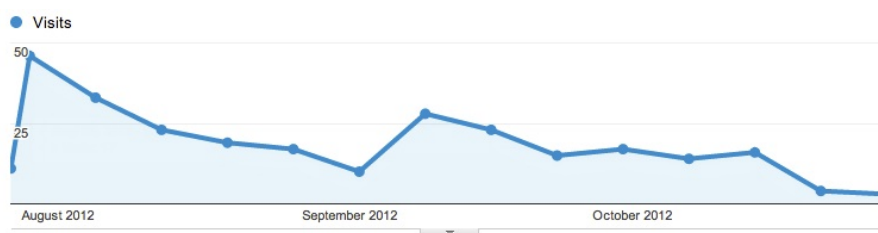


Figure 3.1: Google Analytics data on total number of visits to the application

Figure 3.2 shows most students accessed the software from a desktop computer, while a handful accessed the software exclusively from a mobile device. A single user accessed the software from both a desktop and mobile device.

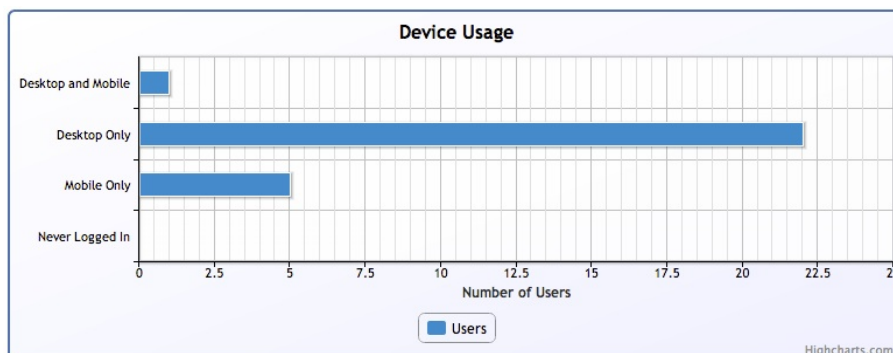


Figure 3.2: The number of users according to the device used to access the software.

The number of reviews completed per user is one of the more important statistics as it shows the diversity of ‘useful’ review data. Too many first reviews are useless as they contain no

useful data on which to later predict an answer. Figure 3.3 shows that eight users completed only new reviews (the 1-20 range) after registering.

For the purposes of the following graphs, users were classified as ‘active’ or ‘inactive’ based on the number of total reviews completed, with active users considered as those who completed more than 200 reviews.

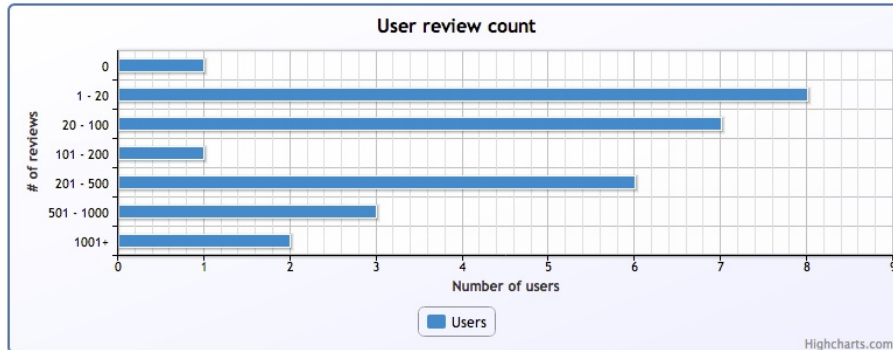


Figure 3.3: Number of users per total reviews. Users who had completed at least 200 reviews were considered ‘active users’.

Figure 3.4 shows the average number of reviews per user across the semester. Users were divided up into ‘active’ and ‘inactive’ groups to avoid inactive users skewing the data, though an average across all users is also shown with the blue line.

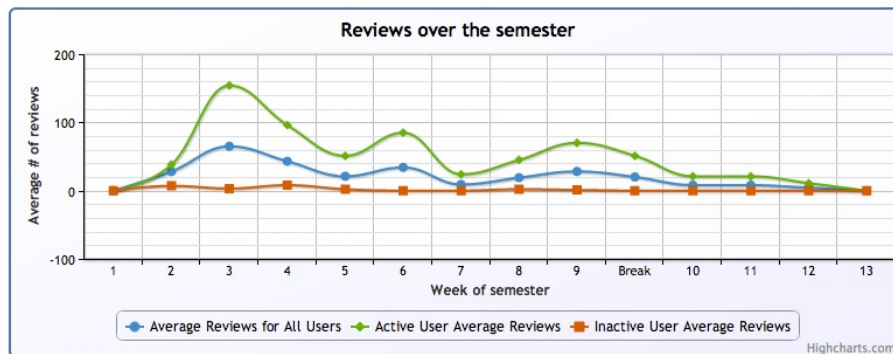


Figure 3.4: Usage of the system over the semester.

Users tended to complete the most reviews on their first day after registering. Figure 3.5 groups reviews by the number of days since registering, showing a very fast decline in number of reviews in the first couple of weeks after registering.

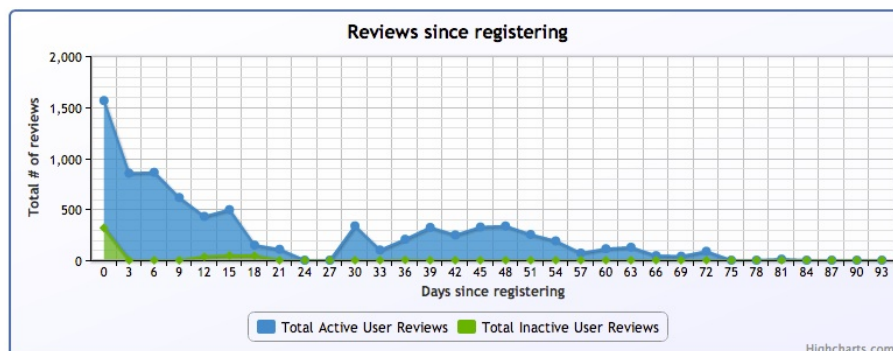


Figure 3.5: Usage of the system shown as the number of days since registering.

Figure 3.5 shows the total number of reviews per user after registering. Only a few users registered in the second week of semester, with many more registering in the following weeks. However many users only completed reviews in the first few days after registration and stopped usage after that.

Only a few users used the software regularly; table 3.1 shows the statistics for the top five users by number of reviews.

Table 3.1: Top users and study statistics

User ID	Number of logins	Vocabulary studied	Total reviews
20	59	100%	1910
10	13	75%	1337
21	13	45%	831
19	9	50%	781
26	9	58%	718

3.2 User Demographics

Table 3.2: Gender of registered users

Gender	No. of users
Male	7
Female	21

Table 3.3: Native language of registered users

Native language	No. of users
English	21
Other	7

3.3 Forgetting Curves

Generated from Recorded Reviews

Figure 3.6 shows the review data grouped as data points by the review number and days since previous review (actual interval). The threshold n is the number of reviews required for a data point to be displayed.

The progression from $n \geq 5$ to $n \geq 100$ shows a significant reduction of noise in the data points, where n is the number of reviews required to generate a single data point. Ideally this threshold would be much higher, however with the limited data set available increasing the threshold any more would reduce the number of data points visible.

A much larger number of reviews were available for the first review, so the standard error is reduced. The standard error as shown on the $n \geq 30$ graph in figure 3.6 for the first data point is 1.7% with the chance of remembering calculated from $n = 774$ review samples. In contrast,

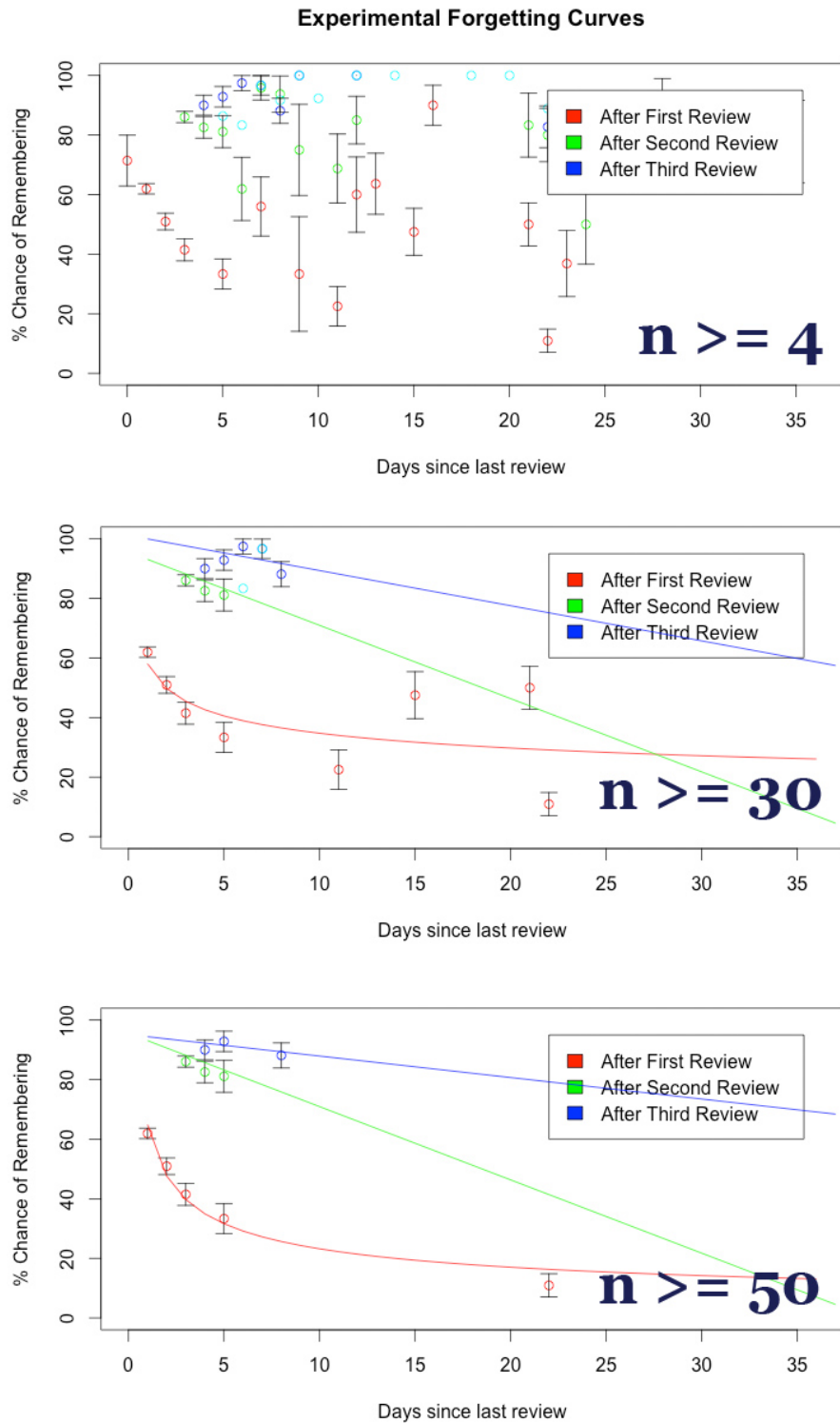


Figure 3.6: Forgetting curves produced from recorded review data with various thresholds for number of reviews required to generate a data point

the data point at fifteen days for the first review was calculated from only $n = 40$ available review samples, and has a standard error of 7.9%.

Equation 3.1 shows the curve that was fitted to the data for recall after the first review, while equations 3.2 and 3.3 show the lines fitted to the data for recall after the second and third reviews respectively.

$$P = \frac{x^{-0.4466}}{0.0154} \quad (3.1)$$

$$P = -2.46x + 93.1 \quad (3.2)$$

$$P = -0.72x + 94.4 \quad (3.3)$$

3.4 Prediction of Recall

The machine learning algorithms tested averaged just under 70% accuracy, with the best performance on the validation set by the radial kernel based support vector machine at 70.1%.

Table 3.4 lists the algorithms tested and the accuracy of classification on both the training and validation sets.

Table 3.4: Comparison of the performance of various machine learning algorithms on the output ‘Correct’. Accuracies are averaged over 15 runs.

Algorithm	Training set accuracy	Validation set accuracy
Linear SVM	69.41%	68.92%
Radial SVM	71.55%	70.14%
Neural Network (Linear, 9 hidden units)	72.65%	69.34%

Table 3.5: Confusion matrix of SVM Radial Kernel on validation data with output ‘Correct’

		Actual	
		Incorrect	Correct
Predicted	Incorrect	742	460
	Correct	321	1008

Table 3.6: Confusion matrix of SVM Radial Kernel on validation data with output ‘User Rated Answer’

		Actual					
		0	1	2	3	4	5
Predicted	0	104	54	34	47	25	43
	1	48	155	76	98	48	40
	2	44	100	217	101	109	78
	3	3	37	9	156	55	12
	4	9	43	48	77	151	69
	5	11	21	50	31	62	266

Chapter 4

Discussion

4.1 Evaluation

4.1.1 Usage of Software

Overall usage of the software was lower than expected. Although 80 consent forms were collected, only 28 students actually registered with the software. Participants were provided with contact details of the author, however no reports of lost registration codes or difficulties in registering were made.

Usage after registration was equally low. A majority of users logged in only once or twice after registration and stopped usage after this. A few users shown in table 3.1 made up the majority of review data and thus the data was likely skewed towards these users, particularly after removing ‘new’ reviews. Unfortunately only a single participant studied all 240 vocabulary items in the list.

The large spike in the first few days in figure 3.5 suggests that students tried the software out and did several reviews immediately after registering, but were unmotivated to return and complete reviews in subsequent days and weeks.

Interestingly, although one might expect study frequency to increase in the lead up to assessment, assessed class quizzes were held in weeks 5, 8, and 11 and appear to have had little effect on the usage of the software. The number of reviews is however not necessarily a good indicator of student motivation to study – it could simply be the case that not as many reviews were due during those weeks.

Qualitative feedback from students on their usage was not gathered, however the course coordinator mentioned that students felt they had no time to use the software as they were too busy with other assessed work.

4.1.2 Forgetting Curves

The forgetting curves appear to bare some resemblance in shape to those hypothesised by Ebbinghaus, however too little data is available to support it with any certainty. It appears that chance of recall for shorter periods of time does increase with each spaced review.

The curve for recall after the first review is the most promising, showing a very sharp decline of recall probability in the first days after the review and flattening beyond that. The chance of recall after the second and third reviews show a much higher probability of recall in the first few days as expected, however data beyond this is unavailable and as a result the curve fit appears linear.

With a larger number of reviews over a longer period of time than was achieved in this project, it should be possible to construct much more accurate forgetting curves. Additionally for students using the software daily, forgetting curves could be constructed per user and compared amongst different students. With a larger review dataset, it would also be possible to compare forgetting curves across different words - some words might be more quickly forgotten while others might remain for longer, for example if students were taught a mnemonic to aid retention [12].

The Mnemosyne project is one such possible source of this extra data, as it has been collecting data from its users since 2006 [14].

4.1.3 Prediction of Recall

Overall the results indicate that there is some possibility of correctly predicting whether a student will recall the meaning and pronunciation of a word given the spaced repetition parameters and a history of student reviews for that word. With a 70% accuracy, the results are promising and warrant further investigation.

The confusion matrix in figure 3.6 shows that predictions for the user's answer are generally close to the actual answer selected, however predictions tend towards the answer 2. It was observed that users often chose the user rated answer 2, suggesting that they almost knew the word but just couldn't quite remember it at that time, recalling it easily after seeing the answer.

With the similarity of results among the machine learning techniques employed, it appears that the limitations are with the data rather than with the machine learning techniques and parameters selected. With these results it seems that there is a reasonable correlation between the variables stored by the spaced repetition algorithm and the chance of recalling the meaning and pronunciation of a foreign word, however they are limited.

The addition of other variables into the review data could improve correlation, however where these variables are sourced is a complicated matter. Additional variables could potentially include relevant information such as how long ago the words were studied during class, the number of kanji in the sentence, the chance of confusing the word with other similar looking words, and the word frequency in the target language.

One drawback of this study is the subjective rating of recall. Although the ratings were defined on the screen where students review, students could select any rating and choices of rating may have differed between participants. Since the prediction algorithms were run across all students, those students who used the software more often likely skewed the data.

4.2 Changes to Original Scope

A number of changes were made over the course of the project. Originally it was planned to include an online quiz with which to score students and compare their scores to a predicted score by the machine learning algorithms. This was removed from the project in order to refine

the scope of work. Since the machine learning algorithms should be evaluated on their own merit, it was decided that these should be focused on rather than prediction of test scores using the machine learning algorithms.

As shown in the original wireframes in appendix C, during review users would be asked to enter their own answer before the correct answer was revealed to them. They would then rate their own answer against the correct answer. This was to be used for gathering possible answer variations for a particular word that could then be used to automatically grade a quiz. However since the quiz component was removed, this was no longer required. Additionally it would introduce too many variables – if incorrect answers were marked as correct, they would then be graded incorrectly in the quizzes. This combined with the fact that manually grading many quizzes could prove too much work, this was removed from the scope and users instead evaluate their recall without entering their answer.

Several variables were added to the review data output after reviewing began in an attempt to improve the accuracy of prediction. These variables were:

- Previous Answer
- Previous Time to Answer
- Word Average Easiness Factor

Since all of these values can be calculated from past data, the values were retroactively added to review data and all new reviews automatically included them. The addition of these variables improved the accuracy across all machine learning algorithms by 2-4%.

4.3 Potential Future Work

Future work should almost certainly gather a larger amount of data over a longer period of time. More motivated students who review more frequently would also provide richer data with a higher number of repetitions per word.

Additionally various spaced repetition algorithms should be trialled, potentially by splitting users into separate groups. Other algorithms keep track of words using different variables which may or may not correlate as well to the chance of recall.

Having participants sometimes review cards just before or after they are due would also provide more diversity among the overdue times – reviewing before cards are due would result in a negative overdue time and it could be expected that recall is higher in these cases. This would also avoid review data grouping around common intervals – for example 1 day and 6 days in the case of the SuperMemo 2 algorithm.

4.4 Conclusion

Overall this project was successful in showing that there is potential for modelling memory through recording reviews. The online learning software built was reliable and had no major issues, however the usage of the software was lower than expected and as a result the data collected was not extensive enough for full analysis.

Bibliography

- [1] G. A. Allen, W. A. Mahler, and W. K. Estes. “Effects of recall tests on long-term retention of paired associates”. In: *Journal Of Verbal Learning And Verbal Behavior* 8.4 (1969), pp. 463–470.
- [2] M. Bachle and P. Kirchberg. “Ruby on Rails”. In: *Software, IEEE* 24.6 (Dec. 2007), pp. 105–108. ISSN: 0740-7459. DOI: 10.1109/MS.2007.176.
- [3] Cloves Carneiro Jr. and Rida Al Barazi. *Beginning Rails 3*. 1st edition. Apress, 2010.
- [4] Shana K. Carpenter and Edward L. DeLosh. “Application of the Testing and Spacing Effects to Name Learning”. In: *Applied Cognitive Psychology* 19.5 (2005).
- [5] Damien Elmes. *Anki Frequently Asked Questions*. 2012. URL: <http://ankisrs.net/docs/OldFrequentlyAskedQuestions.html> (visited on 10/28/2012).
- [6] Frank N. Dempster. “The Spacing Effect: A Case Study in the Failure to Apply the Results of Psychological Research”. In: *American Psychologist* 43.08 (Apr. 1988), pp. 627–634. URL: http://andrewvs.blogspot.com/usu/files/the_spacing_effect.pdf (visited on 05/11/2012).
- [7] Hermann Ebbinghaus. *Memory: a contribution to experimental psychology*. 1913. URL: <http://archive.org/details/memorycontributi00ebbiuoft> (visited on 10/16/2012).
- [8] Ethem Alpaydin. *Introduction to Machine Learning*. 2nd edition. The MIT Press, 2010.
- [9] Torstein Hnsi. *JavaScript Chart Library*. 2011. URL: <http://stackoverflow.com/a/1826573/80393> (visited on 10/26/2012).
- [10] Jeffrey D. Karpicke and Henry L. Roediger III. “The Critical Importance of Retrieval for Learning”. In: *Science* 319.5865 (2008), pp. 966–968.
- [11] Jeffrey D. Karpicke and Henry L. Roediger III. “The Power of Testing Memory: Basic Research and Implications for Educational Practice”. In: *Perspectives on Psychological Science* 1.3 (2006), pp. 181–210.
- [12] Joel R. Levin and Mary E. Levin. “Mnemonic vocabulary instruction: Additional effectiveness evidence”. In: *Contemporary Educational Psychology* 17.2 (1992), pp. 156–174.
- [13] Memrise. *About — Memrise - the fun way to learn anything*. 2012. URL: <http://www.memrise.com/about/> (visited on 11/02/2012).
- [14] Peter Bienstman. *Principles | The Mnemosyne Project*. 2012. URL: <http://www.mnemosyne-proj.org/principles.php> (visited on 08/23/2012).
- [15] Lloyd R. Peterson et al. “Effect of Spacing Presentations on Retention of a Paired Associate Over Short Intervals”. In: *Journal of Experimental Psychology* 66.2 (1963).
- [16] Scott Chacon. *Pro Git*. 1st edition. Apress, 2009.
- [17] Rachel Seabrook, Gordon D. A. Brown, and Jonathan E. Solity. “Distributed and Massed Practice: From Laboratory to Classroom”. In: *Applied Cognitive Psychology* 19.1 (2005).
- [18] Gary Wolf. “Want to Remember Everything You’ll Ever Learn? Surrender to This Algorithm”. In: *WIRED* 16.05 (Apr. 2008). URL: http://www.wired.com/medtech/health/magazine/16-05/ff_wozniak?currentPage=all (visited on 10/14/2012).
- [19] Piotr Wozniak. “Optimization of learning”. PhD thesis. University of Technology in Poznan, 1990. URL: <http://www.supermemo.com/english/ol.htm>.

- [20] John Zornig. *CEIT Project Methodology*. Jan. 2012. URL: <http://ceit.uq.edu.au/content/ceit-project-methodology> (visited on 10/20/2012).

Appendix A

Participant Information Sheet



School of Information Technology and Electrical Engineering

HEAD OF SCHOOL

Professor Paul Strooper

Jordan West

Telephone 0438518251

Email

jordan.west@uqconnect.edu.au

CRICOS PROVIDER NUMBER 000258

Participant Information Sheet

Machine Learning and Spaced Repetition Systems for Predicting Foreign Language Vocabulary Test Scores

Investigator: Jordan West, Undergraduate Engineering Honours Student, Centre for Educational Innovation in Technology

Supervisor: Dr. Mark Schulz, Associate Director, Centre for Educational Innovation in Technology

This study involves monitoring your use of learning software for memorising vocabulary for JAPN1023, from which we build a model of your vocabulary knowledge. By building such a model, we are expecting to predict with some accuracy your future score in a non-assessed vocabulary recognition quiz.

The learning software is web-based and can be accessed from any computer with a modern web browser and internet connection; including from the University library or from your home. You can use the software at your leisure; there are no minimum requirements to participate however the software will be more effective at helping you memorise vocabulary if used for at least a few minutes each day. The study will span the whole of Semester 2, 2012; however you are free to choose how often and for how long you use the software.

In order to gather data on your knowledge of words, a spaced repetition flashcard system is used. Spaced repetition is a method for memorising facts at increasing periods of time and aims to provide the most efficient method of memorisation. The software has been pre-loaded with vocabulary specifically for JAPN1023, however the potential risk of participating is that you might change your study habits to incorporate this software and miss other important content. It is recommended that you use this software as an additional tool to assist your study, and not as a replacement. Participation in this study is expected to help you memorize the vocabulary, however does not constitute a replacement for your normal class study.

Your email address will be collected when you register online, however this will only be used to allow you to login and to send a password reset email if you forget your password.

Your participation in this study is completely voluntary and will not affect your grade in JAPN1023. Participation is anonymous – you will be identified only by a unique code handed out randomly upon your consent. Data on how you use the software will be collected and stored confidentially and securely and in a form such that data cannot be linked with any individual. The teacher will have access to aggregate information on the class as a whole; however will not have access to information about individual students.

You may withdraw from the study at any time, either by logging in to the system using your code and password, or by contacting me on the details listed at the top of this page. Upon withdrawal, your account and all associated data will be deleted.

This study has been cleared by one of the human ethics committees of the University of Queensland in accordance with the National Health and Medical Research Council's guidelines. You are of course, free to discuss your participation in this study with project staff (contactable on 0438518251). If you would like to speak to an officer of the University not involved in the study, you may contact the Ethics Officer on 3365 3924.

If you have any difficulties, questions or concerns about the study, feel free to contact me.

If you would like to learn the outcome of the study in which you are participating, you can contact me at the email above or write your email on the consent form and I will send you an Abstract of the study and findings upon completion.

School of Information Technology
and Electrical Engineering

The University of Queensland
Brisbane QLD 4072 Australia

T (07) 3365 2097

F (07) 3365 4999

Email enquiries@itee.uq.edu.au

Internet www.itee.uq.edu.au



Appendix B

Participant Consent Form



School of Information Technology and Electrical Engineering

HEAD OF SCHOOL

Professor Paul Strooper

Jordan West
 Telephone 0438518251
 Email
jordan.west@uqconnect.edu.au
 CRICOS PROVIDER NUMBER 00025B

Participant Consent Form

Project Title: Machine Learning and Spaced Repetition Systems for Predicting Foreign Language Vocabulary Test Scores

Investigator: Jordan West, Undergraduate Engineering Honours Student, Centre for Educational Innovation in Technology

Supervisor: Dr. Mark Schulz, Associate Director, Centre for Educational Innovation in Technology

Participant Name: _____

- I have read and understand the Participant Information Sheet for this project.
- I understand that my participation is voluntary and that I will not receive any benefit for participating.
- I understand that I may withdraw at any time without penalty.
- I am participating with the knowledge that data will be stored securely, confidentially and anonymously.

Signature of Participant: _____ Date: _____ / _____ / _____
(Day) (Month) (Year)

If you would like to be notified of updates and outcomes of this project, please add your email address below to be added to the mailing list:

Email Address (optional): _____

Appendix C

Original Wireframes

Wireframes/Screen Layouts

Machine Learning and Spaced Repetition Systems for Predicting Foreign Language Vocabulary Test Scores

Register

Thank you for participating in this project. I hope you find this software useful in memorising vocabulary. If you would like to read more about Spaced Repetition, [see this article](#)

You can find the Participant Information Sheet with information about the study and contact details [here](#)

Participation in this project is completely voluntary and you may withdraw at any time.

To create an account and start revising vocabulary, please fill out the following information

Account Code

(This is the code written on the card you were handed after giving consent to participate)

Password

Password again

Gender:

☐ Male

☒ Female

Is English your first language?

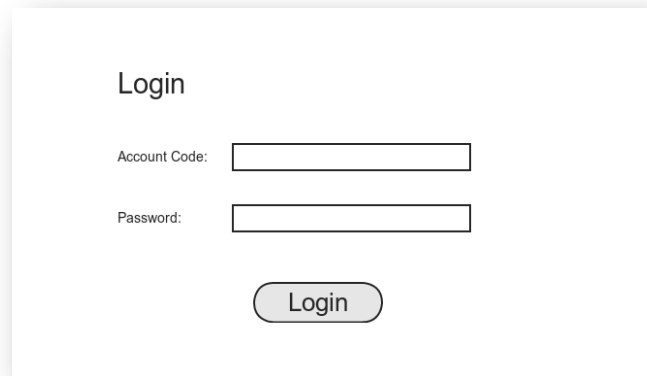
☒ Yes

☐ No

Note:
This is the only questionnaire information that will be asked of participants.

Register

Figure 1 - Registration Page. After participants have signed a consent form, they are randomly handed a card with a web address and a unique code written on it. When the participant accesses the web address, this page will be displayed requesting information from the participant.



A login form with a white background and a subtle drop shadow. It features the title "Login" at the top. Below it are two input fields: "Account Code:" and "Password:". At the bottom is a rounded "Login" button.

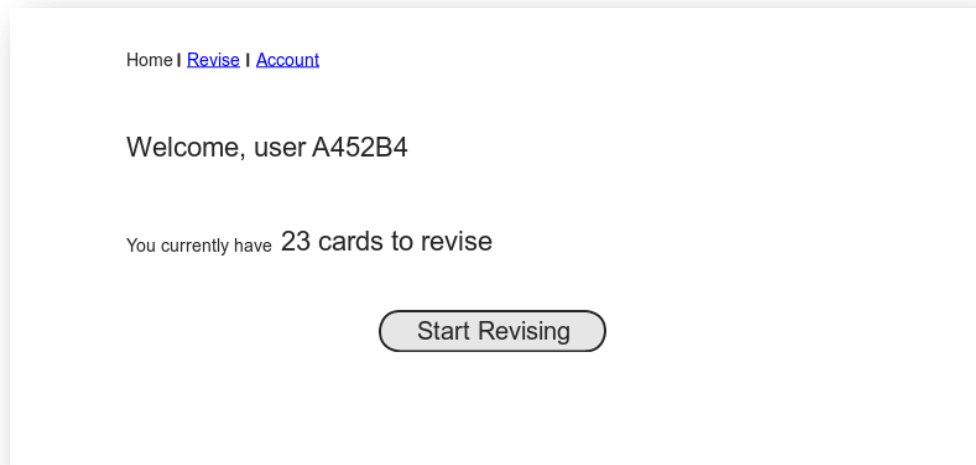
Login

Account Code:

Password:

Login

Figure 2 - Login Screen



A home page layout with a white background and a subtle drop shadow. It includes a navigation bar with links "Home | [Revise](#) | [Account](#)". Below the navigation bar is a welcome message "Welcome, user A452B4". Further down, it states "You currently have 23 cards to revise". At the bottom is a rounded "Start Revising" button.

Home | [Revise](#) | [Account](#)

Welcome, user A452B4

You currently have 23 cards to revise

Start Revising

Figure 3 - The Home Page. Displays the number of cards currently due for revision.



Figure 4 - Revision Screen 1



Figure 5 - Revision Screen 2

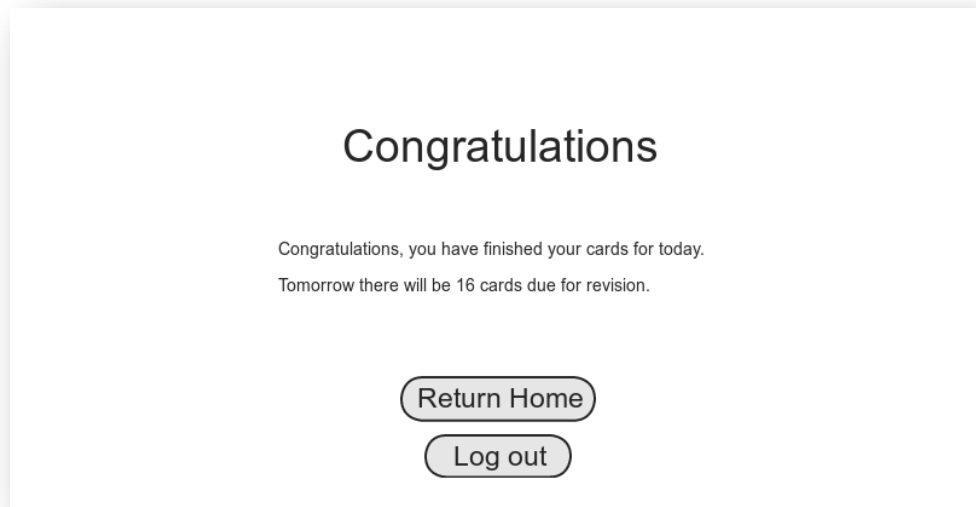


Figure 6 - Revision Complete Screen

Fill in the blanks

[Help](#)

	Japanese	English
Q1	天気予報	<input type="text"/>
Q2	季節	<input type="text"/>
Q3	<input type="text"/>	Temperature
		...
Q28	強い	<input type="text"/>
Q29	変	<input type="text"/>
Q30	<input type="text"/>	Young

Submit

Figure 7 - Quiz Screen

[Home](#) | [Revise](#) | [Account](#)

User A452B4 - Account Settings

Participation in this study is completely voluntary. Should you wish to withdraw from the study you may do so without penalty at any time by clicking the "Withdraw Participation" button below.

Note that if you do so, all data associated with your account will be immediately deleted and will not be included in the final study. You will no longer be able to log in to the system.

If you have any questions regarding withdrawal, or any difficulties using this function, please contact Jordan West at jordwest@gmail.com

Withdraw Participation

Return to Home Page

Figure 8 - Withdrawal of Participation Screen

You are about to withdraw participation from the study

All data associated with your account will be deleted and you will no longer be able to use the system.

Are you sure you want to do this?

Yes, withdraw participation

No, go back

Figure 9 - Withdrawal of Participation confirmation

Appendix D

Membit Screenshots

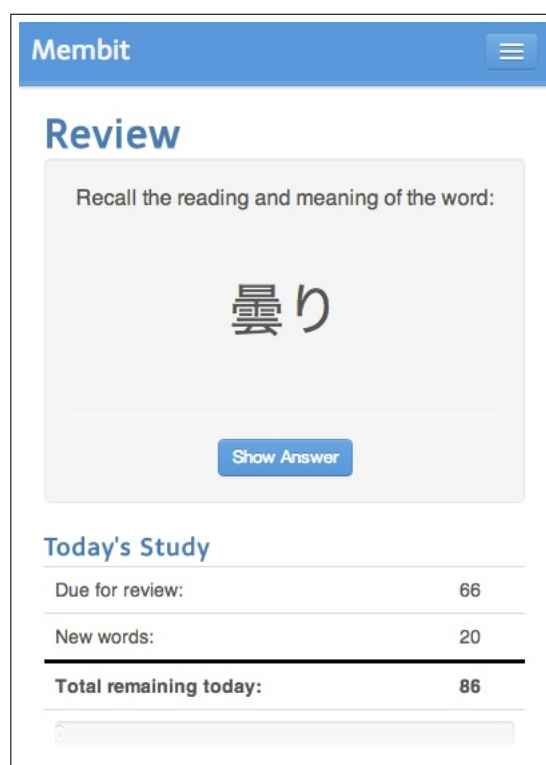


Figure D.1: Mobile review interface

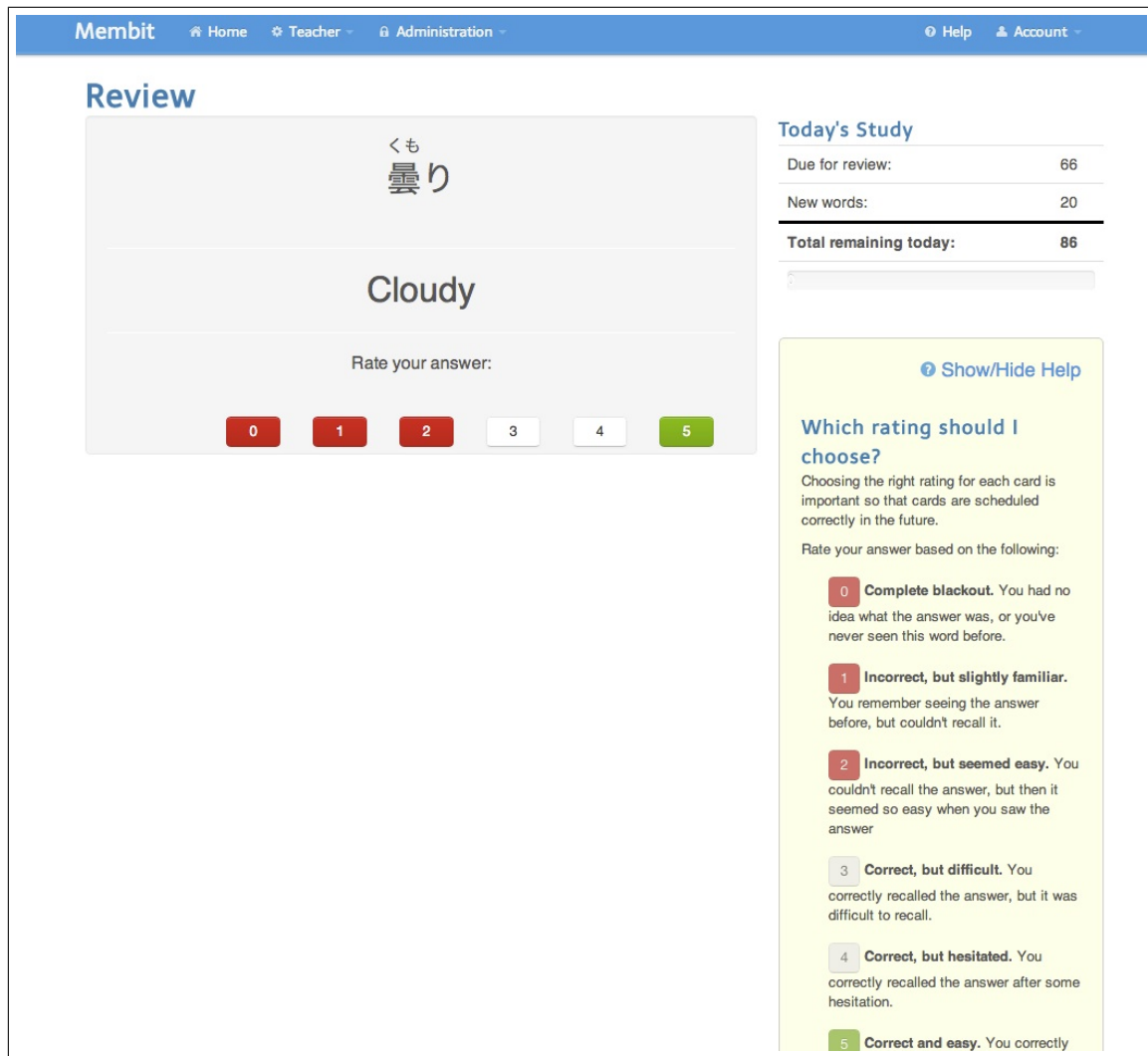


Figure D.2: Desktop review interface

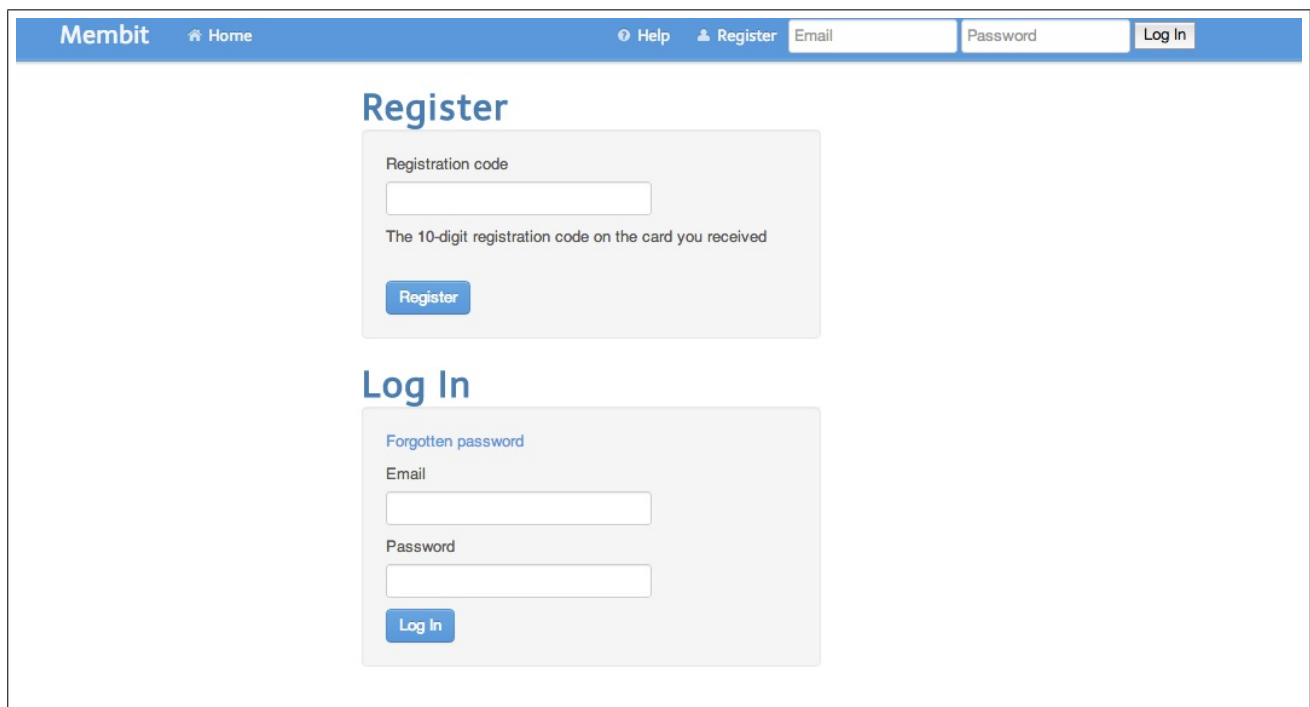


Figure D.3: Log in screen

Appendix E

Data

Data on this CD is also available at <https://github.com/jordwest/thesis>