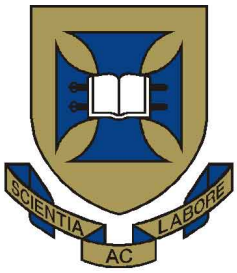


THE UNIVERSITY
OF QUEENSLAND

AUSTRALIA

Undergraduate Engineering Honours Thesis

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

*Author:*
Jordan West

*Supervisor:*
Dr. Mark Schulz

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 ofQueensland
St Lucia QLD 4072

Dear Professor Strooper, In accordance with the requirement of the Degree of Bachelor of Engineering (Honours) in the School of Information Technology and Electrical Engineering, I 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 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 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, with whom none of this would have been possible. I hope also that the students found some value in using the software.

Dr Michael Harrington of the School of Languages and Comparative Cultural Studies for his advice with the online learning software, particularly the suggestion to record the time students take to review individual words.

I would also like to thank the UQ CEIT team for their help with this project - particularly Alan Cody and 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 Mum and Dad for supporting me

# Abstract

# Contents

# Introduction

test

# Chapter 1

# Background

## 1.1 Definitions

This section outlines some terms used for convenience in this report.

**Flash-card** A method of memorising facts where a physical card is written with a question on one side and the answer on the other.

**Review** A single review of a flash-card. The user is shown a vocabulary item in Japanese and must recall the meaning and pronounciation of the word.

**Fact** A single vocabulary item with associated meaning and pronounciation.

## 1.2 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 which produces an output given certain inputs.

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 which 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 as a continuous variables (regression). Additionally, we will only be looking at supervised learning since training data will contain outputs alongside inputs.

The following sections describe the machine learning techniques that were used in this report.

### 1.2.1 Artificial Neural Network

### 1.2.2 Support Vector Machines

## 1.3 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 1.1 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.

$$R = e^{\frac{-t}{S}} \tag{1.1}$$

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.

## 1.4 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 [14]. The type of spacing however is a more controversial topic, with some studies suggesting fixed intervals are better [4], while others suggest expanding intervals are more effective [12]. 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 the how well the student recalls and the history of the flash-card, the flash-card is rescheduled after each review.

Figure 1.1 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 subsquent 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'.

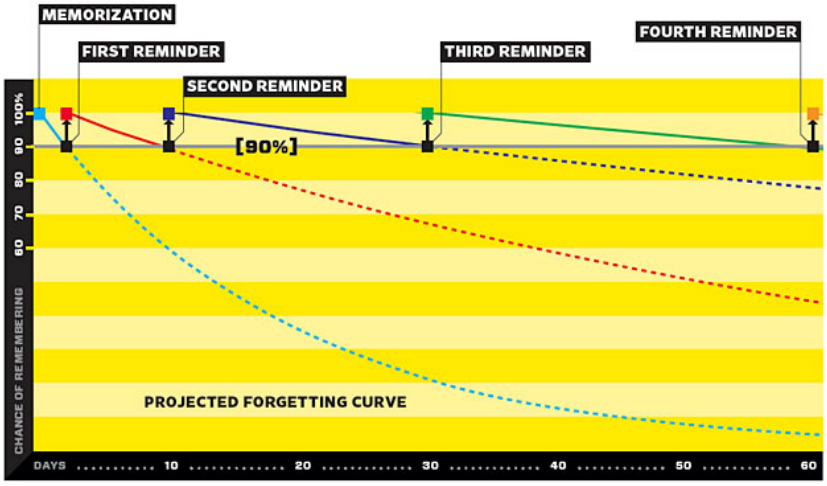


Figure 1.1: Projected Forgetting Curves with Spaced Repetitions [15]

Testing and the associated active retrieval has been shown to improve retention significantly over passive study [1], [9], [10].

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 |
| ... | ... | ... |

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. Most algorithms store the current interval (in days) which represents the spacing, 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 very little publicly available research on spaced repetition algorithms has been carried out. Dempster postulates[6] that it has simply not caught on since it has not been known for very long, has not yet been demonstrated satisfactorily with

### 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 1.5). Developed by Piotr Wozniack[16], 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[16].

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.

$$I(1) := 1 \tag{1.2}$$

$$I(2) := 6 \tag{1.3}$$

For all $n \geq 3$, equation 1.4 applies.

$$I(n) := I(n-1) \times EF \tag{1.4}$$

$EF$ 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 1.5

$$EF := EF + (0.1 - (5-q) \times (0.08 + (5-q) * 0.02)) \tag{1.5}$$

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.

## 1.5 Similar Projects

### Memrise

Memrise is a private company which produces web-based flashcard software.

### 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 [11]. Mnemosyne uses a modified version of the Supermemo algorithm. 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 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 1.2: Screenshot of Mnemosyne in use

# Chapter 2

# Goals

**Develop an online spaced repetition learning environment**

**Reproduce forgetting curves**

**Predict student responses to flash cards**

# Chapter 3

# Methods and Materials

## 3.1 Experimental Design

## 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 which 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.

## 3.3 Online Learning Software Design

### 3.3.1 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 use 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.

**Should be secure and anonymous.** Both for peace of mind for students, and to fulfill ethical requirements 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 the log in. 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.** Since the timespan is

**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**

### 3.3.2 Tools

This section outlines the software tools that will be 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 [13].

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 [17] 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 application framework [3] written in the Ruby language. Rails is designed for rapid development and provides many guidelines which 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 web applications. It was selected because...

**Highcharts** (`http://www.highcharts.com/`)

Highcharts is a commercial Javascript framework which 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 [8].

**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 3.1 for a comparison of default styling with and without Twitter Bootstrap

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

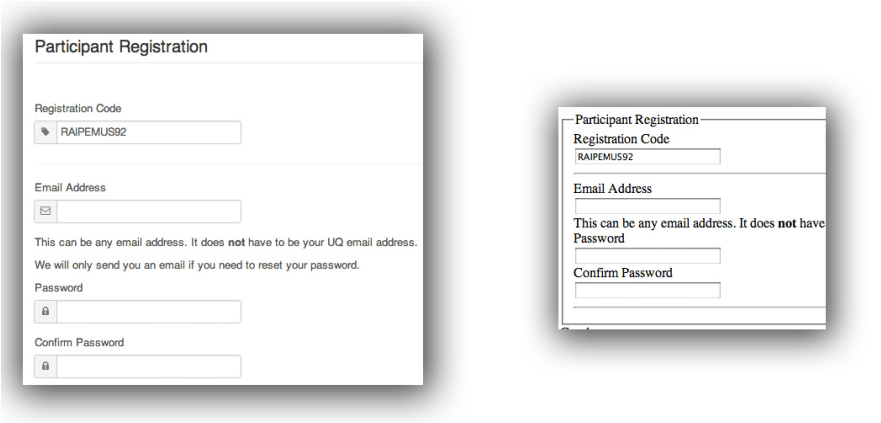


Figure 3.1: Comparison of a page with no styling and Twitter Bootstrap default styling

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 code written 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.

### 3.3.3 Data Entry

### 3.3.4 Screen Mockups

### 3.3.5 Spaced Repetition Algorithm

### 3.3.6 Data storage, formatting and output

* Insert list of tables and fields in the database

Review data is made available via an administrator login. The software converts the entire table of anonymised 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 3.1

Table 3.1: 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 |

## 3.4 Data Analysis and Prediction

### 3.4.1 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 3.1:

$$P(correct, incorrect) = \frac{\sum correct}{\sum correct + \sum incorrect} \tag{3.1}$$

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 3.2:

Table 3.2: Example of too few review samples after grouping

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

Given the sample of the three reviews in table 3.2 – equation 3.1 would yield $P = \frac{1}{1+2} = 0.333$. However the standard error as calculated with equation 3.2 yields $\sqrt{\frac{0.333(1-0.333)}{3}} = 0.272$

$$\text{Standard Error} = \sqrt{\frac{p(1-p)}{n}} \tag{3.2}$$

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 minimum number of reviews to give an adequate probability will be arbitrarily selected based on the amount of data available.

Forgetting curves can also be generated from running data through the machine learning algorithms trained upon the data. By entering a full range of possible variable values through the trained machine learning algorithms, we can generate a much larger dataset than original to work with. This will also be explored and compared with the forgetting curves generated from the true data.

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.

### 3.4.2 Prediction of Recall

The inputs shown in table 3.3 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.

Table 3.3: 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 4

# Results

## 4.1 Online Learning Environment (Membit)

### 4.1.1 Overview

### 4.1.2 Bugs and Issues

## 4.2 Usage Statistics

## 4.3 Forgetting Curves

### Generated from Recorded Reviews

Figures 4.1 to 4.4 show the review data grouped as data points by the review number and days since previous review. This considers only number of correct reviews – if a user fails a review the review number is taken back to zero.

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.


Figure 4.1: Forgetting curves produced from recorded review data with threshold $n \geq 5$


Figure 4.2: Forgetting curves produced from recorded review data with threshold $n \geq 30$

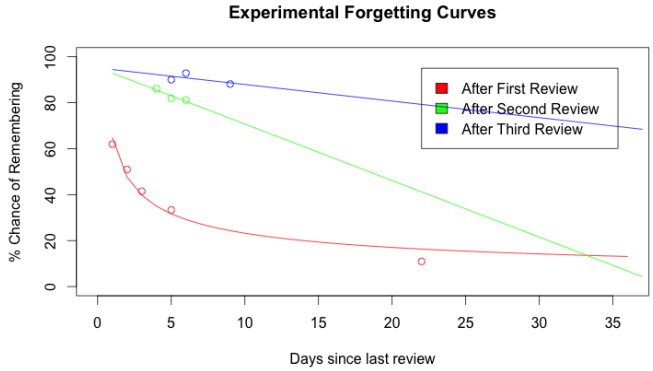


Figure 4.3: Forgetting curves produced from recorded review data with threshold $n \geq 50$

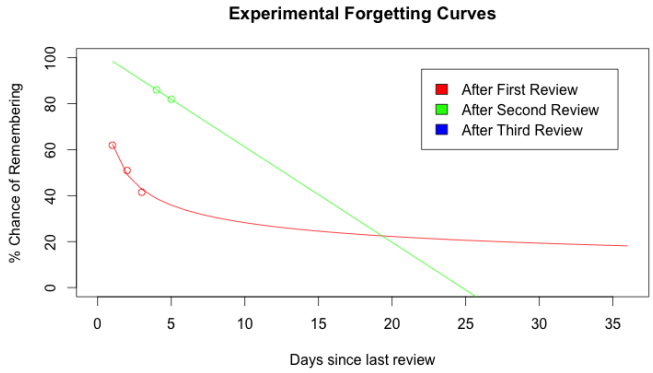


Figure 4.4: Forgetting curves produced from recorded review data with threshold $n \geq 100$

### Generated from Machine Learning Algorithms

## 4.4 Prediction of Recall

### Support Vector Machines

# Chapter 5

# Discussion

## 5.1 Changes to Original Scope

* Removed testing and scoring tests * Removed typed answers, self evaluation only

## 5.2 Evaluation

### 5.2.1 Usage of Software

### 5.2.2 Prediction of Recall

## 5.3 Potential Future Work

* Vary the reschedule date slightly from the interval so that data is available for more than just the common intervals

# 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.blogs.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] Torstein Hnsi. *JavaScript Chart Library*. 2011. URL: http://stackoverflow.com/a/1826573/80393 (visited on 10/26/2012).

[9] Jeffrey D. Karpicke and Henry L. Roediger III. "The Critical Importance of Retrieval for Learning". In: *Science* 319.5865 (2008), pp. 966–968.

[10] Jeffrey D. Karpicke and Henry L. Roediger III. "The Power of Testing Memory: Basic Research and Implications for Educational Practice". In: *Perpectives on Psychological Science* 1.3 (2006), pp. 181–210.

[11] Peter Bienstman. *Principles | The Mnemosyne Project*. 2012. URL: http://www.mnemosyne-proj.org/principles.php (visited on 08/23/2012).

[12] 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).

[13] Scott Chacon. *Pro Git*. 1st edition. Apress, 2009.

[14] 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).

[15] 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).

[16] Piotr Wozniak. "Optimization of learning". PhD thesis. University of Technology in Poznan, 1990. URL: http://www.supermemo.com/english/ol.htm.

[17] 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

THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

**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 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.

# Appendix B

# Participant Consent Form

THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

**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**

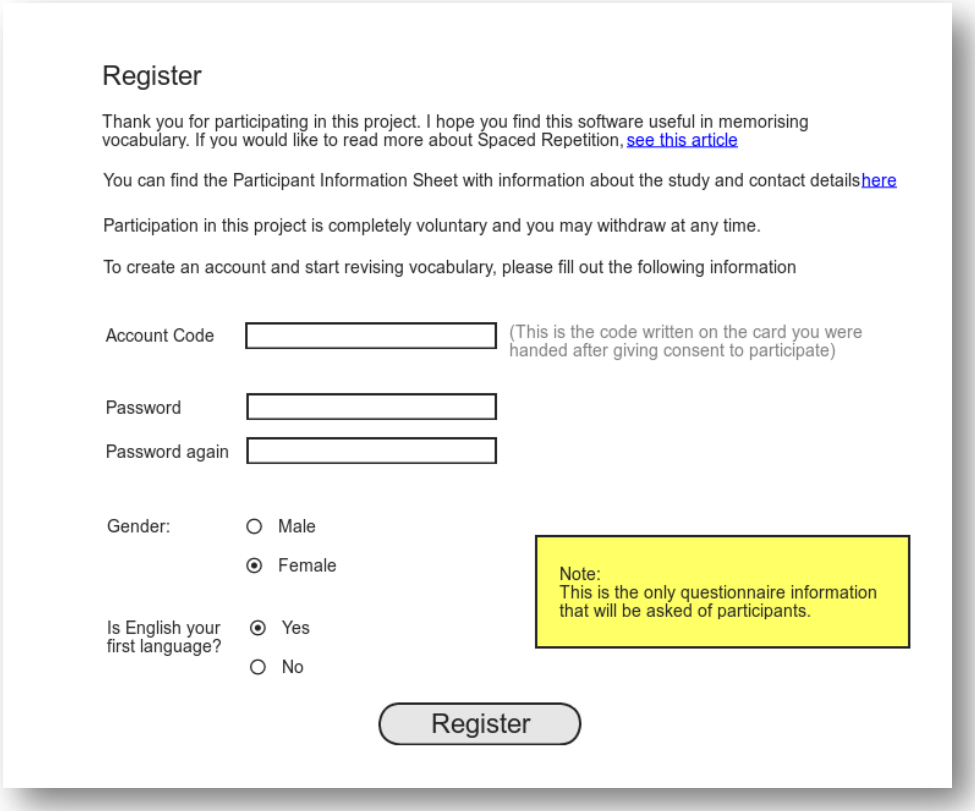


Figure 1 - Registration Page. After participants have signed a consent form, they are randomly handed a card with a web address and a unique code written on it. When the participant accesses the web address, this page will be displayed requesting information from the participant.

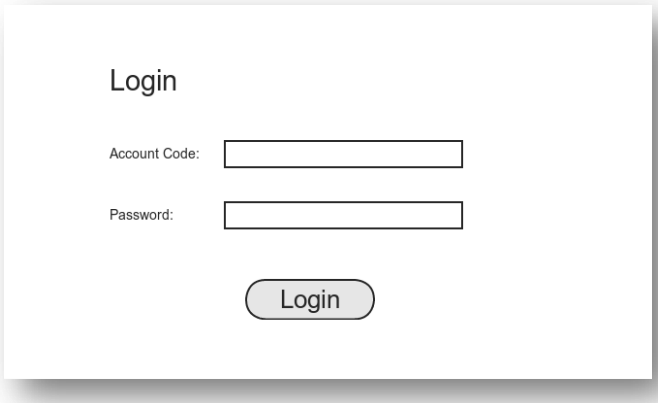


Figure 2 - Login Screen

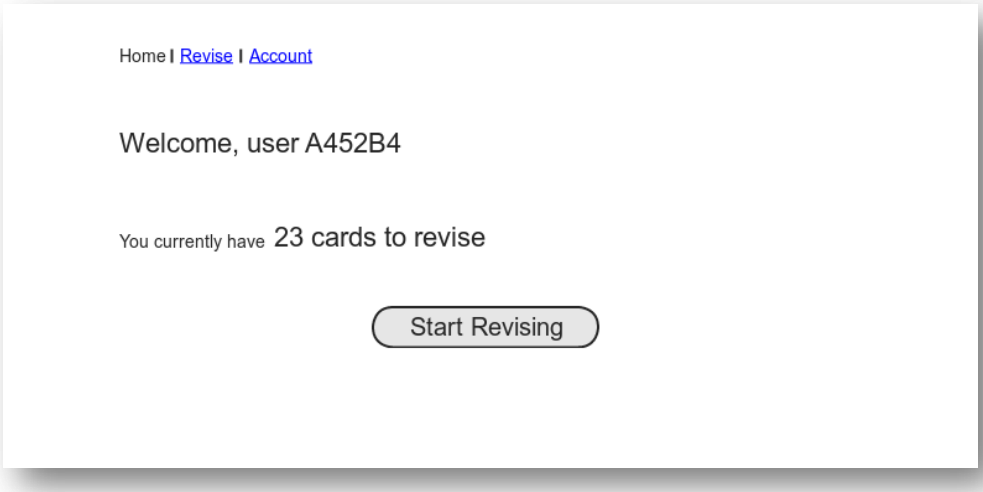


Figure 3 - The Home Page. Displays the number of cards currently due for revision.


Figure 4 - Revision Screen 1

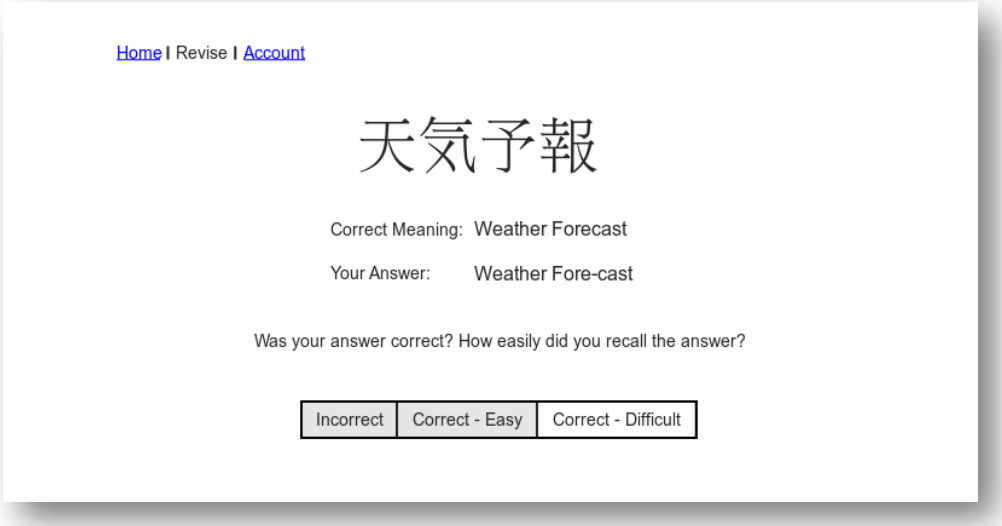


Figure 5 - Revision Screen 2

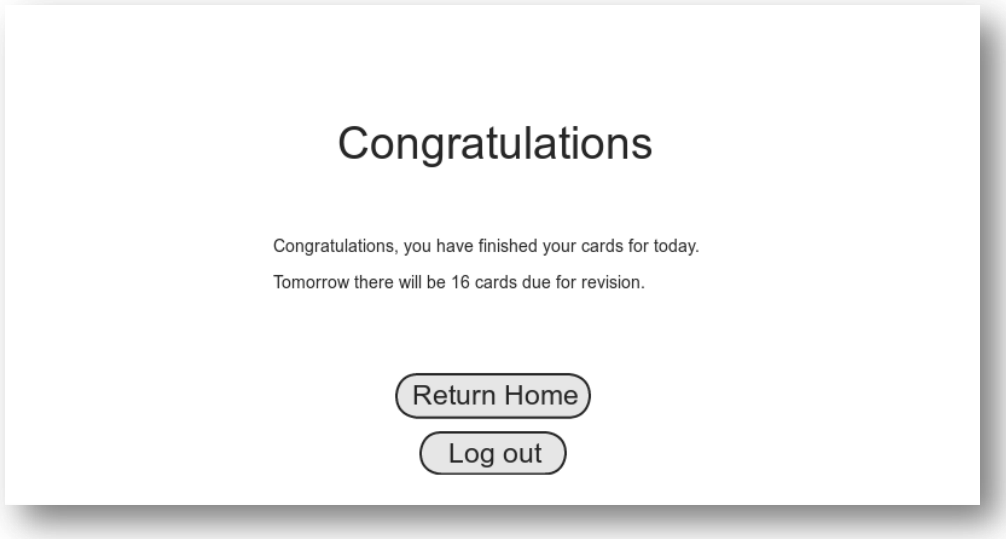


Figure 6 - Revision Complete Screen

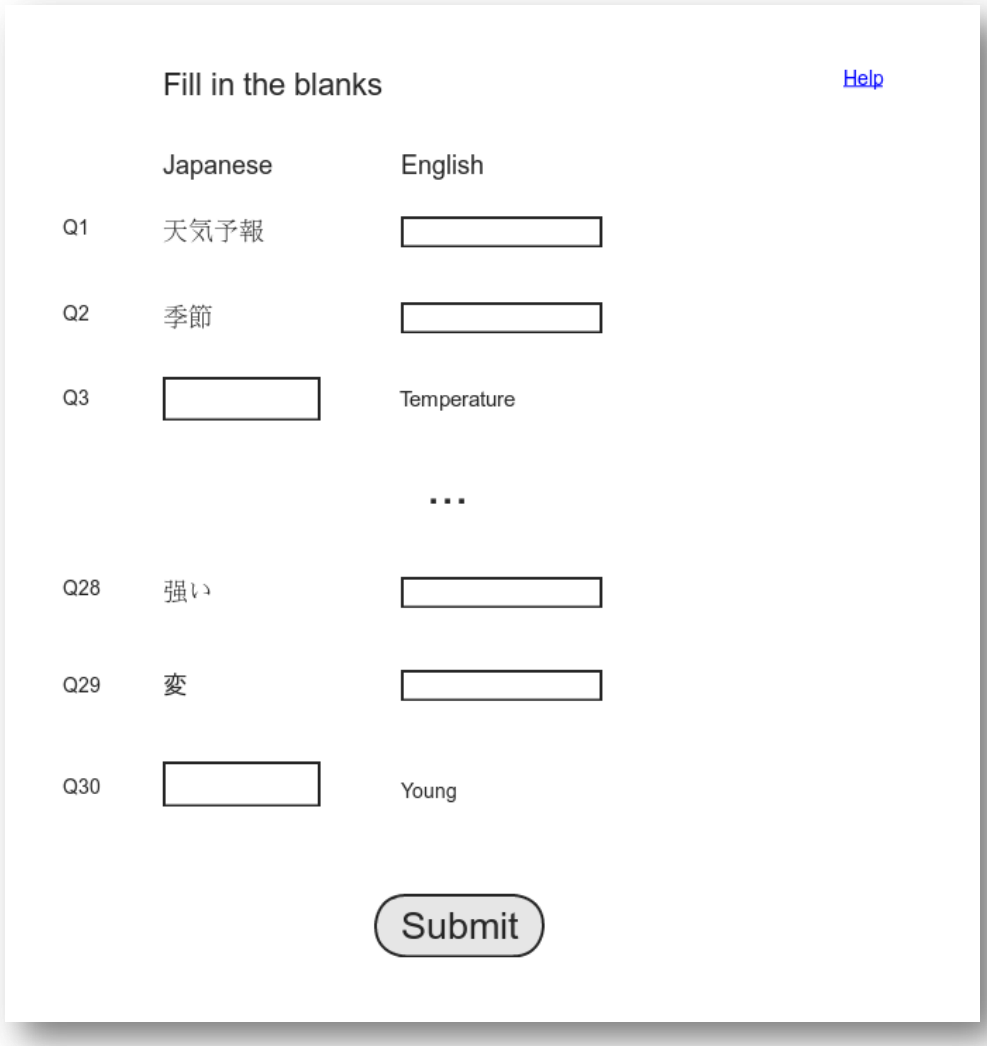


Figure 7 - Quiz Screen

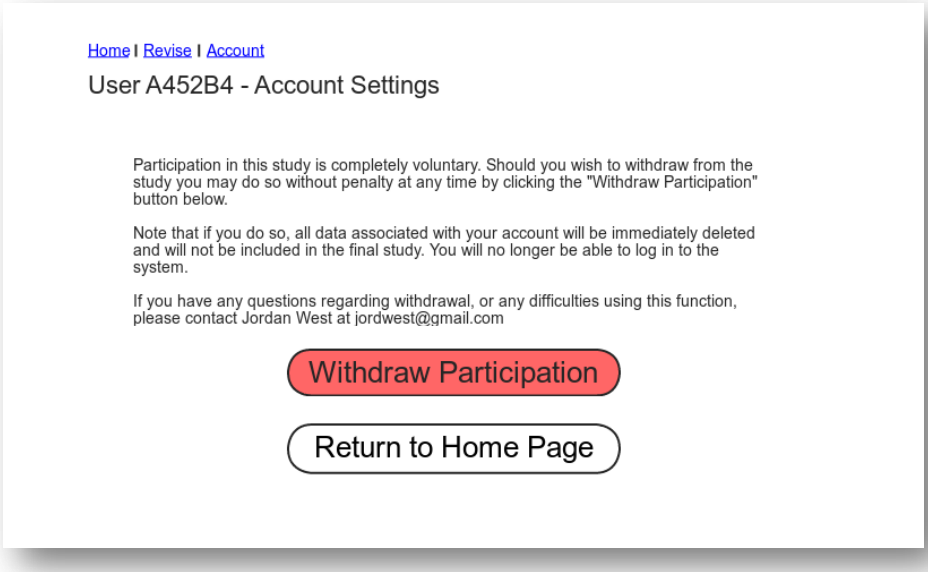


Figure 8 - Withdrawal of Participation Screen


Figure 9 - Withdrawal of Participation confirmation