Department of Creative Informatics
Graduate School of Information Science and Technology
THE UNIVERSITY OF TOKYO

Master's Thesis

Run-time estimation of the reading materials' relative difficulty

言語学習コンテンツ相対難易度の実行時推定手法

Mao Yetian
マオ イエティエン

Supervisor: Professor Chiba Shigeru

July 2018

Abstract

To recommend reading materials to users based on their language competence, the system needs to be able to estimate the difficulty of reading in correspondence to the user. The difficulty is typically estimated through a large amount of pre-test. However, it requires much effort from users, which is not applicable to a large-scale user reading applications. Our approach does not rely on the pre-test, instead, adopts relative difficulty, which is the percentage of unknown words in materials and can be estimated at runtime by monitoring user-system interactions. The original relative difficulty, which assumes that all unknown words are marked, however unintentional or miss clicks may exist in the runtime estimation. To further improve the estimation of relative difficulty, this thesis proposes weighted relative difficulty based on word importance to mitigate the effect of human errors. This thesis also devise a collaborative filtering-based algorithm to estimate the weighted relative difficulty. We evaluate our method through a case study with 328 users, and results show that estimation based on our weighted relative difficulty is more accurate than the one based on original relative difficulty.

概要

語学学習において、言語能力に基づいて読書コンテンツをユーザに推薦するためには、読書の難易度をユーザに対応させて推定することができる必要がありる。難易度は、通常、大量の事前テストによって推定される。しかし、ユーザに多大な労力を強いるため、大規模な規模の読書アプリケーションには適さない。本研究では、事前テストに頼るのではなく、相対的難易度に基づくコンテンツ推薦を行う。相対的難易度はコンテンツ中の未知語の割合であり、ユーザシステムの相互作用を監視することによって実行時に推定することが期待できる。既存研究での相対的難易度はすべての未知語がマークされていると仮定しているが、ユーザの意図しない、もしくは誤りによるクリックミスによって実行時の推定精度が低下する可能性がある。そこで、相人間の誤差の影響を軽減するために、単語の重要度に基づいて重み付けされた加重相対難易度を提案する。また、重み付けされた相対的難易度を推定するために、協調フィルタリングに基づくアルゴリズムを考案する。 提案手法は、328 人のユーザを対象としたケーススタディを通じて評価され、加重相対難易度に基づく推定は、元の相対的難易度に基づく推定よりも高精度であることを示した。

Contents

Chapter 1 Introduction 1
1.1 Motivating problem . . . 2
1.2 Solution by this thesis . . . 3
1.3 Contributions . . . 5
1.4 The structure of this thesis . . . 6

Chapter 2 Problem Settings 7

Chapter 3 Literature Review 9
3.1 Overview . . . 9
3.2 Article readability . . . 9
3.3 User language level . . . 10
3.4 Relative difficulty . . . 11
3.5 Collaborative filtering . . . 12
3.6 Computerized annotations . . . 14
3.7 Existing personalized recommendation system . . . 14
3.8 Limitation of the existing methods . . . 15

Chapter 4 Recommendation Beased on Weighed Relative Difficulty 17
4.1 Overview . . . 17
4.2 Translation annotation module . . . 21
4.3 Relative difficulty vs weighted relative difficulty . . . 21
4.4 A run-time estimator of relative difficulty . . . 23

Chapter 5 Dataset and Data Collection Process 28
5.1 Article data . . . 29
5.2 User data collection . . . 30
5.3 User data validation . . . 32

Chapter 6 Results and System Evaluation 34
6.1 Summary of the users' basic information . . . 34
6.2 Demonstration of similarity in users' relative difficulty . . . 34
6.3 Estimator evaluation . . . 37
6.4 Evaluation of the recommedation system . . . 40

Chapter 7 Conclusion 42
7.1 Limitation . . . 42
7.2 Future work . . . 42

References 44

A An example of a validated user's dataset obtained in this research 48

List of Figures

3.1 Illustration of the relative difficulty and the general difficulty 11
3.2 A typical movie rating recommedation matrix 12
3.3 The basic idea of collabrative filtering 13
3.4 The interface of translation annotation module in Kindle 14

4.1 The overall system design schmatic 20
4.2 The proposed translation annotation module vs the translation annotation module in Kindle 22
4.3 A recommedation matrix filled with the relative difficulty 23
4.4 A box plot of user's relative difficulty distribution on of each reading . . 24
4.5 Plots of relative difficulty vs user's language level measured in user's total number of unknown clickings 25
4.6 Features used in linear regression model: the user feature 0-1000 indicates the number of unknown clickings this user made in total that has a frequency between 0 and 1000, the article feature 0-1000 indicates the number of words that this articles contains with a frequency between 0 to 1000 26

5.1 Reading readiablity calculated using Flesch-Kincaid Algorithm 29
5.2 Reading readiablity calculated using Lexile Framework 29
5.3 User data collection procedure 31

6.1 Age distribution of the participants 34
6.2 Education distribution of the participants 35
6.3 The relative difficulty vs reading readability 35
6.4 Plots of every article's relative difficulty for all users 36
6.5 A typical sparsed recommendation matrix 37
6.6 The recommedation matrix in this study 37

List of Tables

3.1 The interpretation of the FRES 10

6.1 The RMSE for all the collaborative filtering algorithms 38
6.2 The RMSE of KNNWithZScore using leave-p-out cross validation 39
6.3 The coefficient of determination of the linear regression model and the best collaborative filtering model 40
6.4 The accuracy of ranking the articles using the linear regression model, Collaborative Filtering with Carver's Relative Difficulty, and Collaborative Filtering with the Weighted Relative Difficulty 41

Chapter 1

Introduction

The process of natural language learning can be mainly split into four parts, listening, speaking, reading, and writing. Among these four topics, reading is often recognized as the most important aspect [1]. To train the reading ability of individual students, and to increase the joy of language learning, a variety of readings covering different topics are often given to the students as in class or outside the class. Preparing these articles to be read by the students has become an important task in conducting language learning activities in school.

One critical problem in traditional natural language learning is the ignorance of personal factors. [2, 3] It is often the case that teachers hand out same instructional materials, especially reading articles, for all the student to study. [4] This results in students with higher language skills find the readings to be too easy while students with lower language skills find the readings to be too difficult. In either case, the too easy or too difficult material may cause students to lose interest in keep on learning the language. Researches have shown that it is not appropriate to give every student the same reading articles because every student's language competence, personal interest, and needs are quite different. [5, 6] Hence, to improve students' reading experience, it is crucial to provide personalized readings with appropriate degrees of difficulty to the learners according to their current skill level.

Computer technology has always been regarded as a viable support for meaningful educational experiences. The field of teaching/learning natural languages, as a crucial part of education, contains many applications and researches using computer technology. Shinohara and Luo both provided innovative ways of improving students' English learning experience. [7, 8] Many other computer-based language learning systems have been proposed in the past decade. Although it is demonstrated that students with different skill levels need to receive different educational materials, most of the online learning platforms and existing researches provide fixed teaching materials to all the users.

Among the very few researches which have studied on how to provide the individual user with different language skill level with different learning reading materials, they all proposed the idea of a system that can provide personalized article recommendation based on user's language skill level. To be able to recommend articles with the proper degree of difficulty to each user, the system requires crucial inputs from both users and articles, which can be described as two questions: how good the user is and how hard the article is. With knowing of the level of both side: the user language level and reading difficulty level, the system can finally be able to match a user with specific level to readings with a specific difficulty level.

Kuo [9] developed a system that can make personalized recommendations based on both user's profile and the difficulty of the readings. In Kuo's research, he adopted the idea of Input hypothesis [10] which states that when a learner is at stage I, then acquisition

occurs when a learner is exposed to comprehensible input that belongs to the level I + 1. Kuo would like to find the reading material that is in the category of level I + 1 for each individual user so that every time a student reads an article, he/she will be able to acquire knowledge. To be able to make such recommendation system, Kuo needs to answer the two questions proposed previous: how good the user is and how hard the article is. The way that Kuo determines each user's skill level is using user's own writing in the second language that they are learning as input; the way that Kuo determines each article's reading difficulty is through lexical and grammatical analysis to find features in the text such as sentence length and word length. Then KNN (k-nearest-neighbor) and Naive Bayes model are used to match a user with specific language level with articles that belong to the category of level I + 1.

1.1 Motivating problem

As revealed in Kuo's research, many of the personalized article recommendation system [11, 12, 9] based on user language level requires users to manually construct their user profile through procedures such as pre-tests before they can use the recommendation system itself. According to Liu, this kind of approach places an extra burden on users, something very few are willing to take on. It is possible that Kuo's system will be suitable to environments where user's own writing is easily accessible such as at schools. However, when it comes to an online large scaled language learning system such as Newsela [13], most of the existing research won't be very helpful due to the fact that complicated procedures before users can start using the application cause bad user experience which leads to losing interested potential users.

Using a fixed pre-test as user profile also causes a second problem which is forever the same user profile. In a real-world situation, learner's language skill level will decrease or increase depending on many factors such as how much the learner has studied or how familiar learner is with specific topics. However, only obtaining the user profile once at the beginning of the use of the system cannot observe such change in user's profile, which will lead to a situation that learner has improved quite a lot, but the system still recognizes him/her as the level at the very beginning and only recommend materials that are too easy for him/her now.

Specifically to Kuo's study, input hypothesis [10] does propose a reasonable foundation for making personalized recommendations based on user language skill level. However, the level system including level I, level I + 1, and level I + 2, suggested in the input hypothesis is very ambiguous. Concept-wise, It is easy to understand that level i is the current user language level; level I + 1 is the level that slightly higher than what user is currently capable of; level I + 2 is the level that is too hard for the user to understand. The variable "i" is a very clear concept, but +1 and +2 are very unclear. it is very hard to decide the +1 for each individual user.

In summary, in this study, three problems towards the previous studies have been proposed:

- Using pre-test to determine user language level is not applicable to large scaled online user learning system due to its user-unfriendly design;
- Using pre-test to determine user language level will cause the user profile to forever stay the same, which will cause inappropriate recommendation later on when the user language level changes;
- Specifically, towards Kuo's research, the measure called input hypothesis used to

categorize the distance between a reading and a user is very ambiguous; a better measure is needed.

Towards the three existing problems, several research questions were proposed as follow:

- What's a good metric for distance between a user and a reading?
- How to achieve automation in constructing user profile
- How accurate is the proposed estimator at predicting relative difficulty?
- Can proposed system be able to adapt to user's profile change?
- How good is the proposed recommendation system?

1.2 Solution by this thesis

To address the problems listed above, this paper suggests a personalized recommendation system that collects user profile at run-time by tracking user's interaction.

What's a good metric for distance between a user and a reading?

The recommendation system proposed in this paper adopts a concept called Relative Difficulty from Carver [14] stating that there is a relative difficulty between one specific reading to one specific user, and one way of measuring relative difficulty is through the percentage of unknown words that a user has towards a specific reading. The relative difficulty is used in this thesis instead of input hypothesis as a measure to decide the difficulty of a reading corresponding to a user due to the reason that relative difficulty is a much more concrete concept.

Based on our literature review, it is possible that this thesis is the first study that utilizes the idea of relative difficulty as the base to provide personalized article recommendations to users. Because of being the pioneer in such a method, there doesn't exist any public datasets that provide relevant data for calculating the relative difficulty for each user. Therefore a responsive web application is developed for this study to collect user data for relative difficulty calculation. Originally in Carver's study, the relative difficulty [14] is measured through experiments where each participant intentionally circles out all the unknown words. To combine this circle action into the modern online web application, a translation annotation module is developed and added to the recommender system. The translation annotation module allows a user to click on an unknown word for a dictionary translation. By tracking the clicks that each user has performed, the system is able to calculate the relative difficulty, which the percentage of unknown words of a reading corresponding to a user. Later in the study, a problem with Carver's definition of relative difficult is revealed. Human errors exist in an online system. Sometimes a user will misclick on a known word; and sometimes even though a word is unknown, a user may decide not to see its translation due to reasons such as laziness or the word doesn't affect the understanding of the article. To mitigate this issue, this study proposed a weighted relative difficulty. In Carver's definition, every unknown word weights the same. In the weighted relative difficulty, word frequency is used; an unknown word with a higher frequency such as "the" will weigh less while word with a lower frequency will weigh more. Both weighted relative difficulty and Carver's relative difficulty are solid measures for the distance between a reading and a user.

How to achieve automation in constructing the user profile?

The proposed system constantly monitors users' interaction with the system through the translation annotation module. Every time the translation annotation module is triggered by the user, his/her profile is created or updated automatically. These unknown clickings performed by the users were later converted automatically into the relatived difficulty.

How accurate is the proposed estimator at predicting relative difficulty?

By inviting users to participate in the data collection phase, the final dataset collected includes 269 users reading over 7 different articles with different level of difficulty. The recommendation system in this study labels the relative difficulty of each reading according to each user's current language level. One important thing to be achieved is estimating the relative difficulty of each reading according to each user. In total, 9 different algorithms were evaluated on the same datasets. To evaluate which algorithms estimated the relative difficulty the best on the collect dataset, RMSE (Root Mean Squared Error) and coefficient of determination were used as metrics. Through comparison, a simple linear regression model outperformed all the collaborative filtering based algorithms.

Can proposed system be able to adapt to user's profile change?

To show that the proposed recommendation system is capable of changing according to the change of user, the leave-p-out cross-validation [15] is used to check how the estimator will predict when only one reading's relative difficulty is given as input vs when six readings' relative difficulty is given as input. The result shows that the more reading is used as input, the lower the RMSE get indicating that proposed system is capable of changing according to the change of user profile.

How good is the proposed recommendation system?

To evaluate how good the recommendation system is, many test subjects were invited to read the articles and rank them based on difficulty. The recommendation system was programmed to rank the readings based on relative difficulty as well. The result of the ranking generated by the recommendation system and actual ranking were compared. The results showed that a recommendation system with KNN-based estimator using weighted relative difficulty outperforms the system using Carver's relative difficulty.

1.3 Contributions

The contributions of this study is as follow:

Proposed a robust run-time estimator for relative difficulty

By adopting and improving the concept from previous researches, specifically the relative difficulty from Carver, the proposed system is capable of estimating the distance between an unique user and an unique reading.

A recommendation system based on the run-time estimator

The relative difficulty of the readings in the database are estimated based on user's interaction with the translation annotation module. The system is capable of recommending articles to users based on their language level. Every reading displayed to the users is lebaled with difficulty that is specific to their current language level.

Various attempts on different estimators to find the best one

In total, 9 different data models are evaluated as the estimator for the relative difficulty. The RMSE of each estimator shows that the linear regression outperforms the other eight algorithms.

Dataset Collection

This study presented a new dataset that doesn't exist publically at the moment. The dataset includes 269 participants read over 7 different articles using the translation annotation module that is proposed in this study. By using the features collected from the translation annotation module, Carver's relative difficulty and proposed weighted relative difficulty are calculated. The dataset also includes the time length that each user takes on reading each article and each user's personally rated reading difficulty.

1.4 The structure of this thesis

From the next chapter, we presented the problem setting of this research, the existing studies and their limitations, the proposed method, and evaluations of the proposed method. The specific structure of the rest of this thesis is as follows.

Chapter 2: Problem Settings

In this chapter, the problem settings of this study is illustrated

Chapter 3: Literature Review

In this chapter, an extensive literature review is presented.

Chapter 4: Recommendation Based on Weighted Relatvie Difficulty

In this chapter, we proposed our system design including the translation annotation module, the relative difficulty estimator, and the recommendation system.

Chapter 5: Dataset and Data Collection Process

In this chapter, we describe how we collected the dataset and how these data were processed and filtered before getting the final data that is input into the evaluation phase.

Chapter 6: Results and System Evaluation

In this chapter, we illustrated how the proposed method is evaluated by answering the three research questions as follows:

1. What is a good metric for measuring the distance between a user and a reading?
2. How can a personalized article recommendation system based on user language level automatically create and update user profile?
3. How accurate is the proposed estimator at predicting relative difficulty?
4. Can proposed system be able to adapt to user's profile change?
5. How good is the proposed recommendation system?

Chapter 7: Conclusion

Finally, we conclude this thesis in this chapter. The limitation and future work are also presented in this chapter.

Chapter 2

Problem Settings

As the society of IOT (internet of things) develops, many learning platforms shifted from offline classrooms such as schools or tutoring centers to online educational platforms such as Coursera [16], Udemy [17], and Teachable [18]. Second language learning is a major part of education curriculum. The process of natural language learning can be mainly split into four parts, listening, speaking, reading, and writing. Among these four topics, reading is often recognized as the most important aspect [1]. In the traditional settings such as in school or one-to-many tutoring, same second language reading materials are given to all students despite that every student's profile including learning habit, current language skill level, and etc is very different.

To solve this problem, many researchers proposed the idea called personalized article recommendation [5, 4, 19, 20, 9]. Most of the existing researches were carried out in the traditional settings which are inside the school classroom [5, 19, 9]. These studies mainly focus on how to recommend articles to students at school and their methods are restricted in this traditional specific settings. Majority of the existing work proposes to use complicated pre-tests to obtain the user profile and to have language experts to label the reading in the databases [11]. These kinds of actions are only allowed in the traditional settings due to the small user scale. However, when it comes to personalized recommendations based upon user language level in a large-scale online user application, many methods proposed in the existing studies will not be applicable.

In most of the online news or article applications, recommendation system is a necessity to filter out relative information to specific users among the overwhelming articles that exist on the internet. These traditional reading recommendation systems usually focus merely on recommending readings that suit user's interest or based upon the current trend such as popularity and big events. However, for second language learners using these traditional online article application, the title of the reading that is recommended to these learners could be very attractive, but the body of the reading could be linguistically too difficult for these learners to understand, which causes them to dislike the recommendation and stop reading. So a simple question to be answered is that:

Are the current online article recommendation systems suitable to users reading articles in their second language?

In this study, instead of following the existing studies which focus on solving the personalized recommendation problem in the traditional settings, we decided to look at the problem from a different perspective. Because of development of IOT, many education systems have been slowly shifted from offline to online [16, 17, 18]. Newsela [13] is an online article application that is specifically designed for English learners. There exist many more online reading applications for language learning just like Newsela. In this study, we decided to carry out our research in an online large scaled user application problem

settings and we sought a method that is suitable to the proposed problem settings unlike any of the existing methods.

Chapter 3

Literature Review

3.1 Overview

There are numberless applications that assist learner studying natural languages. Most of the applications focus on helping individuals study a second language such as English and Japanese using their first language. Each application uses different technologies to assist language learning. Shinohara did research on using audiovisual training to correct Japanese English pronunciation [7] and Luo did research on using dubbing practice to train English learners' speaking skill [8]. However, when it comes to reading recommendations based on user language level, there only exists a very limited amount of previous studies. To be able to recommend articles based upon user language level, the system needs to be capable of finding out the distance between the article and the user, more specifically how difficult the article is in correspondence to each user's language level. One way of finding out the distance between the user and the article is to determine the article readability and user language level separately. The other is to find out the distance directly using measures such as relative difficulty [14]. The following of this chapter is organized as follows:

1. Background information and terminology will be introduced in section Article Readability, User Language Level, Relative Difficulty, and Computerized annotations.
2. Studies that have researched on a similar topic as the one proposed in this thesis are provided in section Existing Personalized Recommendation.
3. Finally, the limitations of the existing work will be explained in section Limitation of the existing methods

3.2 Article readability

Reading readability (also called difficulty) is often used to estimate the linguistic complexity of texts or sentences, so that language learners can choose proper learning materials. One of the most fundamental obstacles in natural language processing is how to assess reading readability. Heilman [6] described reading difficulty as a function that is capable of mapping a reading material to a numerical value corresponding to a difficulty or grade level. In most of the studies, researchers extracted series of lexical and grammatical features (such as text length, word frequency, and grammatical complexity) from a document acting as the inputs of this function, and it outputs a numerical value which describes the difficulty of the document.

The issue of reading difficulty has been studied by many researchers who have applied various lexical and grammatical features in statistic models to analyze this problem. The very early studies such as the Dale-Chall readability formula [21, 22], the Flesch-Kincaid

Table 3.1. The interpretation of the FRES

Score	School level	Notes
100.00-90.00	5th grade	Very easy to read.
90.00-80.00	6th grade	Easy to read.
80.00-70.00	7th grade	Fairly easy to read.
70.00-60.00	8th & 9th grade	Plain English.
60.00-50.00	10th to 12th grade	Fairly difficult to read.
50.00-30.00	college	Difficult to read.
30.00-0.00	college graduate	Very difficult to read.

measures [23], and the Lexile Framework [24] only used simple lexical and grammatical features as input and developed a regression model to predict the reading difficulty levels. By adopting more complicated features, Schwarm and Ostendorf [10] demonstrated that their model can significantly increase the performance of readability prediction. While all the above studies are designed for native language readers, Huang [25] proposed a system that adopts features specifically for second language learner in Taiwan, which outperforms previous models in the situation of non-native speakers learning English. Most recent researches started to adopt machine learning models to measure the reading difficult, such as Ildiko and Sowmya's work [26].

Flesch-Kincaid score and Lexile score are the two primary metrics used in this study to determine the articles' readability. Equation (3.1) shows the calculation of Flesch reading-ease score (FRES).

$$FRES = 206.835 - 1.015\frac{total\,words}{total\,sentences} - 84.6\frac{total\,syllables}{total\,words} \tag{3.1}$$

Flesch reading-ease score can be interpreted as shown in the Table 3.1 below [27]. In general, a lower flesch reading-ease score indicates a higher readability of an article.

Lexile Framework on the other hand is not an open-sourced program. The Lexile Analyzer [28] is used to find the Lexile score of all the readings in the database. In general, the Lexile score increases as the reading's readability increases.

3.3 User language level

User language level is a measurement used to define the language fluency of users. Traditionally, user language level is measured through numerous tests such as TOEFL, TOEIC for English language level, and N1 for Japanese language level. However, traditional means of measuring user language level is certainly not applicable to user applications since it will require all the users in the application to take certain tests before they can actually use the application. Numerous researches have devoted to determining user language level through simple tasks. Kise [29] proposed a method of using the eye tracking devices to analyze user's eye movement while doing readings to predict user's language level. Hwang [12, 11] designed a search engine system that tracks students' search keyword to build a knowledge base of each student in order to measure the user language level. Kuo [9] conducted a research which takes user's own writings as input and extracts lexical and grammatical features from these writings to measure the user language level.

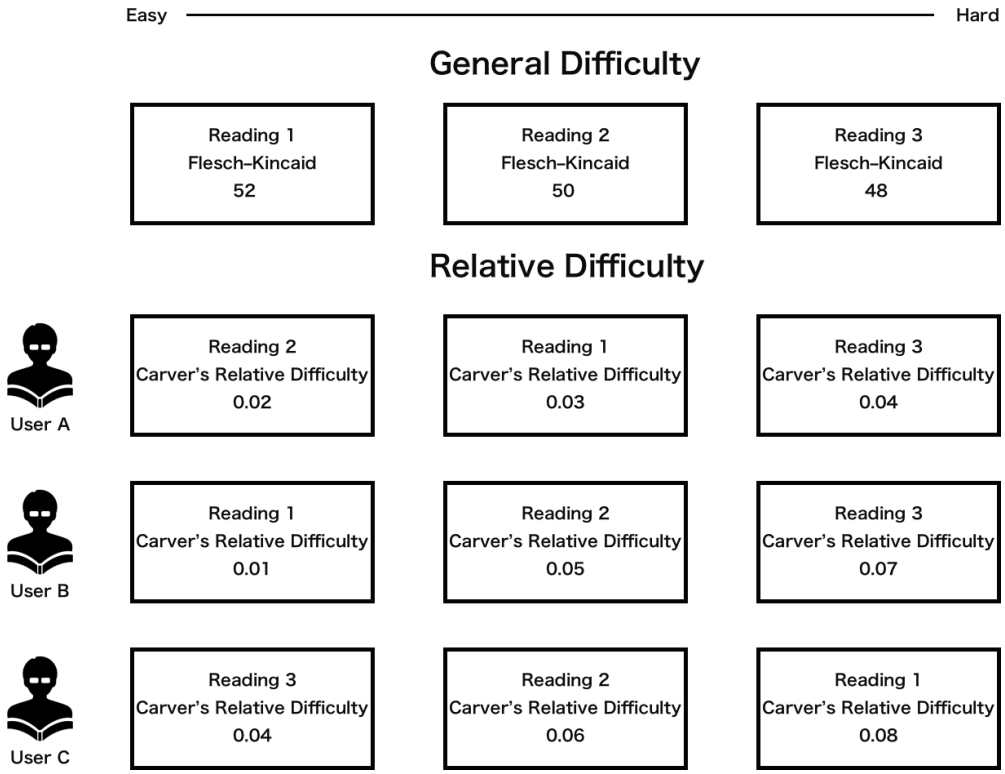


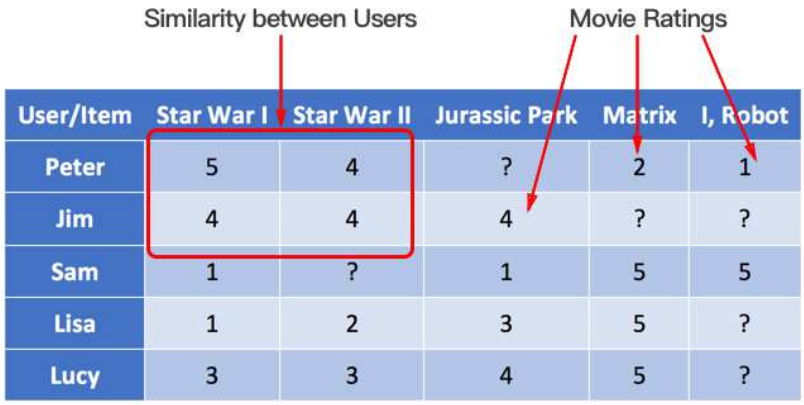
Fig. 3.1. Illustration of the relative difficulty and the general difficulty

3.4 Relative difficulty

The relative difficulty is a terminology that is used throughout this paper. If we call the reading difficulty measurements such as Dale-Chall readability formula [21, 22], the Flesch-Kincaid measures [23], and the Lexile Framework [24] as general difficulty meaning that all these measures provide an average hardness of the reading towards an average linguistically skilled individual, relative difficulty is the measure that illustrates the difficulty of a specific reading corresponding to a specific user [14].

Because every user's language skill is different and everyone has a different knowledge base (such as knowing specific terminologies in the medical field), one specific reading's readability should be different for each user. For instance, a user with a medical background can easily understand a medical-related article. But It will be much harder for a non-medical background individual to understand the same article. But this difficulty difference between these two individuals will not be shown if we only look at the readability measures of the reading.

Carver [14] also approves the idea of relative difficulty and proposed a definition of relative difficulty which is measured by the percentage of unknown words that the user has towards one specific reading. This study adopts Carver's definition of relative difficulty as the measurement of the distance between a specific user to a specific reading. Carver's definition of relative difficulty will be used as the basis for the recommendation model that we will propose later in this paper. Equation (3.2) shows Carver's definition of Relative Difficulty.



User/Item	Star War I	Star War II	Jurassic Park	Matrix	I, Robot
Peter	5	4	?	2	1
Jim	4	4	4	?	?
Sam	1	?	1	5	5
Lisa	1	2	3	5	?
Lucy	3	3	4	5	?

Fig. 3.2. A typical movie rating recommedation matrix

$$Carver's\ Relative\ Difficulty = \frac{Number\ of\ unknown\ vocabulary}{Number\ of\ total\ unique\ vocabulary\ in\ the\ reading} \tag{3.2}$$

Figure 3.1 is an illustration of the difference between the relative difficulty and the general difficulty. The general difficulty of a reading is always fixed if the same readability measure is used, because all the features that used to determine the general difficulty is within the content of the reading itself. Therefore, the ranking of the articles based on the general difficulty is always the same for every user. On the other hand, the relative difficulty takes in features from both the reading and the user as shown in Equation (3.2), which leads to a more personalized measure of the difficulty of each reading. Hence, the ranking by relative difficulty changes based on each user language level.

3.5 Collaborative filtering

Collaborative filtering is a method often used by recommendation systems [30]. It is a process of data filtering to find specific information, patterns, or similarities using techniques involving collaboration among multiple users. Typical collaborative filtering system will construct a 2-d recommendation matrix based upon all users' profiles. Figure 3.2 is an example of a rating recommendation matrix, where each column indicates a different user and each row indicates a different item. So the cell at column A and row W means user A gives item W a rating of 4. Each column in the matrix is a user profile vector whereas each row in the matrix is an item profile vector. By using similarity measures such as Cosine similarity or Euclidean distance [40] between these vectors, collaborative filtering is capable of finding similar users and predict the ratings of unknown items. Formula (3.3) shows the equation of consine similarity whereas Formula (3.4) shows the equation of Euclidean distance.

$$Cosine\ Similarity = cos(x, y) = \frac{x \cdot y}{||x|| \cdot ||y||} \tag{3.3}$$

$$Euclidean\ Distance = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{3.4}$$

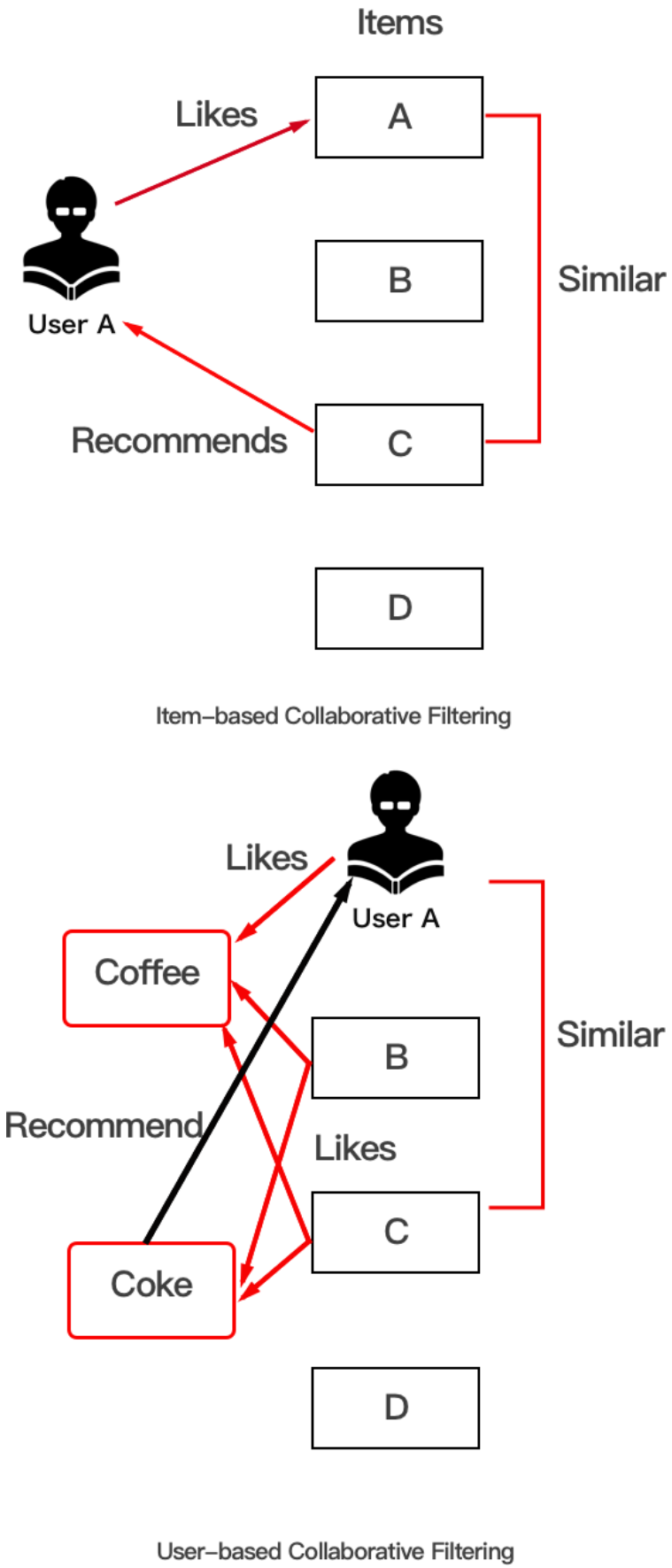


Fig. 3.3. The basic idea of collabrative filtering

Fig. 3.4. The interface of translation annotation module in Kindle

3.6 Computerized annotations

The advancement of network and computer technologies encourages users nowadays to carry out reading activities on the web or in electronic form [31]. Many webs or electronic text readers provide users with annotation functionalities, such as notes annotation (such as Adobe's pdf expert) [32] or translation annotation (such as amazon's kindle) [19]. Scholars have found that using annotation while reading can reduce the cognitive load of students, which can ease the understanding of the reading material [33].

Figure 3.4 shows the interface of Amazon Kindle's translation annotation module.

Several studies suggested that students who make annotations while reading tend to outperform those who do not take any action while doing reading [34]. Quade [35] conducted a study that demonstrated that students tend to achieve the better learning outcomes while using computers to take notes than using pen and paper. This study installs a translation annotation module into the recommendation system that simulates the translation annotation process of second language learner when running into unknown vocabulary while doing readings. Users can simply touch on the word that they are not familiar with, the system will provide translations to their native language.

3.7 Existing personalized recommendation system

Most of the online learning applications provide same teaching materials to all users, however, towards different users, the same materials can be too easy or too hard [36]. To

solve this issue, researchers have developed various recommendation systems or adaptive learning systems that are capable of providing personalized learning materials to individual students by analyzing their profiles or learning portfolios [19, 9]. For example, CAMLES [37] is a personalized context-aware mobile learning application for supporting students to learn English as a foreign language in order to prepare for TOEFL test. The system provides adaptive content for different learners based upon individual context. In their model, the individual context includes many parameters such as location (places that the study takes place, i.e. restaurant), time (what time of the day the study takes place), manner (i.e., concentration, interest level) as well as learner's knowledge. By analyzing the parameters above combining with the answers to the questions at the end of each study, the system adaptively pushes appropriate material to the users.

The University of Rijeka in Croatia is also currently developing an adaptive e-learning system for language learning on a web platform. [38] As shown in their proposed adaptive language e-learning system, they developed a system that constantly evaluates learners' skill and providing suitable educational content based upon individual language competence.

In the meantime, several learning systems specifically designed for language learning with recommendation mechanisms have been developed. For example, Kuo [9] has adopted the idea of Input hypothesis [39] stating that when a learner is at stage I, then acquisition occurs when a learner is exposed to comprehensible input that belongs to the level I + 1. In his research, he extracted features such as vocabulary difficulty and sentence length from each article and classify them into three category based upon text similarity using the above features: Level I (The level that learner currently belongs to), Level I + 1 (The level that learner needs to read at to acquire knowledge), and Level I + 2 (The level that is too difficult for the user). To determine each user's language level I, the system takes user's writings as input and uses machine learning method kNN and Naive Bayes to predict the difficulty level of the readings.

3.8 Limitation of the existing methods

There are many limitations to the existing methods:

Not applicable to a large-scale user application

Most of the existing researches were carried out in the traditional settings which are inside the school classroom from elementary school class to college class. All the methods share a limitation which is restricted to the small-scale user environment. Majority of the existing work proposes to use complicated pre-tests to obtain the user profile and to have language experts to label the reading in the databases [9, 11]. These heavy prerequisites will only work in school because students are very obedient to teachers' request. However, if the same method is applied to an online large scaled user application, the heavy prerequisites of the existing method will stop many interested users from trying the application due to the bad user experience. Only very few users are willing to take on heavy pre-test before they can use the application.

Not capable of updating user profile change later on

Existing work only collects user's profile once at the beginning of their experiments by asking users to manually input their information through pre-test or similar. The user profile in their system never changes after the first collection. This means that the system is not capable of adapting to user's change in language level and recommend items

according to these changes. However, users language level does change a lot especially for second language learners who try to absorb as much knowledge as they can every day. Most of the existing methods lack the ability to automatically create and update user's profile and adapt to user's change in language ability later on.

No accurate metric to determine the distance between a user and a reading

Most of the existing work tried to find out both user language level and articles readability to match a user with specific language level to an article with specific difficulty. However, regular way of determining the difficulty of the reading only depends on the features of the article itself. It ignores the feature of the user. A reading of difficulty 10 could be as easy as 1 for a native speaker while a reading of difficulty 1 could be as hard as 10 for a beginner. As Carver suggested, there is a relative difficulty between one specific reading and one specific user. Most of the existing methods haven't used any metric similar to the relative difficulty to accurately define the distance between a reading and a user.

Chapter 4

Recommendation Beased on Weighed Relative Difficulty

4.1 Overview

In this study, a responsive web application, Adaptive Learner, was developed in this research for both data collection and experimental purpose. Adaptive Learner includes an estimator that predicts the relative difficulty of a reading for a user, a reading recommendation system that will suggest readings to each user based upon the prediction of relative difficulty from the estimator, and a translation annotation module that allows users to translate any unknown words or phrase into their native language by simply selecting these words or phrases.

In a traditional article recommendation system, constructing an accurate profile of each user's interests is essential for the recommendation systems to be successful. [40] There exist many means of obtaining an accurate user profile. One of the most typical ways of acquiring user profile is asking users to manually create their profile before they start using the application and later on ask them to update their profile if their interest changes. According to Liu [40], this kind of approach places an extra burden on users, something very few are willing to take on. Instead, many systems take the approach of constructing the user profiles automatically from users' interaction with the system. These interactions include elements such as followings:

Tracking user's clicking

Through tracking all articles that user has clicked on, it becomes easy to distinguish what kind of category that this user is interested in.

The length of time that a user has spent on a reading

By tracking the time that a user has spent on a reading, it can be easily determined whether a user has finished the reading or not. If a user exits from an article very fast, the system can read this action as a misclick and update the user profile.

Whether a user has commented on an article

Commenting action requires a user to take time and efforts which usually indicates that user is paying extra attention to this article.

Tracking users' interaction with the system and automatically create and update user's profile has become the default requirements in most of the large-scale user applications

today. Automatically updating user's profile also allows the system to be able to adapt quickly to sudden changes in user's interest. Good user experience is an essential requirement for a user-oriented application to success. Anything that gives extra burden such as manually creates a user profile to the users will not work in an online large-scale user application. Because this study focus specifically on dealing with personalized article recommendation based on user language level, it is necessary for us to find a way to automatically create and update users' profile as well as detect users' improvement or decline in their language level.

To avoid putting extra burdens on the users to create and update user profile through pre-test as it was done in most of the previous studies, in the proposed system, a translation annotation module is integrated to achieve the automation of creating and updating users' language level profile as it is being done in most of the online recommenders today. The translation annotation module allows users to select any unknown words or phrases to see its translation in their native language. The translation annotation module current supports translating English to Chinese and English to Japanese. When users read articles on the proposed the system and use the translation annotation module to find meanings for unknown words, the system can construct a user profile of each user's known and unknown words. The next question is how to utilize this user profile information to recommend personalized articles to the users based on their language level.

To utilize users' unknown clickings from the translation annotation module, Carver's Relative Difficulty is adopted. Carver states that there is a relative difficulty between one specific reading to one specific user, and one way of measuring the relative difficulty is through the percentage of unknown words that a user has towards a specific reading. By utilizing the users' unknown clickings data, Carver's relative difficulty can be easily calculated. And by calculating the relative difficulty of each reading towards each user, the system can obtain the distances between each user and each reading.

Carver's relative difficulty is used as the foundation for the recommendation system proposed in this study. Because the relative difficulty illustrates the distance between a reading and a user, knowing the relative difficulty means accurately understanding how hard a reading is for a specific user. And the recommender system can simply use the relative difficulty of a reading towards a user to recommend items that suit user's current language level. Hence, we believe that it is critical to predict the relative difficulty of each reading towards each user in order to make appropriate articles to users in our system. However, through the experiment phase, we realized that unlike the experiment that Carver has conducted in his research where every student intentionally circles out every unknown word in the reading, in the proposed system, there exist many human errors such as misclicking and unknown words that weren't clicked by the users. In Carver's definition of relative difficulty, every unknown word is weighted equally. However, this seems to be flawed in our problem settings. To deal with this issue, we proposed a weighted relative difficulty derived from Carver's definition. The proposed weighted relative difficulty uses frequency of the words as standard to apply a weight on every unknown word that user has clicked through the translation annotation module. By doing so, a word with a higher frequency such as 'the' and 'you' will weigh less to reduce the effect of misclicking on these words whereas a word with a lower frequency will weigh higher to increase its importance in calculating relative difficulty.

To predict the relative difficulty, an estimator for relative difficulty is proposed in this study. Both definitions of relative difficulty (Carver's and ours) are passed in as features to the various estimators to find out which relative difficulty measure is better at determining the distance between a reading and a user. One type of estimator algorithms that was attempted in this study is to call collaborative filtering [30], which is one of the most popular recommendation methods.

Collaborative filtering is a process of data filtering to find specific information, patterns, or similarities using techniques involving collaboration among multiple users. Typical collaborative filtering system will construct a 2-d recommendation matrix based upon all users' profiles. In a simple traditional reading recommendation system, the data used in the 2-d recommendation matrix could be 1 and 0 where 1 stands for a user has read an article and 0 stands for a user has not read an article. So if both user A and user B have read similar types of articles, then if user A has read an article that user B hasn't read, since these two users are so similar, the traditional recommender will recommend this item to user B.

In this study, the data passed into the 2-d recommendation matrix is the relative difficulty. Two different 2-d recommendation matrices were constructed using both Carver's definition of relative difficulty and ours. 8 different collaborative filtering algorithms were tested with the collected datasets. Another type of estimator that is also tested in this study is the linear regression estimator. Due to the continuity feature of relative difficulty which is shown in the collected dataset, linear regression estimator is also attempted and compared with the collaborative filtering based methods.

The personalized recommendation system takes the estimator's prediction of a user's relative difficulty of each reading as input and ranks articles with relative difficulty. The recommendation system suggests articles with lower relative difficulty to each user to ensure that the content of the article is understandable. In the proposed system, each article will be labeled with relative difficulty which is estimated specifically for each user. Learners using the proposed system can also challenge articles with higher relative difficulty.

Figure 4.1 shows the design schematics of the proposed system, which consists of five major module including:

1. a database where all the readings, user info, and tracking data were stored,
2. a server where all API calls such as user login were handled,
3. a responsive web interface with the translation annotation module where users read all the articles at,
4. a relative difficulty estimator where each user's relative difficulty of each unread article is predicted,
5. and the reading recommendation system where appropriate article recommendations are suggested to each user based on the relative difficulty estimator.

Many technologies are used in this study including:

1. MongoDB for the database,
2. Meteor Framework for backend development,
3. React and Redux for frontend responsive web application,
4. NTLK toolkit for lexical manipulation of the texts,
5. Surprise package for developing the proposed reading recommendation system,
6. And Scikit-learn for evaluation of the proposed methods

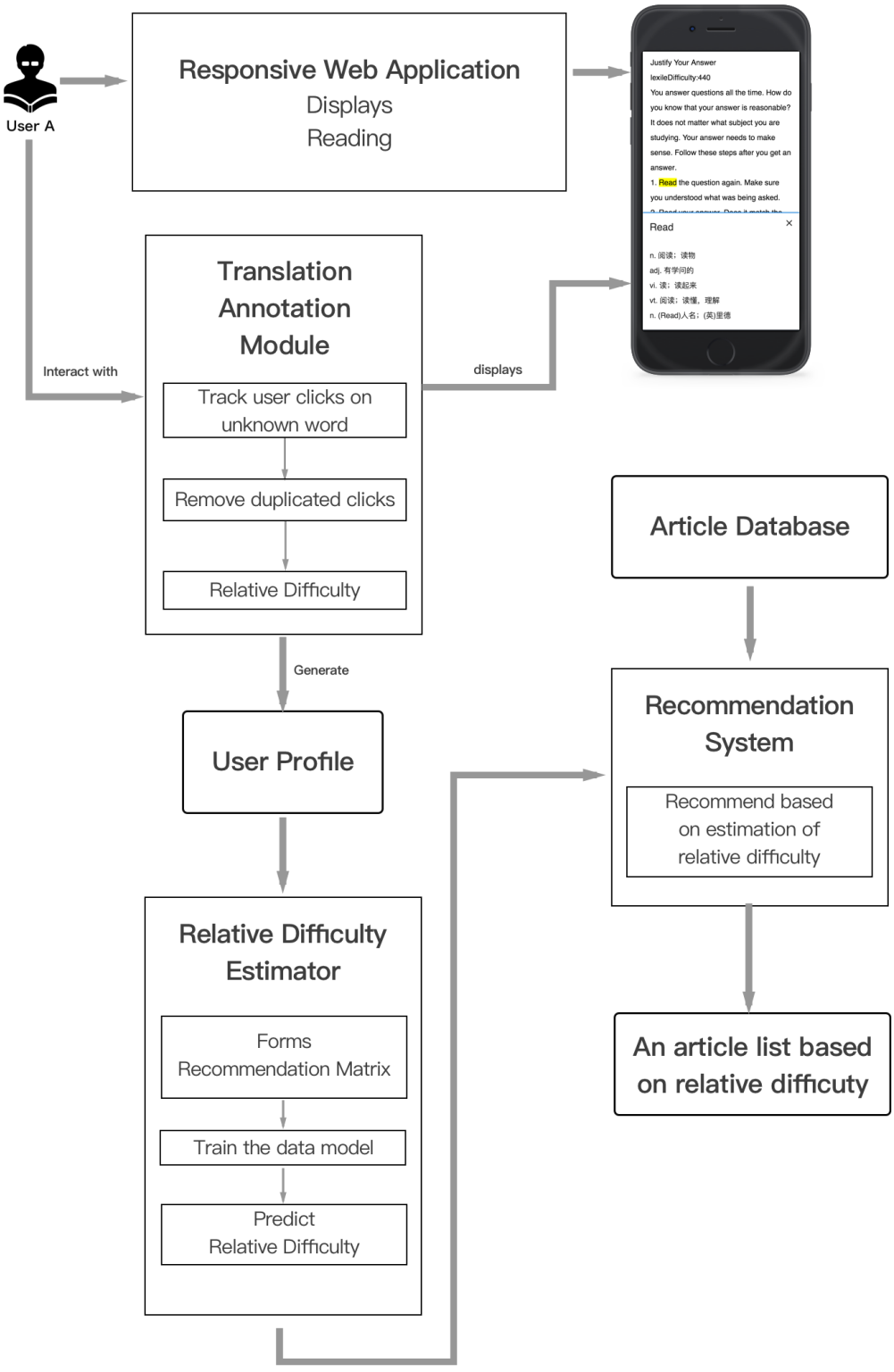


Fig. 4.1. The overall system design schmatic

4.2 Translation annotation module

Csikszentmihalyi [41, 42, 43] proposed a concept called flow, which is a state that people will enter while performing any activity. A person is most likely to enter a flow state while wholeheartedly concentrating on a task or activity for intrinsic purposes. Krashen also proposed similar idea regarding language learning indicating that language learners are more likely to acquire knowledge if he/she is in the right state of mind. However, flow state can be easily broke out of due to reasons such as boredom or anxiety. It is particularly hard to be in a flow state if a person is reading materials that are much more difficult than his/her language level. One of the reasons that a hard reading material may break user's flow state is unknown vocabulary. A very typical action that a language learner does when he/she runs into unknown words is simply using a dictionary or translation application to find out the meaning of the word. However, this translation action will take user's eyes away from the reading, which ends the flow state. And readings with a lot of unknown words will cause this translation action to repeat many times which may lead to user's annoyance and lose interest in the articles.

Carver [14] proposed a formula for measuring relative difficulty between a specific user and a specific reading. The formula requires the amount of unknown vocabulary that a user has towards a reading and the amount of total unique words that a reading contains. During Carver's experiment, to find out the relative difficulty, he simply conducted an experiment that asked all users to circle out the words that they are not familiar with in the reading and counted these circles manually. In our research, to calculate the relative difficulty of each user toward each reading, we will have to collect the number of unknown words that each user has towards each reading.

To solve the two problems above, a translation annotation module is added to our system. Users can click on any unknown words, our translation annotation module will provide dictionary meanings in user's native language. Users can also select multiple words or sentences to see a machine-translated meaning of the selected texts. The translation module currently supports translations from English to Chinese and English to Japanese. By using the translation annotation module, users no longer need to look up unknown vocabulary on another device or an actual dictionary. By stopping users from looking away from the reading materials to look up unknown words' meanings, it helps users stay focused on the material and stay in the flow state. On the other hand, by collecting the unknown words that user has selected, we can find out the amount of the unknown words that a user has towards a specific reading easily. We can simply query in the database with the user id and the reading id to find out how many unique unknown words that a user has translated using the provided translation annotation module on a specific reading. The total number of unique words in the reading can be very easily obtained through NLTK package. Therefore, the calculation of Carver's definition of relative difficulty can be accomplished through the proposed system.

Figure 4.2 shows the comparison between the interface of the translation annontation module implemented in this study and the one implemented by Amazon Kindle [44].

4.3 Relative difficulty vs weighted relative difficulty

Carver first proposed the idea of relative difficulty in 1994 stating the difficulty of the reading isn't same for every learner, to more accurately measure the difficulty of the reading in correspondence to each learner, a measure of relative difficulty between one specific reading and one specific user is needed. Later on, in Carver's study, he conducted

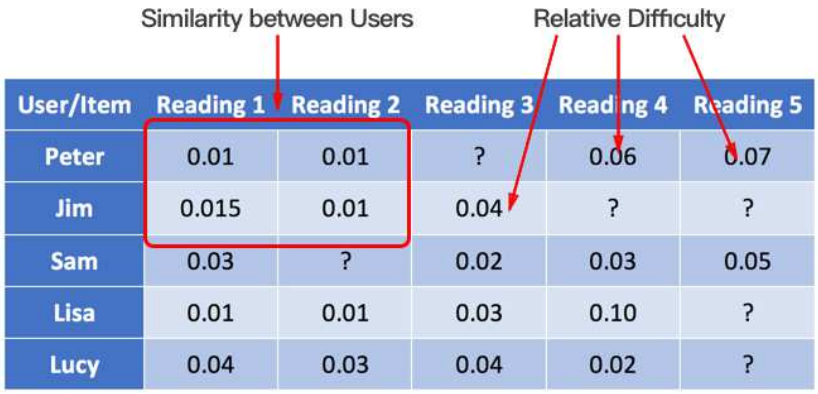
Fig. 4.2. The proposed translation annotation module vs the translation annotation module in Kindle

experiences on students and provided a formula for measuring the relative difficulty which is defined as the percentage of unknown words that a user has towards a specific reading. At the early stage of this study, only Carver's definition of relative difficulty is adopted as the based on the proposed recommendation system. However, in the data collection process, several problems were detected:

1. First, unlike Carver did in his research where every user intentionally picked out all the unknown vocabulary. While using the Adaptive Learner, no all user will click on every unknown word, because sometimes knowing or not knowing a single vocabulary in the reading doesn't affect the overall understanding of the reading, or the unknown word could be simply omitted by the user due to careless or skip on sentences. Because all users have their own reading habits, and these habits closely connect to how they utilize the translation annotation module that the system provides.

2. Second, human error exists while users are reading on the proposed system. Because there exist many actions that a user can perform on a web page such as scrolling, sometimes the user meant to perform another action, scroll to see the rest of the article, but a word was clicked the translation annotation module was triggered and recorded by the system. This happens particularly often when users are using the application on their smartphones. How to mitigate the effect of these mishandling on the result of relative difficulty has been an issue.

3. Third, in Carver's definition of relative difficulty, every unknown word are weighted the same. In Carver's definition, all the words are weighted equally meaning that



User/Item	Reading 1	Reading 2	Reading 3	Reading 4	Reading 5
Peter	0.01	0.01	?	0.06	0.07
Jim	0.015	0.01	0.04	?	?
Sam	0.03	?	0.02	0.03	0.05
Lisa	0.01	0.01	0.03	0.10	?
Lucy	0.04	0.03	0.04	0.02	?

Fig. 4.3. A recommedation matrix filled with the relative difficulty

the unknown word "the" has exactly the same effect to user's language level profile as the word "apocalypse". However, if a user doesn't know a very simple word, then it's very unlikely that he knows any word that is way harder than the unknown simple word.

To mitigate the issues that were detected during the data collection process, a weighted relative difficulty is proposed. The proposed weighted relative difficulty uses frequency of the words as standard to apply a weight on every unknown word that user has clicked through the translation annotation module. By doing so, a word with a higher frequency such as 'the' and 'you' will weigh less to reduce the effect of misclicking on these words whereas a word with a lower frequency will weigh higher to increase its importance in calculating relative difficulty. Formula shows how Carver's relative difficulty is calculated while Formula shows how the proposed weighted relative difficulty is calculated. Both definitions are used later on to distinguish which measure is better suited for our method.

4.4 A run-time estimator of relative difficulty

The proposed system uses relative difficulty as the base measure to determine the distance between a reading and a user. Hence, it is critical for the system to be able to estimate readings' relative difficulty in terms of each user in order for the recommendation system to recommend articles to the users based upon these estimations. For finding the best estimator that can most accurately predict the relative difficulty of each reading in correspondence to each user, two types of estimation models were attempted. One is called the collaborative filtering which is a method used extensively in most of the online recommendation system, the other one is called linear regression method, which is chosen due to the continuous nature in the dataset.

4.4.1 Collaborative Filtering

Unlike the tradition recommendation system that recommends items based on user interest, the recommender in this study recommends items based on relative difficulty. To accomplish such tasks, our system first calculates the relative difficulty of each user towards each reading using the translation annotation module. More specifically, natural language processing is used to find all unique words in each reading. Then the number of unknown words was collected through the translation annotation module by counting how

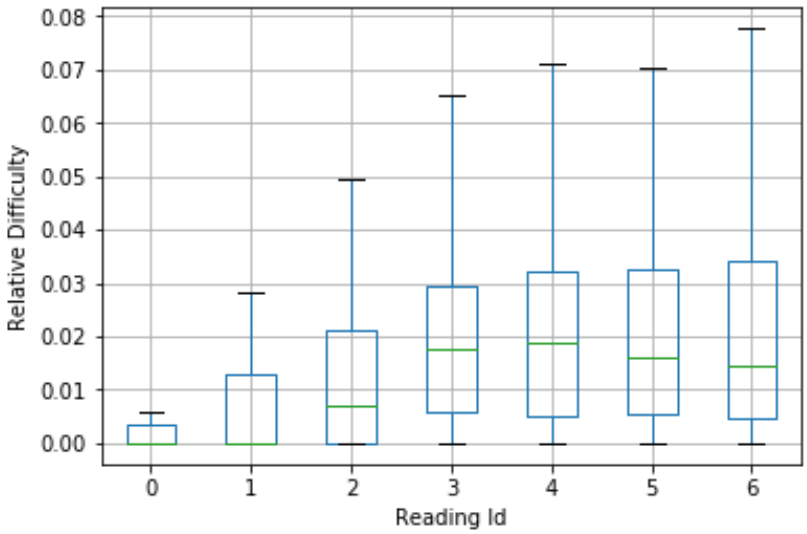


Fig. 4.4. A box plot of user's relative difficulty distribution on of each reading

many unique words that a user has selected. Finally, the relative difficulty is calculated using Equation (4.1), as it is defined by Carver.

$$Carver's\ Relative\ Difficulty = \frac{Number\ of\ unknown\ vocabulary}{Number.\ of\ total\ unique\ vocabulary\ in\ the\ reading} \tag{4.1}$$

In this study, another weighted relative difficulty defined as shown in Equation (4.2) is also proposed to mitigate the issues detected as it was explained in the section Relative difficulty vs Weighted relative difficulty.

$$Weighted\ Relative\ Difficulty = \frac{\sum_{k=1}^{n} vocabulary\ frequency}{Number\ of\ total\ unique\ vocabulary\ in\ the\ reading} \tag{4.2}$$

After obtaining all the relative difficulty of each user towards each reading, the recommendation matrix was constructed as shown in Figure 4.3. So the cell at column A and row W means the relative difficulty of item W to user A is 0.014. With the recommendation matrix constructed, we can now calculate the similarity user-wise and item-wise using cosine similarity as it is defined in Formula (3.3). 8 collaborative filtering based algorithms were used in this study to test and evaluate which algorithm will most accurately predict the relative difficulty of articles that user hasn't read given user's reading history. The 8 algorithms include SVD, Normal Predictor, SVDpp, KNNBasic, KNNBaseline, BaselienOnly, SlopeOne, KNNWithMeans, KNNWithZScore.

4.4.2 Linear Regression

Due to the dataset's regressive data nature, a simple linear regression model is also used in this study to compare with the traditional recommendation algorithms. Figure 6.3 is a box plot where y-axis is the relative difficulty and x-aixs is the reading id. As it was stated before, the readings in the database are from easy to hard. Therefore, this box plot illustrates that there is a positive relationship between the relative difficulty and

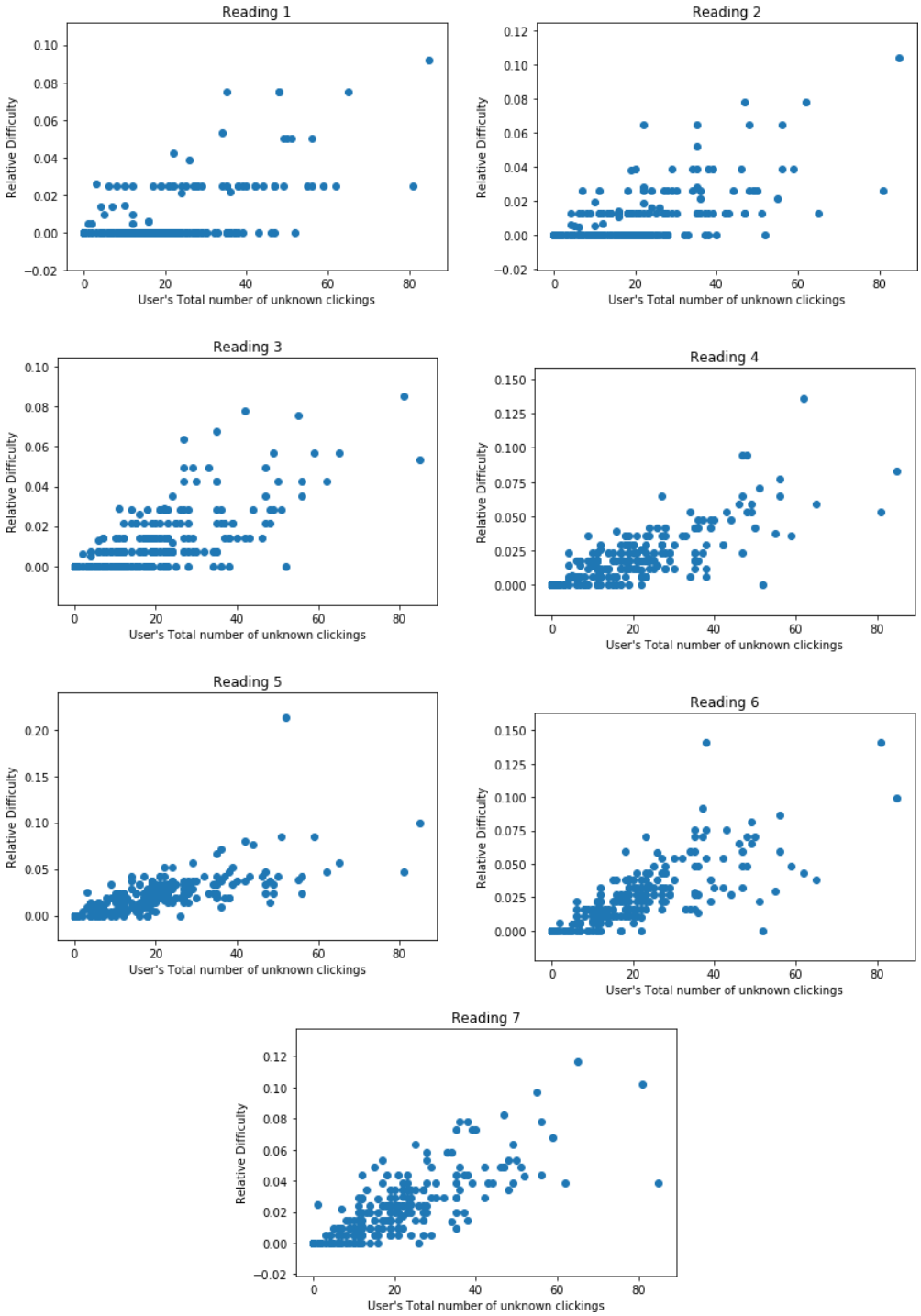


Fig. 4.5. Plots of relative difficulty vs user's language level measured in user's total number of unknown clickings

	User Features				Article Features		
Relative Difficulty	0-1000	1000-2000	2000-3000	...	0-1000	1000-2000	2000-3000
0.02	4	10	15	...	4	25	32
0.04	0	0	20	...	5	20	44
0.04	2	10	25	...	10	22	24
0.01	0	0	10	...	4	10	40
0.05	0	0	13	...	15	13	31

Fig. 4.6. Features used in linear regression model: the user feature 0-1000 indicates the number of unknown clickings this user made in total that has a frequency between 0 and 1000, the article feature 0-1000 indicates the number of words that this articles contains with a frequency between 0 to 1000

the reading difficulty level. It is reasonable to think, in general, the harder the general difficulty is, the higher the relative difficulty would be for each user. Of course, there will be outliers in the situations, such as, readings with specific terminologies in which people with knowledge of these words can easily understand, but people without knowing them can hardly understand. Because the relative difficulty is not only affected by the difficulty level of the readings, it is also affected by the language level of the users. The plots of each reading, where y-axis is the relative difficulty and x-axis is the user language level presented numerically as the measure of total amount of clickings that a user has made on all the readings, are also presented in Figure **??**. There are 7 graphs in total in Figure **??** where each graph presents the relationship between the total number of unknown clickings of the users and the relative difficulty. A positive relationship between the total number of unknown clickings of the user and the relative difficulty can be observed in all graphs, this relationship is particularly obvious in reading 4 and reading 5.

Data prepared for linear regression model is quite different from the data prepared for collaborative filtering. The linear regression model takes slightly different features as input. Instead of the recommendation matrix shown in Figure 4.3, the linear regression model takes both features from the articles and from the user forming a feature vector as shown in Figure 4.6. For the linear regression model, the accumulated data is input as the user profile. This means that if a user has read 3 articles and the total amount of unknown clicking is 25. All 25 unknown clickings are used as user language level profile. All the unknown clickings are then classified into many groups based on their frequency. For example, the word "hello" has the frequency of 47.86 and it is placed into the group of frequency 0 - 1000. Each group is used as a feature for the user profile and the frequency range of the group is determined through running all combinations of possible groups from 1 to 100. In every combination, the linear regression model was constructed.

4.4.3 A recommendation system based on the run-time estimator of relative difficulty

An accurate user profile is the key to good recommendations for the users [20]. There are systems [2, 9] that require users to manually input their profiles, which places an extra burden on users. These kinds of manual work reduce users' interest in continuing using the application [20]. Instead, systems that are capable of automatically constructing user

profile through tracking users' interaction with the system are more preferable. Recent recommendation systems often track user's interaction action such as likes/dislikes, review, rating, and etc to form the user profiles.

In personalized reading recommendations for the language learner, most of the systems [11, 9] are still in the stage of having users manually input their profiles through procedures such as pre-tests. In Hsu's research [11], a pre-test and a questionnaire were given to evaluate the English reading ability and preferences of the students. In Kuo's research [9], student's writing homework was selected as input to evaluate student's language level. Like the traditional recommendation systems, the use of pre-test will reduce user's interest in continuing user the application. Hence, either Kuo or Hus's system is applicable to large-scale user systems such as Newsela [13], a platform for English learning through news reading.

In this study, we proposed a recommendation system that provides personalized reading recommendations based on the run-time estimator of the relative difficulty. The recommendation system takes the estimation of the relative difficulty of the readings and labels these values to next to each reading that is recommended to every user. The recommendation system was designed to recommend readings with lower relative difficulty to the users so that they can fully understand the content of the reading.

Chapter 5

Dataset and Data Collection Process

As suggested in Chapter Method, the dataset that needed for the proposed method of work requires relative difficulty, which can be calculated through tracking user's unknown clickings on the articles. Because the weighted relative difficulty is later proposed, not only the total number of unknown words is required, but also the word itself needs also be recorded in the dataset. To be able to acquire such data from the users, the system needs to at least provide a translation annotation module and record all the interactions that users make with the system through the translation module.

Even though many applications such as Amazon Kindle [44] and Seed [45] provide the translation annotation module in their applications. It doesn't necessarily indicate that they track the data that were required for testing the proposed method. After thorough research, no public datasets exist that provides the dataset, which is required for testing and evaluating the proposed system. We have contacted Amazon Kindle and Seeder to ask them if they have tracked the user unknown clickings in their application, and if so could they provide the data to us for research purpose. None of them replied. Therefore, we had no choice but to conduct the data collection process on our own. Later on, a one-year long data collection was carried out online. In March 2017, Adaptive Learner, the responsive website that was developed specifically for collecting the proper data from the users in order to verify the proposed method, was published. By March 2018, the total amount of users registered on the Adaptive Learner is 328.

All the participants' native language is Chinese. Even though our application's translation annotation module supports English to Chinese and English to Japanese and a research question of whether the method will work the same for people with different native language was proposed at the beginning of the research, due to the time limitation, this question and functionality isn't verified in this thesis. The participants are mainly students from different grades. The invitation of participants is mainly through two means:

1. Peer-to-peer, where I asked my friends to distribute the website and ask them and their friends to participate in the data collection;
2. Working with several tutoring-centers, where I provided them with an online Toeic test practice website in exchange for having their students to join the data collection.

In total three different tutoring centers have collaborated with us in data collection phase. One of them is located in Tokyo which is the one that we have received a lot of help from throughout this research. We would like to express our sincere appreciation to them in the Acknowledgement section. The final datasets can be found at [46]. An example of a user's final data in JSON format is shown in Appendix A.

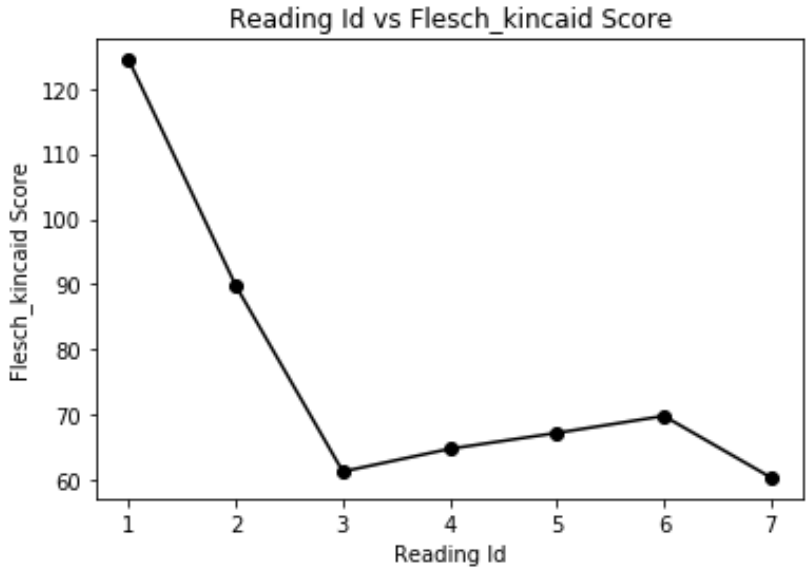


Fig. 5.1. Reading readiablity calculated using Flesch-Kincaid Algorithm

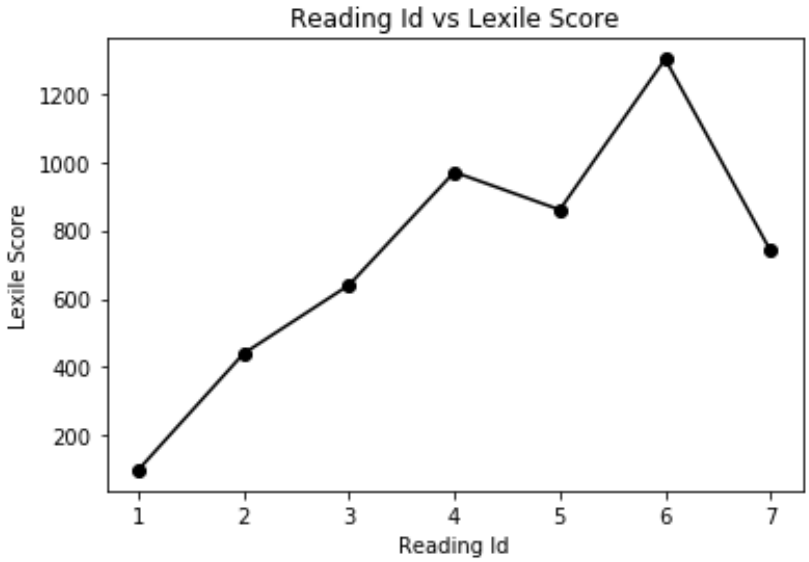


Fig. 5.2. Reading readiablity calculated using Lexile Framework

5.1 Article data

All participants joined our research were required to read 7 different readings which are arranged from easy to hard. The article readability is determined based on both the Flesch-Kincaid measures [23] and Lexile Score [24]. The Flesch-Kincaid scores of the readings are shown in Figure 5.1. The Lexile scores of the readings are shown in Figure 5.2. The reading 1 through 3 has clear difference in difficulty, however, in reading 4 through 7, their difficulty is very close to each other. This variation between the distance of each reading in difficulty provides variation in data for evaluation purpose. All 7 readings are

selected from the K12Reader, which is a popular educational reading material website [47]. At the beginning of the research, there was the debate of whether more articles should be added to the database. Initially, 83 readings were imported to the article database to provide users with variation in content. And each user would still read seven articles but the articles provided to the users would be different, which may result in some users don't share anything in common meaning that two users haven't read the same article. After serious consideration, we decided to provide the same 7 readings for all users to generate more connections in data.

5.2 User data collection

The flowchart of the steps in the conducted data collection is shown in Figure 5.3. These steps can be ordered as follow:

5.2.1 Instruction for the experiment and user agreement

When participants first visit the Adaptive Learner, there shows a thank you note and a short instruction for this data collection. By the end of the instruction, it shows a short user agreement stating that all users participate in this research agree to have the application record and track their data and agree to have the data to be used for research purpose.

5.2.2 User registration

To be able to identify each user and add relative user profile information to each unique user, a simple user registration and login module are implemented. Many implementations were considered to simplify the signup process for each participant, such as record the IP address of every access to identify users or embed a cookie in the browser to identify each user. By the end, we decided to have users pick a unique nickname as their login to ease the user account registration process. This implementation is relatively easy and did work smoothly in a not so big user group. If users couldn't finish all the readings at once, they can always log back in using only the username they have picked to continue reading at where they have left off. Later on, to further reduce participants workload, a unique username is automatically generated for each user when they first visited the website, they can choose to edit it or just use it for registration.

5.2.3 Collecting user's basic information

After a simple registration, participants were asked to complete a short questionnaire about their background including education status, age, and etc. Users can choose not to answer most of the question, but one field is required called self-evaluation of language level. This field is proposed at the beginning of the research, for providing users with readings that closely match their language ability when no information has been collected about them.

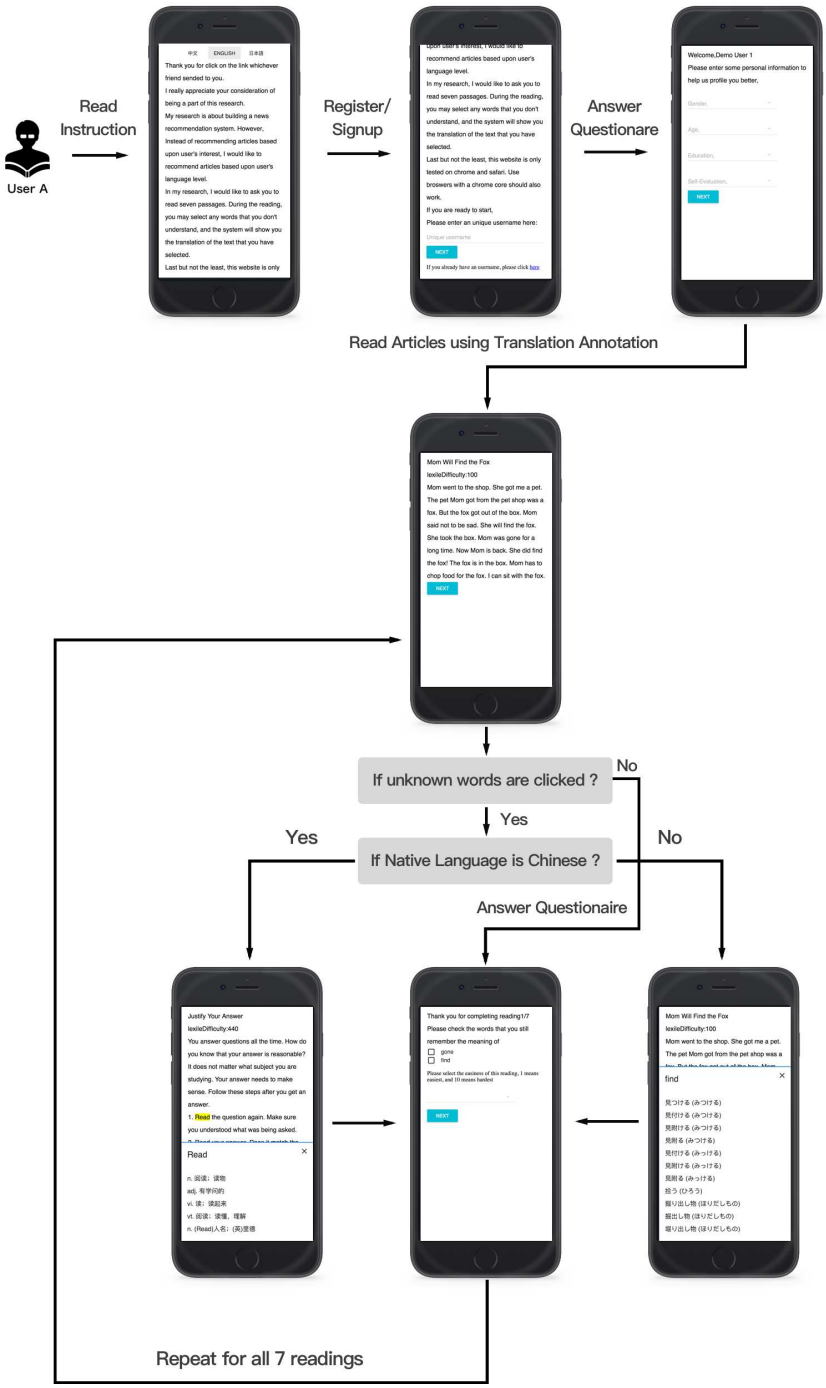


Fig. 5.3. User data collection procedure

5.2.4 Reading the articles

After filling the short questionnaire, the readings are presented to each user one by one in order from the easier ones to the harder ones. During the reading process, the translation annotation module was provided to the users. Participants can press or select any unknown words or phrases, the translation of the words and phrases will appear at the bottom in their native language. During this process, every time that participants used the translation annotation module, the system will record it. Also, the system will record the start time and end time of each user reading each article to calculate how much time it takes each user to finish each article.

5.2.5 Questionnaire after each reading

All participants are asked to fill out a short questionnaire after each reading. If the participant did use the translation annotation module during the reading process for unknown word translation, the questionnaire would ask the user to pick out the words that they still remember the meaning of. Another question is how hard each user personally thinks the reading is according to their own current language level. The scale of this question is between 1 to 10, where 1 means that it is not hard at all, and 10 means it is way too difficult. Later on, we found that whether user remembers what they have translated using the translation annotation module isn't so relevant to the proposed method. These collected data weren't used in the proposed method.

5.2.6 Reading data preprocessing

In the proposed method, the readings are not imported directly as input to our data model. So it requires some preprocessing work before importing them into the data model. Because the proposed method adopted the idea of Carver's relative difficulty [14], it is important that we can calculate the relative difficulty from the data we collected. Carver states that one way of measuring relative difficulty is through the percentage of unknown words that a user has towards a specific reading. To calculate the percentage of unknown words, we need to obtain both the number of unknown words that a user has towards a reading, which can be obtained from the translation annotation module and the total number of unknown words in the article, which can be obtained through reading data preprocessing. In this step, the NLTK package was used to find all the unique words in a reading. The NLTK package provides a function to identify words in singular form and plural form. In regular string manipulation, "animal" and "animals" are treated as two different words. By using the NLTK package, it is possible to identify words in its singular form and plural form as the same word so that an accurate total number of words can be found.

5.3 User data validation

User data validation occurs after user finished reading the materials that we provided. If a user didn't finish the entire data collection process meaning that this user didn't finish reading all the articles, this user's data will be treated as invalid data and excluded from the final datasets. Since the amount of time that each user reads each reading is recorded

during the data collection, if a user took extra long or extra less time during the data collection, this abnormal behavior will be detected and the user data will be treated as invalid. These types of users' data were excluded from the final datasets. After user data validation, there remain 239 users whose data are verified for the research. These 239 users' data are entered into the final datasets and being used for training and validating the data model in the evaluation section.

Chapter 6

Results and System Evaluation

6.1 Summary of the users' basic information

The total amount of users participated in this research is 328, in which 269 users' data passed through the data validation and added to our final datasets. The 328 users are all from China, which can be easily determined through the native language that they have picked while using the translation annotation module that the system provides. The age distribution of the participants is as shown in Figure 6.1. Most of the participants are in the age of between 18 to 34 due to the reason that all the participants are invited through peer-to-peer who are all in the similar age range as me. The education distribution of the participants is shown in Figure 6.2. 157 out of 269 users have obtained their bachelor degree and 29 out of 269 participants are graduated as masters.

6.2 Demonstration of similarity in users' relative difficulty

There are in total of 7 readings in our reading database, and each test subjects are required to read through all of them. The general difficulty of the reading increases from reading 1 to reading 7. Figure 6.3 shows the distribution of relative difficulty for each user towards each reading. It is illustrated that on average the higher the reading difficulty is, the

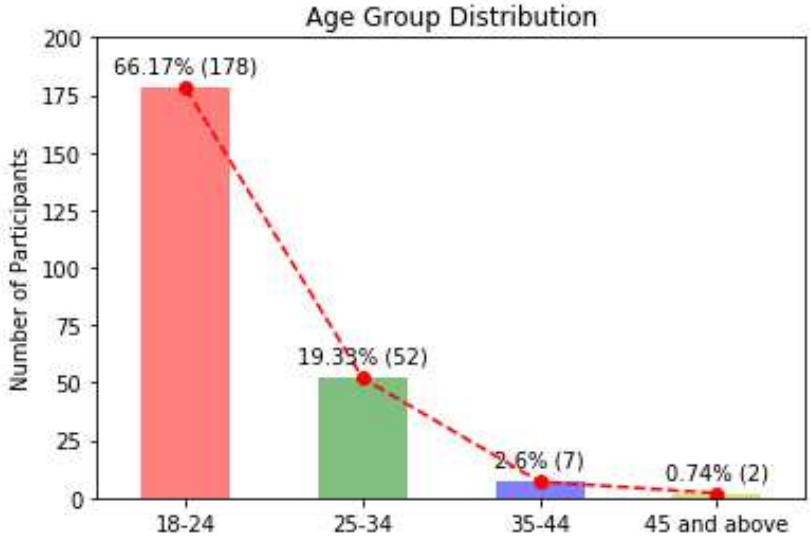


Fig. 6.1. Age distribution of the participants

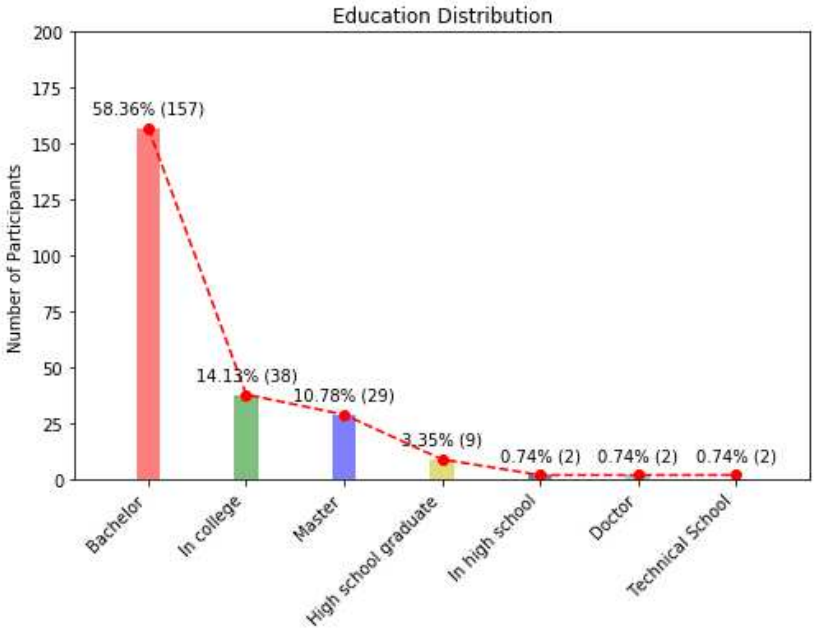


Fig. 6.2. Education distribution of the participants

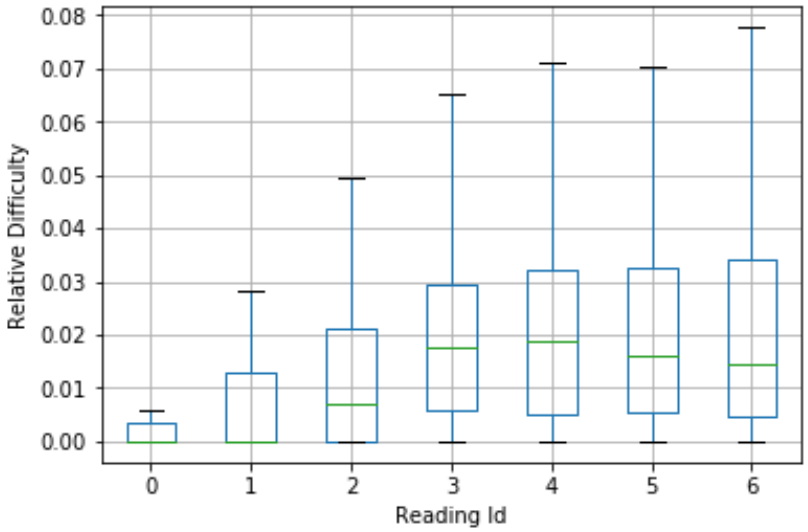


Fig. 6.3. The relative difficulty vs reading readability

higher the relative difficulty for each user gets. The variation in users' relative difficulty of the same reading demonstrates the theory that the relative difficulty of the reading is different for each user. If we look at the relative difficult distribution of reading 6, the highest relative difficulty for a user is close to 0.08, while the lowest relative difficulty is 0. And the majority of the relative difficulty of reading 6 lies in between 0.005 to 0.035, which is a relatively big range.

Collaborative Filtering requires there to be similarities between users in order to predict the unknown information from a collaboration of other user's existing data. So it is critical to demonstrate that there exist similarities in users' clicking behavior, which indicates that

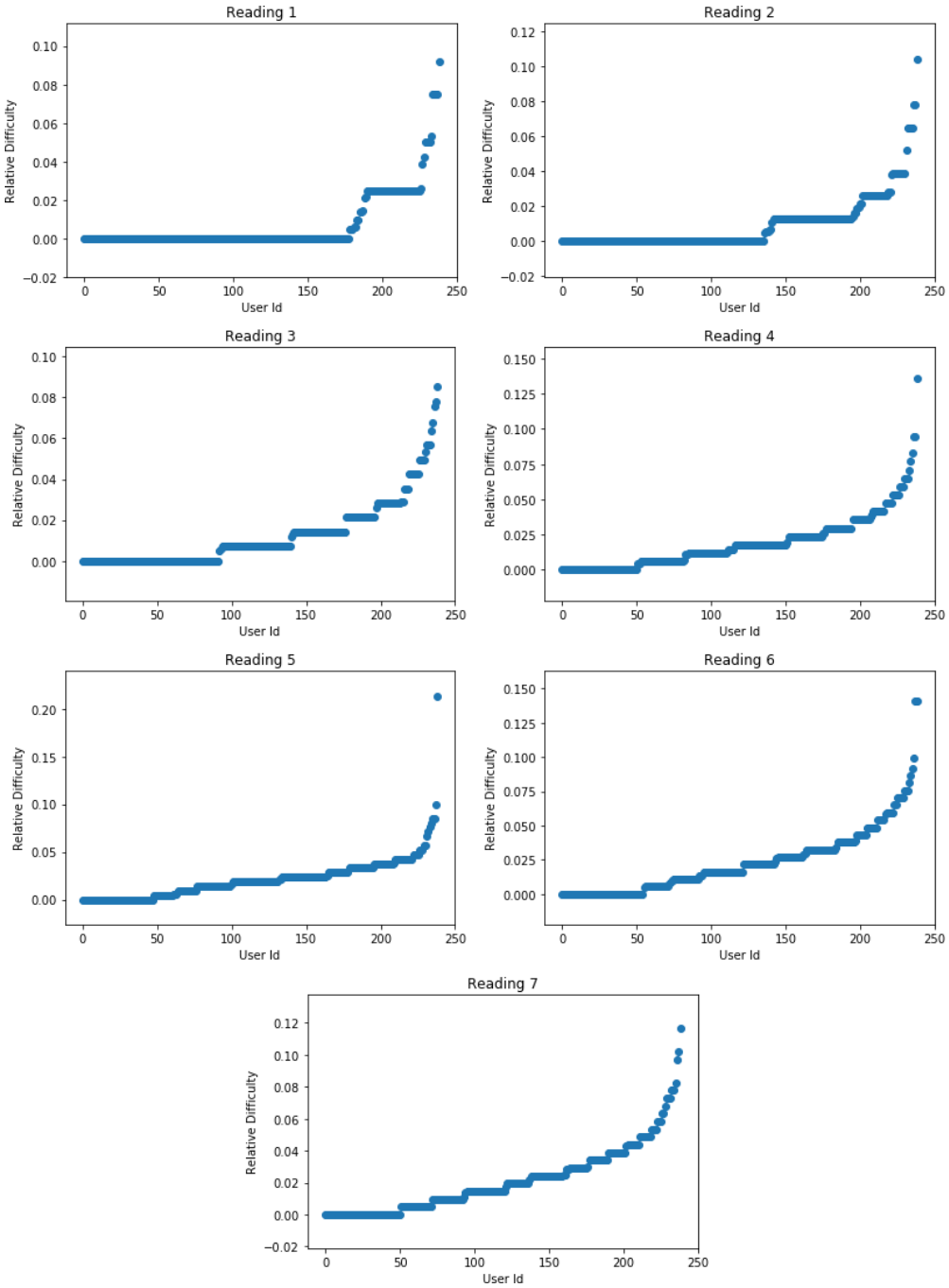
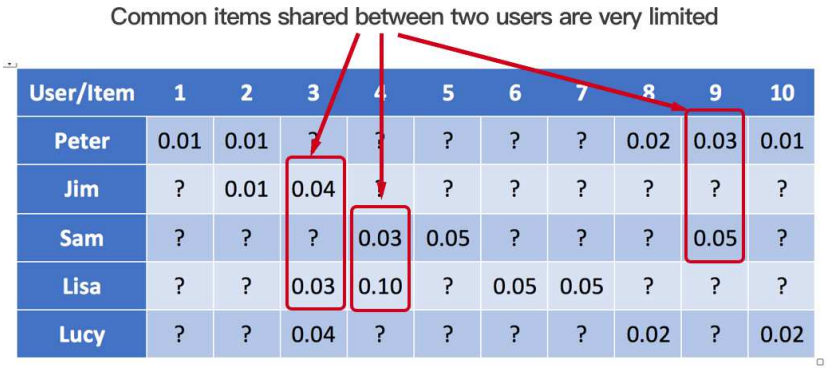
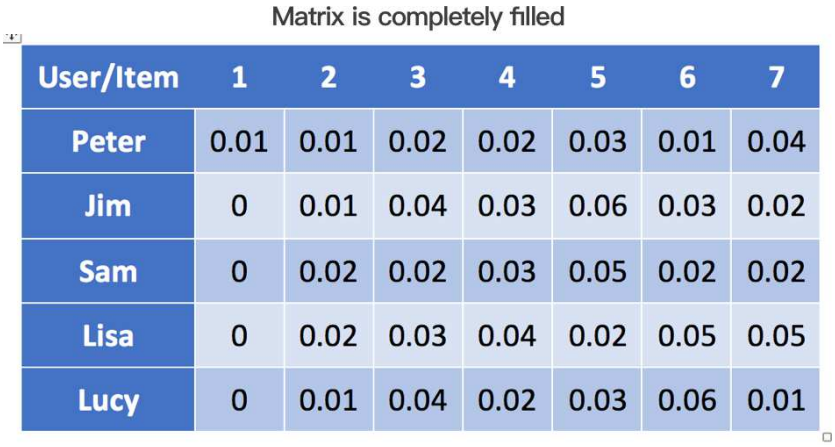


Fig. 6.4. Plots of every article's relative difficulty for all users



User/Item	1	2	3	4	5	6	7	8	9	10
Peter	0.01	0.01	?	?	?	?	?	0.02	0.03	0.01
Jim	?	0.01	0.04	?	?	?	?	?	?	?
Sam	?	?	?	0.03	0.05	?	?	?	0.05	?
Lisa	?	?	0.03	0.10	?	0.05	0.05	?	?	?
Lucy	?	?	0.04	?	?	?	?	0.02	?	0.02

Fig. 6.5. A typical sparsed recommendation matrix



User/Item	1	2	3	4	5	6	7
Peter	0.01	0.01	0.02	0.02	0.03	0.01	0.04
Jim	0	0.01	0.04	0.03	0.06	0.03	0.02
Sam	0	0.02	0.02	0.03	0.05	0.02	0.02
Lisa	0	0.02	0.03	0.04	0.02	0.05	0.05
Lucy	0	0.01	0.04	0.02	0.03	0.06	0.01

Fig. 6.6. The recommedation matrix in this study

there are also similarities in users' relative difficulty. Figure 6.4 shows graphs of each user's relative difficulty for each reading. The x-axis stands for each user's id whereas the y-axis shows the value of relative difficulty. All graphs in Figure 6.4 appears to be in a shape of stairs. Each step of the stair indicates a group of users who share the same relative difficulty for a specific article. For instance, 94 users' relative difficulty on reading 3 is 0; 62 users' relative difficulty on reading 3 is 0.01. In our datasets, we can split users into 14 groups based on each user's relative difficulty. The similar trend can also be found in the rest of the articles which are shown in Figure. Because there exist similar users in our datasets, collaborative filtering methods should be a worthwhile data modal to attempt for predicting the relative difficulty of each user towards each reading.

6.3 Estimator evaluation

Scikit-learn and pandas python package is used extensively in the estimator evaluation. After data preprocessing, the obtained dataset is filtered and only the data needed for the recommendation matrix are used, which are user id, reading id, and the corresponding relative difficulty. After datasets are prepared into the right format for data model training purpose, nine popular algorithms including SVD, Normal Predictor, SVDpp, KNNBasic, KNNBaseline, BaselienOnly, SlopeOne, KNNWithMeans, KNNWithZScore are selected as data models for our estimator. To find out which existing algorithm can most accurately

Table 6.1. The RMSE for all the collaborative filtering algorithms

	Algorithm	RMSE (Root mean square error)
1	KNNWithZScore	0.0157
2	KNNWithMeans	0.0158
3	SlopeOne	0.0171
4	KNNBaseline	0.0178
5	BaselineOnly	0.0180
6	KNNBasic	0.0188
7	SVDpp	0.0236
8	NormalPredictor	0.0266
9	SVD	0.0459

estimate the relative difficulty using the collected dataset, cross-validation was adopted in this study. The cross-validation is a model validation technique for evaluating how the results of a data model will generalize to an independent data set. There exist many types of cross-validation method. The one that was picked for evaluation is called k-fold cross-validation. In this validation method, the original data sample is randomly split into k equally partitioned subsamples. From the k subsamples, one sample is picked as the validation data for the testing purpose, and the rest k - 1 subsamples are used as training data. The cross-validation was then repeated k times, each time one of the randomly partitioned subsamples is used as test data while the other k - 1 subsample are used as training data. The advantage of this validation method is that all the data are used both as testing and training.

6.3.1 How accurate is the proposed estimator at predicting relative difficulty?

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\Big(\frac{d_i - f_i}{\sigma_i}\Big)^2} \tag{6.1}$$

At each repetition of the cross-validation, the RMSE (root mean squared error) is calculated as the metrics for measuring the accuracy of all the data models. Equation (6.1) shows how RMSE is calculated. The average of the RMSE calculated at each repetition is used as the final RMSE for the data models. The RMSE is a very commonly used measurement of the difference between the predicted value by an estimator and the true value. Formula shows how the RMSE is calculated. Because the RMSE is a squared value, it is always non-negative. An RMSE of 0, which will never be achieved in practice, indicates a perfect data model that predicts all the data correctly. In general, a lower RMSE indicates better fitness of the data model. However, lower RMSE doesn't necessarily indicate a better recommendation.

The cross-validation along with RMSE calculation was performed on every popular algorithm that was listed above. The number of folds for cross-validation was set to 10. The RMSE of each algorithm is listed in Table 6.1. As indicated in the Table 6.1, KNNWithZScore data model outperforms the rest with an RMSE of 0.0157 while the SVD algorithm performed worst with an RMSE of 0.0443. The famous SVD [48] algorithm was popularized during the Netflix Prize in which Simon Funk used it to won the prize. However, SVD algorithm performed worst in the collected dataset was very surprising. The possible reason is that most of the recommendation matrix used in regular recommendation models are very sparse meaning that a lot of the users only watched movie A, B, C while the reading C, D, E haven't been watched. These kinds of recommendation matrix

Table 6.2. The RMSE of KNNWithZScore using leave-p-out cross validation

p	RMSE
Leave 1 out	0.0162
Leave 2 out	0.0164
Leave 3 out	0.0197
Leave 4 out	0.0218
Leave 5 out	0.0235
Leave 6 out	0.0212

are shown in Figure 6.5. SVD based algorithms are very strong at finding out the latent factors between two users with no movies that both of them have watched and rated. However, in dataset collected in this study, every participant are required to read all the readings provided meaning that the recommendation matrix formed in this study has 0% sparsity as shown in Figure 6.6. This fully filled recommendation matrix is suspected to be the cause of the low performance of the SVD based algorithms.

6.3.2 Can proposed system be able to adapt to user's profile change?

Another question to answer is that how many articles that user should read so that data model can give a relatively good recommendation. To answer this question, another cross-validation method called leave-p-out cross-validation is used. Leave-p-out cross-validation is designed to use p observations as the validation set and the remaining observations as the training set. While using it in the collected dataset, p set to 1 meaning that one reading is used as validation set while the other 6 is used as the training set; p set to 6 meaning that training the model with one reading to predict the relative difficulty of the other 6. The result of the leave-p-out cross-validation is shown in Table 6.2. According to the Table 6.2, it is indicated that use six articles as training data to predict the relative difficulty of the remaining one has clearly the best performance. In general, the more the user reads in the system, the higher the accuracy the estimator gets. But even with one reading in the user profile, the system still has the ability to estimate. The decreasing trend in RMSE when more user profile data are added to the data model illustrates that the estimator does have the ability to adapt to the change in the user profile.

6.3.3 Evaluation of the linear regression model

The relative difficulty was also estimated using a linear regression model due to the continuous nature of the data. In the linear regression model, the frequency of the words in the reading are used as features for the readings, and the frequency of the words that users have clicked on are used as the features for the users. By using the features of both user and reading, the linear regression model was used to estimate the relative difficulty of a user with certain language skill towards a reading with certain readability.

The same cross-validation method is used for testing the linear regression model. Surprisingly, the linear regression model was capable of achieving an RMSE of 0.0120, which outperforms all tested recommendation model. The coefficient of determination [49], which is also called r-squared, is also used as an evaluation method for the linear regression. The coefficient of determination is a value between 0 to 1 that measures how well-observed outcomes are replicated by the model using the proportion of total variation of outcomes explained by the model. The result of the coefficient of determination and RMSE of both linear regression model and the best recommendation model can be found

Table 6.3. The coefficient of determination of the linear regression model and the best collaborative filtering model

Data Model	Coefficient of Determination	RMSE
Linear Regression	0.539	0.0120
KNNWithZScore	0.144	0.0157

in Table 6.3. We can see that both coefficient of determination and RMSE indicates that linear regression estimator is better at estimating users' relative difficulty in this dataset.

6.4 Evaluation of the recommedation system

How good is the proposed recommendation system?

Even though both collaborative filtering and linear regression models appear to be able to estimate the relative difficulty defined by Carver, the definition of relative difficulty by Carver is only one way of defining the distance between a user and a reading. A good prediction of the relative difficulty defined by Carver doesn't represent an accurate actual distance between a user and a reading. To truly evaluate the performance of our system at distinguishing the distance between a different user and a different reading, 10 test subjects were recruited. Each user was invited to read the first 5 articles which are used for constructing the user profile and rank the last two articles based on the difficulty of the artile in his/her own perspective. For example, if user A believe reading 6 is harder than reading 7 for him/her, then he/she can rate select reading 6 as the harder one between the two readings. The proposed system will also pick the harder readings based on the relative difficulty that the estimator estimates. All the data model evaluated in this study were tested for this experiment.

As it was mentioned above, the relative difficulty defined by Carver is merely one way of defining the distance between a user and an article. When Carver defined it during his research in the 1980s, he did it through participants intentionally circling out all the unknown words in the reading. In our system, participants can use the translation annotation module when they ran into unknown words. But they can also decide to ignore an unknown word and continue reading the article or click on a known word by accident or purposely checking out the functionality. It is very difficult to know whether a click made is equivalent to an unknown vocabulary. To mitigate these issues, a weighted relative difficulty is proposed in this study. A weighted relative difficulty is an idea based on Carver's research. In Carver's definition, each unknown words weighted the same as shown in Formula 3.2. However, in the proposed system, each click can mean many things, such as misclick, system malfunction, or human error. We adopted the frequency of the vocabulary as the weight of each click. By doing so, a high-frequency word will weight less, such as misclick on common words such as "the" or "an", while a low-frequency word will weight more to emphasize on this action's importance. The weighted relative difficulty is calculated through Formula (4.2).

Same cross-validation method and estimators were performed using the new datasets where the weighted relative difficulty replaced the relative difficulty defined by Carver. Because the definition of the metrics changed, the comparison between the evaluation metrics such as RMSE or coefficient of determination is quite meaningless. Hence, only the percentage of the correct ranking estimation is compared. The percentage indicates the number of users' rankings matches exactly as the model estimated over the total number of users' rankings. Table 6.4 shows the best ranking results among all the data

Table 6.4. The accuracy of ranking the articles using the linear regression model, Collaborative Filtering with Carver's Relative Difficulty, and Collaborative Filtering with the Weighted Relative Difficulty

Algorithm	Correctness
Linear Regression	50%
Carver's Relative Difficulty KNNBaseline	80%
Weigthed Relative Difficulty KNNBaseline	90%

models. It can be easily recognized that the estimator using the weighted relative difficulty outperformed the estimator using Carver's relative difficulty.

Chapter 7

Conclusion

In this thesis, a system that can recommend articles to the users based upon each user's language level is proposed. A responsive web application with a translation annotation module was created. The application constantly tracks users' interactions with the system through the translation annotation module to determine their language levels. Carver's relative difficulty is adopted in this study which states that one way of measuring relative difficulty is through the percentage of unknown words that a user has towards a specific reading. Based upon the analysis of users' clicking, the system estimates the relative difficulty of each article and labels the relative difficulty of each reading in correspondence to each user's language level. Through multiple experiments using numerous algorithms, it is revealed that the proposed system is able to estimate the relative difficulty for each user towards each reading. The linear regression model was capable of giving the best estimation of Carver's definition of relative difficulty. While the KNNWithZScore is best recommendation algorithm which is evaluated through ranking the articles using the estimated relative difficulty and comparing the estimated ranking with user's own ranking of the difficulty of readings. Due to possible human errors and system malfunction that may have happened during the data collection, a modified relative difficulty called weighted relative difficulty, is proposed in this paper based on the original definition by Carver to reduce the noise in the data. Another experiment was conducted to determine whether the new measurement of the distance between a reading and a user can outperform the original definition by Carver. The result shows that the proposed weighted relative difficulty recommendation model outperforms the previous recommendation model.

7.1 Limitation

There are many limitations to this study. First, the data sample is quite small and also very possibly biased. Because only friends and families were invited to this study, it is difficult to assure a very balanced dataset. Second, it is a study that is quite hard to replicate the result to ensure its validity, due to the nature that the data sample others collect could be very different from the dataset used in this study. Third, the proposed estimator using collaborative filtering has the same problem that all other recommendation models have such as cold start meaning that it is very difficult for the proposed method to predict the relative difficulty of the readings that no one in the recommendation matrix has read.

7.2 Future work

Due to the relatively small data sample, it is very hard to state that currently trained prediction model can be used in real life. Enlarge the data sample is definitely necessary

for further research on the same topic using similar methods. For a trained prediction model to be used in real life, the problems that exist in the regular recommendation models such as cold start need also be solved. An experiment between learners with different native language using the same system can be conducted to find whether this method is applicable to people with different native language.

References

[1] Chen-Chung Chi, Chin-Hwa Kuo, and Chia-Chun Peng. The designing of a web page recommendation system for esl. In *Advanced Learning Technologies, 2007. ICALT 2007. Seventh IEEE International Conference on*, pages 730–734. IEEE, 2007.

[2] Herbert J Klausmeier, Richard A Rossmiller, and Mary Saily. Individually guided education: Concepts and practices. *New York: Academic Press*, 1977.

[3] Y. C Wu. The effects of repeated reading and text difficulty on fifth grade reading performance. *Bulletin of Educational Psychology*, 35(4):319–336, 2004.

[4] Pei-Lin Liu, Chiu-Jung Chen, and Yu-Ju Chang. Effects of a computer-assisted concept mapping learning strategy on efl college students' english reading comprehension. *Computers & Education*, 54(2):436–445, 2009.

[5] Mei-Hua Hsu. A personalized english learning recommender system for efls. *Expert Systems with Applications*, 34(1):683–688, 2008.

[6] Michael Heilman, Kevyn Collins-Thompson, Jamie Callan, and Maxine Eskenazi. Combining lexical and grammatical features to improve readability measures for first and second language texts. *The Human Language Technology Conference*, 2007.

[7] Yasuaki Shinohara. Audiovisual training effects for japanese children learning english /r/-/l/. *Interspeech*, pages 204–207, 2016.

[8] Dean Luo, Ruxin Luo, and Lixin Wang. Naturalness judgement of l2 english through dubbing practice. *Interspeech*, pages 200–203, 2016.

[9] Chin-Hwa Kuo and Chen-Chung Chi. Designing a reading material recommendation system for efl learners. *Journal of Applied Science and Engineering*, 17(4):371–382, 2014.

[10] Sarah E Schwarm and Mari Ostendorf. Reading level assessment using support vector machines and statistical language models. *The 43rd Annual Meeting on Association for Computational Linguistics*, 2005.

[11] Ching-Kun Hsu, Gwo-Jen Hwang, and Chih-Kai Chang. Development of a reading material recommendation system based on a knowledge engineering approach. *Computers & Education*, 55(1):76–83, 2010.

[12] Gwo-Jen Hwang, Pei-Shan Tsai, Chin-Chung Tsai, and Judy C. R. Tseng. A novel approach for assisting teachers in analyzing student web-searching behaviors. *Computers & Education*, 51(2):926–938, 2008.

[13] Newsela. `https://newsela.com/`, 2018.

[14] Ronald P Carver. Percentage of unknown vocabulary words in text as a function of the relative difficulty of the text: Implications for instruction. *Journal of Reading Behavior*, 26:413–437, 1994.

[15] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. *the Fourteenth International Joint Conference on Artificial Intelligence*, (12):1137–1143, 1995.

[16] Coursera. `https://www.coursera.org/`, 2018.

[17] Udemy. `https://www.udemy.com/`, 2018.

[18] Teachable. `https://teachable.com/`, 2018.

[19] Ching-Kun Hsu, Gwo-Jen Hwang, and Chih-Kai Chang. A personalized

recommendation-based mobile learning approach to improving the reading performance of efl students. *Computers & Education*, 63:327–336, 2013.

[20] Ah-Hwee Tan and Christine Teo. Learning user profiles for personalized information dissemination. In *Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on*, volume 1, pages 183–188. IEEE, 1998.

[21] Jeanne Sternlicht Chall and Edgar Dale. *Readability Revisited: The New Dale-Chall Readability Formula*. Brookline Books, Cambridge, MA, 1995.

[22] Jeanne Sternlicht Chall and Edgar Dale. A formula for predicting readability. *Educational Research Bulletin*, 27(1), 1948.

[23] J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. Derivation of new readability formulas for navy enlisted personnel. *Branch Report*, pages 8–75, 1975.

[24] A. Jackson Stenner. Measuring reading comprehension with the lexile framework. *In Fourth North American Conference on Adolescent/Adult Literacy*, 1996.

[25] Yi-Ting Huang, Hsiao-Pei Chang, Yeali Sun, and Meng Chang Chen. A robust estimation scheme of reading diculty for second language learners. *Advanced Learning Technologies (ICALT)*, 2011.

[26] Ildiko Pil'an, Sowmya Vajjala, and Elena Volodina. A readable read: Automatic assessment of language learning materials based on linguistic complexity. *International Journal of Computational Linguistics and Applications*, 2016.

[27] Rudolf Franz Flesch. *How to write plain English: A book for lawyers and consumers*. Harpercollins, 1979.

[28] Seed. `https://lexile.com/`, 2016.

[29] Kazuyo Yoshimura, Koichi Kise, and Kai Kunze. The eye as the window of the language ability: Estimation of english skills by analyzing eye movement while reading documents. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 251–255. IEEE, 2015.

[30] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer, 2011.

[31] Hsin-chou Huang, Chiou-lan Chern, and Chih-cheng Lin. Efl learners' use of online reading strategies and comprehension of texts: an exploratory study. *Computers & Education*, 52(1):13–26, 2009.

[32] Tamas Makany, Jonathan Kemp, and Itiel E Dror. Optimising the use of note-taking as an external cognitive and for increasing learning. *British Journal of Educational Technology*, 40(4):619–635, 2009.

[33] Itiel Dror and Stevan Harnad. Offloading cognition onto cognitive technology. *Distributed cognition: How cognitive technology extends our minds*, pages 1–23, 2008.

[34] Ann L Brown and Sandra S Smiley. The development of strategies for studying texts. *Child Development*, 49(4):1076–1088, 1978.

[35] Ann M Quade. An assessment of retention and depth of processing associated with notetaking using traditional pencil and paper and an on-line notepad during computer-delivered instruction. 1996.

[36] Tiffany Ya Tang and Gordon McCalla. Smart recommendation for an evolving e-learning system. *International Journal on E-Learning*, 4(1), 2005.

[37] Viet Anh Nguyen, Van Cong Pham, and Si Dam Ho. A context-aware mobile learning adaptive system for supporting foreigner learning english. In *Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2010 IEEE RIVF International Conference on*, pages 1–6. IEEE, 2010.

[38] Igor Jugo, Božidar Kovačić, and Vanja Slavuj. Increasing the adaptivity of an intelligent tutoring system with educational data mining: A system overview. *International*

Journal of Emerging Technologies in Learning (iJET), 11(03):67–70, 2016.

[39] Stephen D Krashen. *Principles and practice in second language acquisition*. New York, 1987.

[40] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on Intelligent user interfaces*, pages 31–40. ACM, 2010.

[41] Jeanne Nakamura and Mihaly Csikszentmihalyi. The concept of flow. *Handbook of Positive Psychology*, 2002.

[42] Mihaly Csikszentmihalyi. Finding flow. *New York: Basic*, 1997.

[43] Mihaly Csikszentmihalyi. Beyond boredom and anxiety. *Handbook of Positive Psychology*, 1975.

[44] Kindle. `https://www.amazon.com/Kindle-eBooks/b?ie=UTF8&node=154606011`, 2018.

[45] Seed. `http://a.app.qq.com/o/simple.jsp?pkgname=com.seed.app`, 2018.

[46] Dataset obtained in this research. `https://github.com/myt588/research-data`, 2018.

[47] K12reader. `http://www.k12reader.com/`, 2016.

[48] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. *the 20th International Conference on Neural Information Processing Systems*, pages 1257–1264, 2007.

[49] Stanton A. Glantz and Bryan K. Slinker. *Primer of Applied Regression and Analysis of Variance*. McGraw-Hill, 1990.

Acknowledgements

I would like to express my special thanks of gratitude to my professors (Prof. Honiden Shinichi, Prof. Tei Kenji, Prof. Sakamoto Kazunori, and Prof. Chiba Shigeru) who gave me the opportunity to explore freely on any projects, which allows this research to be conducted. I would like to thank Prof. Honiden for continuing his supervising on me even after his retirement from the University of Tokyo. I would like to thank Prof. Tei for helping me finalize this project and advising me on my research skills. I would like to thank Prof. Sakamoto for designing the proposed methods with me and offering me his extensive knowledge on machine learning. I would like to thank Prof. Chiba for being my official advisor at the University of Tokyo after the retirement of Prof. Honiden.

Secondly, I would also like to express my sincere gratitude to all the participants and especially to those (Shen Xin, Kang Yi) who have helped invite their friends to join this research. I would like to express my gratitude to Juku Momo, a growing tutoring center located in Saitama, for allowing me to conduct the data collection on the students there. Without the help from all my friends and my friends' friends, this research would not be possible. All the participants' efforts are especially important to this research and to me personally. I would like to once again thank all of them to be a part of this research. Thank you.

Finally, I would like to thank the University of Tokyo for having me conduct research here and providing many resources for research purpose. It is a great honor to be a member of the Graduate School of Information Science and Technology at the University of Tokyo.

A

An example of a validated user's dataset obtained in this research


```
{
  "user": {
    "_id": 10,
    "username": "MY",
    "profile": {
      "gender": "Male",
      "age": "25-34",
      "grade": "Master Degree",
    }
  },
  "count": 7,
  "readings": [
    {
      "readingId": 1,
      "count": 2,
      "unknonws": [
        "lexileDifficulty",
        "lexileDifficulty"
      ]
    },
    {
      "readingId": 2,
      "count": 3,
      "unknonws": [
        "council",
        "individuals",
        "mayor"
      ]
    },
    {
      "readingId": 3,
      "count": 16,
      "unknonws": [
        "accomplished",
        "Cherokee",
        "Islanders",
```

```
      "Cheyenne",
      "Torres",
      "Wounded",
      "devastating",
      "surrendered",
      "tribes",
      "Seminole",
      "Choctaw",
      "treaty",
      "abused",
      "endured",
      "Indigenous",
      "Trail"
    ]
  },
  {
    "readingId": 4,
    "count": 9,
    "unknonws": [
      "particular",
      "trading",
      "surplus",
      "an",
      "tribes",
      "ability",
      "had",
      "surplus",
      "tribes"
    ]
  },
  {
    "readingId": 5,
    "count": 3,
    "unknonws": [
      "question",
      "justify",
      "neat"
    ]
  },
  {
    "readingId": 6,
    "count": 5,
    "unknonws": [
      "diverse",
      "shooting",
      "aware",
      "habitat",
      "swamp"
    ]
  },
  {
```

```
      "readingId": 7,
      "count": 20,
      "unknonws": [
        "liberation",
        "comrades",
        "tracts",
        "starving",
        "apparatus",
        "soil",
        "Parliament",
        "Fleet",
        "subjugated",
        "outlive",
        "convince",
        "tyranny",
        "growing",
        "utmost",
        "odious",
        "invasion",
        "Majesty",
        "menace",
        "confidence",
        "orator"
      ]
    }
  ]
}
```