– Independent Work Report Spring 2021 –

CH-ENG Reader: A Tool for Guided Chinese Reading through Parallel English Accompaniment

Suki Yip
Adviser: Robert Dondero

Abstract

Like other languages, Chinese requires constant practice and exposure to learn. Chinese web novels, a popular form of online media, come in a plethora of genres that provide readers a great way to understand Chinese culture and contemporary usages of the language. While there are online communities that translate these novels for English readers, there is currently no tool that puts the English version together with the original text. By aligning the two texts together at the paragraph level and providing Google Translate in an easily accessible form, CH-ENG is an online web-novel-reading application where English readers can not only appreciate the original Chinese text, but also have a helpful aid to improve their reading ability. Serving also as a resource for finding books of the appropriate reading level, this tool attempts to make reading Chinese novels in their raw form more approachable. For Chinese learners who are novel readers, this would not only help improve their language proficiency, but also allow them to later tap into the vast untranslated selections available online.

1. Introduction

1.1. Motivation: Problem Description

Chinese web novels are books primarily or solely published online. However, unlike traditional books whose contents are published all at once, they are often written and released chapters at a time. Due to their accessibility and convenience, web novels have become a popular medium of literature in China, with revenues nearing 2.5 billion [13]. These web novels often come in a large variety of genres, ranging from historical to contemporary slice-of-life. Although much is fiction, much can be learned about Chinese history and culture through reading these works. Additionally, the raw text provides a great opportunity to learn contemporary Chinese, especially modern slang and idioms. However, it may be difficult for non-proficient Chinese readers to begin reading these novels, as they may contain many characters or phrases that are not commonly taught in school. Luckily, with the growth of Chinese web novels, communities of translators have emerged that translate these novels into their English equivalent. While many of these translations are "unofficial" (free fan translations not commissioned by a publisher), they can be a great reference for those struggling to read the raw Chinese text just by itself.

However, there are currently no existing tools that bring Chinese web novels together with their English translations. Web novel applications usually focus on either the Chinese or English market, not both. For Chinese learners reading these novels in Chinese for the first time, they often have a separate window open with Google Translate to look up the vocabulary. However, this breaks the reading experience; an activity that should be relaxing would instead become annoying and tedious, as they would constantly have to copy and paste words into Google. An alternative some readers use is the Google Translate extension tool, which allows them to highlight the Chinese text for an instant translation. However, these translations do not account for the context that the text came from, which can impact the meaning of what's highlighted.

In addition to these inconveniences, some texts are noticeably harder to read than others. From the characters used to the phrasing, it may be difficult for readers to know what book they should start with or read next. There are not many online guides available for this either, as most applications recommend books by genre or user ratings, not by reading difficulty. It can be discouraging for a new reader to start with novels much above their reading level. Hence, while there is much potential in reading web novels for Chinese learning, the tedious work and discouragement that non-proficient Chinese readers may face could make the process and option less approachable.

1.2. Goal

The goal of this independent work is to develop an application that taps into the potential of web novels as a language-learning tool. The primary target users are advanced Chinese learners who already read web novels in English. They have a foundation in Chinese and are seeking to improve by reading the original text of these novels. As a platform built with the learning aspect in mind, there will be three main points of attention:

1. Paragraph-Aligning Chinese and English Texts Together
2. Providing Google Translation as a Convenient Aid
3. Building Reading Sequences by Reading Difficulty

As mentioned previously, no tools currently display both the Chinese and English texts together. As such, the app seeks to aid these users with parsing the unfamiliar Chinese text by providing a companion side-by-side English translation as a reference. By aligning the texts together on the paragraph level, users can easily refer to the same paragraph in either language. This is intended to minimize breaking the reading experience and provide a translated text where context is accounted for. Recognizing that Google Translate can still serve as a great supplemental tool for looking up specific words or phrases, the secondary focus is to provide Google Translation in a one-click convenient format. Emphasis will be on making the experience less tedious and troublesome than it presently is. Finally, the last focus of the app will be on helping these readers develop a sequence of books to read by reading level. Serving as a guide to navigating the plethora of books, this tool will be designed to encourage readers to continue reading more novels in Chinese and to improve their reading proficiency such that they can gradually read more difficult books.

Additionally, this project recognizes the potential symmetrical opportunity of this tool, where Chinese readers who are learning English can also find a use for this application. As such, the secondary users of the application are English learners who have foundation in the language and are seeking to improve by having a Chinese companion text. In the app, whenever something is in English, there will also be a Chinese version and vice versa. However, Chinese learners will still be given priority, and this will be reflected in the app's design.

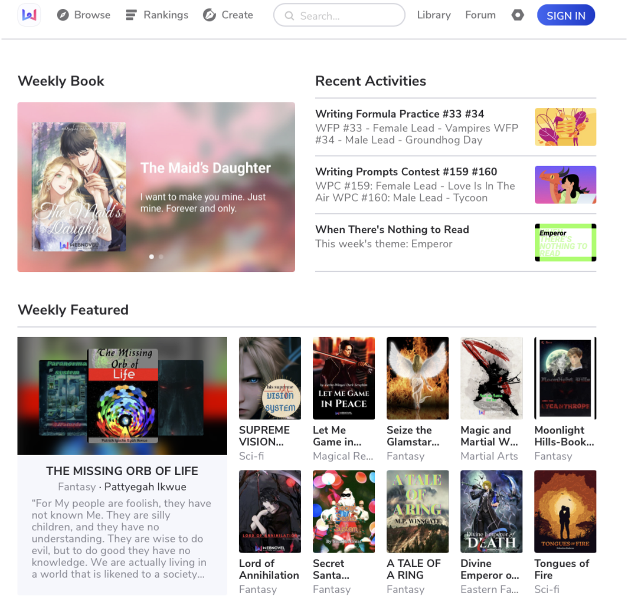
2. Related Work

For this project, there are three sources of related work:

1. Novel-Reading Websites
2. Websites with Parallel Texts
3. Google Translate Chrome Extension

2.0.1. Novel-Reading Websites

There are currently many apps for web novel reading. Most follow a similar format. This paper will use the browser version of Webnovel, an app that hosts translated English web novels, as an example. On the app, the first page that users often see is the Home page (Figure 1), which promotes new or popular novels. In addition to featuring books, this page usually displays a ranking chart with the top X books of popular genres. Additionally, apps will often have a library page, where users can browse through all the novels on the app with sorting and filtering features. In Webnovel's case, they have a "Browse" page to do this, as seen in the navigation bar of Figure 1.



The Maid's Daughter

I want to make you mine. Just mine. Forever and only.

Writing Formula Practice #33 #34
WFP #33 - Female Lead - Vampires WFP #34 - Male Lead - Groundhog Day

Writing Prompts Contest #159 #160
WPC #159: Female Lead - Love Is In The Air WPC #160: Male Lead - Tycoon

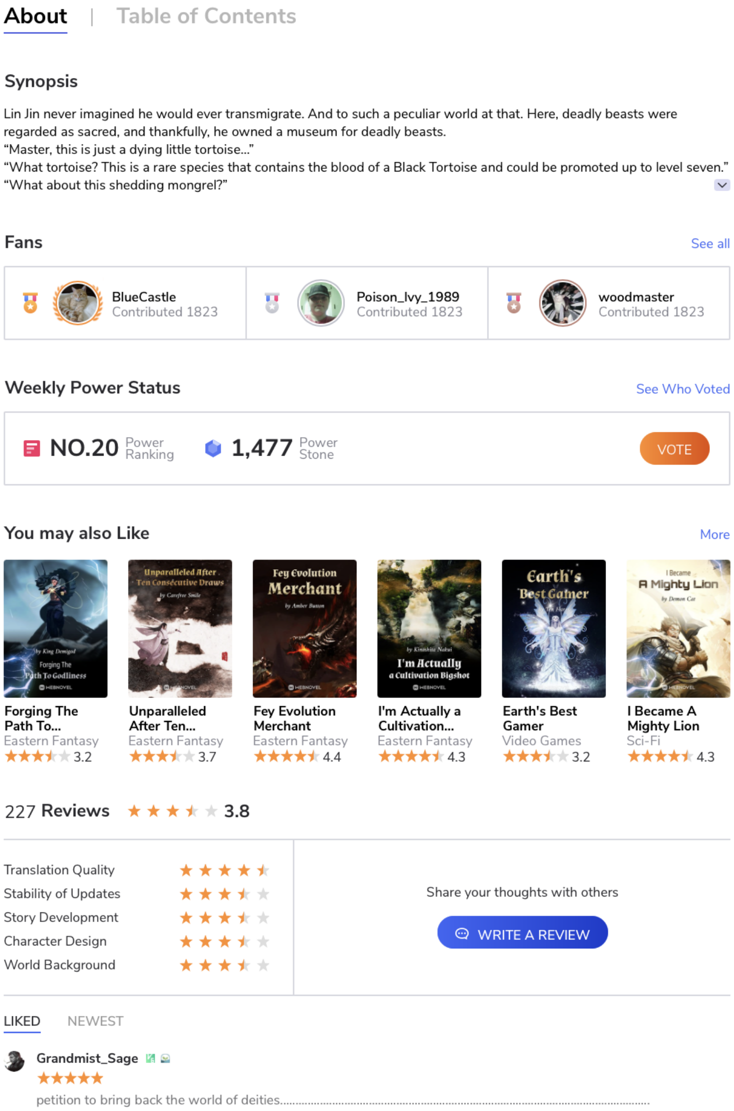
When There's Nothing to Read
This week's theme: Emperor

THE MISSING ORB OF LIFE
Fantasy · Pattyegah Ikwue
"For My people are foolish, they have not known Me. They are silly children, and they have no understanding. They are wise to do evil, but to do good they have no knowledge. We are actually living in a world that is likened to a society...

SUPREME VISION... Sci-fi	Let Me Game in... Magical Re...	Seize the Glamstar... Fantasy	Magic and Martial W... Martial Arts	Moonlight Hills-Book... Fantasy
Lord of Annihilation Fantasy	Secret Santa... Fantasy	A TALE OF A RING Fantasy	Divine Emperor o... Eastern Fa...	Tongues of Fire Sci-fi

Figure 1: Home page for Webnovel, which shows the featured books. [1]

Typically, clicking on a book cover will lead the user to a page with information just about the book, providing a brief description of the plot and a list of chapters. There is also usually a "You may also like..." section on this page. Here, apps recommend books to read next based on genre and plot similarities to the current book. Most also have a comments section, where users can rate the book and describe their opinion about it. Figure 2 is a snippet of a page about a novel, where the various sections just mentioned can be seen.



Synopsis

Lin Jin never imagined he would ever transmigrate. And to such a peculiar world at that. Here, deadly beasts were regarded as sacred, and thankfully, he owned a museum for deadly beasts.
“Master, this is just a dying little tortoise...”
“What tortoise? This is a rare species that contains the blood of a Black Tortoise and could be promoted up to level seven.”
“What about this shedding mongrel?”

Fans

BlueCastle Contributed 1823	Poison_Ivy_1989 Contributed 1823	woodmaster Contributed 1823

Weekly Power Status

NO.20 Power Ranking 1,477 Power Stone

You may also Like

Forging The Path To...	Unparalleled After Ten...	Fey Evolution Merchant	I'm Actually a Cultivation...	Earth's Best Gamer	I Became A Mighty Lion
Eastern Fantasy	Eastern Fantasy	Eastern Fantasy	Eastern Fantasy	Video Games	Sci-Fi
3.2	3.7	4.4	4.3	3.2	4.3

227 Reviews 3.8

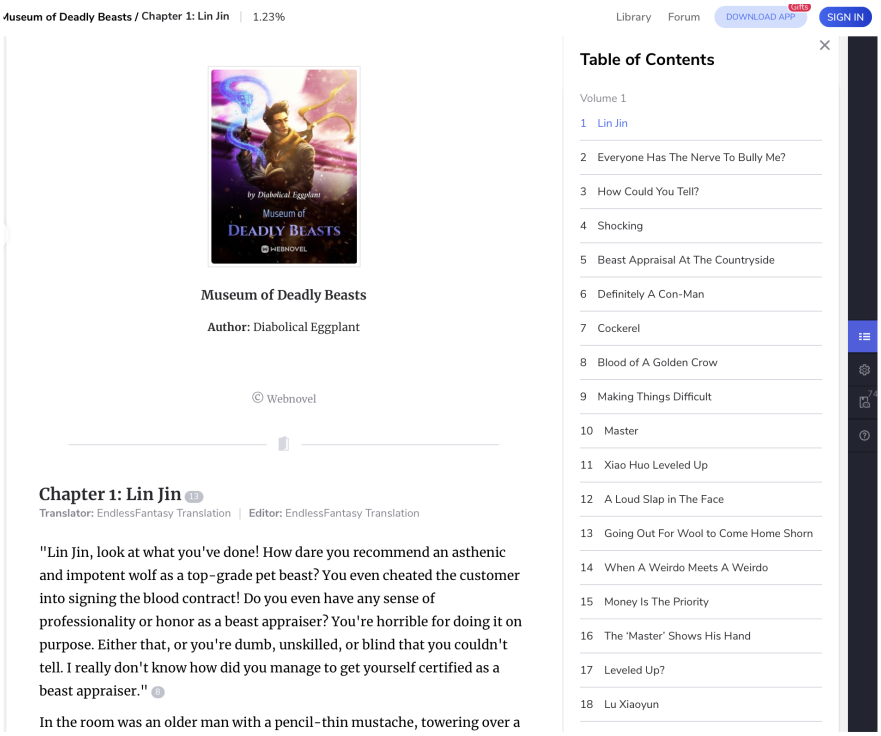
Translation Quality
Stability of Updates
Story Development
Character Design
World Background

Share your thoughts with others

Grandmist_Sage
petition to bring back the world of deities...

Figure 2: Snippet showing the information Webnovel provides in a page about a specific book. [1]

Beyond these pages, the most important of these apps is an interface where users can read the novels. These apps typically dedicate one page to each chapter of a book, providing a navigation menu with "next" or "previous" to go to different chapters. Figure 3 is how Webnovel structures their reading interface. They have a progress bar on top and a navigation bar to the left. On the navigation bar, the top button is for the chapter list, the second button for changing the text and background, the third for viewing the user comments for the chapter, and the last for visiting the "help" page. One can see that other than this navbar, the page is relatively simple, bringing focus to the text so that readers can just continue scrolling down. However, because of the one-column format, users can only read novels in only one language at a time.



Museum of Deadly Beasts

Author: Diabolical Eggplant

© Webnovel

Chapter 1: Lin Jin

Translator: EndlessFantasy Translation | Editor: EndlessFantasy Translation

"Lin Jin, look at what you've done! How dare you recommend an asthenic and impotent wolf as a top-grade pet beast? You even cheated the customer into signing the blood contract! Do you even have any sense of professionalism or honor as a beast appraiser? You're horrible for doing it on purpose. Either that, or you're dumb, unskilled, or blind that you couldn't tell. I really don't know how did you manage to get yourself certified as a beast appraiser."

In the room was an older man with a pencil-thin mustache, towering over a

Table of Contents

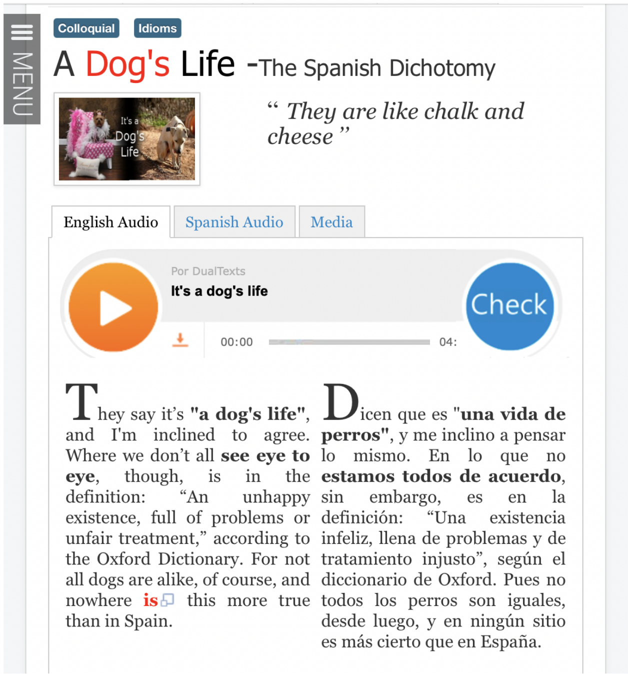
Volume 1

1 Lin Jin
2 Everyone Has The Nerve To Bully Me?
3 How Could You Tell?
4 Shocking
5 Beast Appraisal At The Countryside
6 Definitely A Con-Man
7 Cockerel
8 Blood of A Golden Crow
9 Making Things Difficult
10 Master
11 Xiao Huo Leveled Up
12 A Loud Slap in The Face
13 Going Out For Wool to Come Home Shorn
14 When A Weirdo Meets A Weirdo
15 Money Is The Priority
16 The 'Master' Shows His Hand
17 Leveled Up?
18 Lu Xiaoyun

Figure 3: The interface Webnovel uses for users to read the novels. [1]

2.0.2. Websites with Parallel Texts

There are also some examples of related applications that line up English text together with another language. One is a Spanish-English magazine called DualTEXTS. This website offers magazine articles with both an English and Spanish version side-by-side, aligned at the paragraph level. Figure 4 is an example of their interface for reading an article. Notably, on the page, they also have the option for users to hear the audio in both languages [19].



A Dog's Life -The Spanish Dichotomy

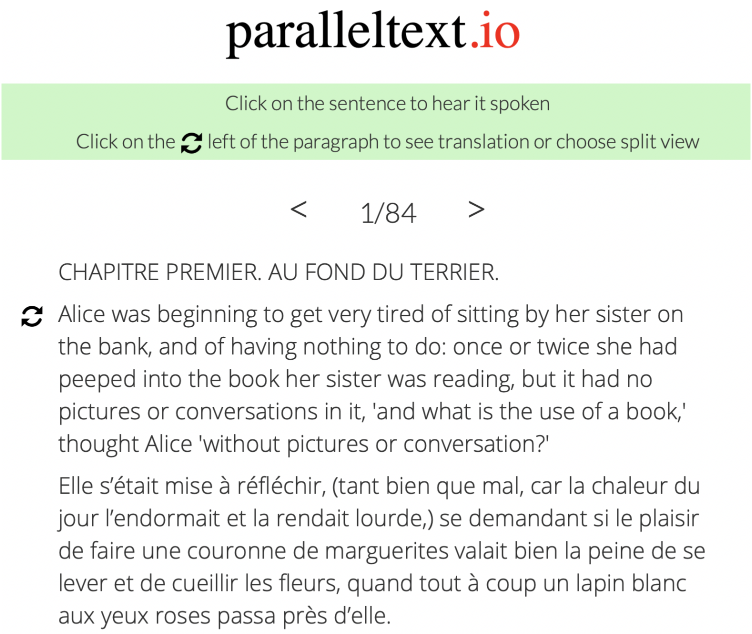
“ They are like chalk and cheese ”

They say it’s **"a dog's life"**, and I'm inclined to agree. Where we don’t all **see eye to eye**, though, is in the definition: “An unhappy existence, full of problems or unfair treatment,” according to the Oxford Dictionary. For not all dogs are alike, of course, and nowhere **is** this more true than in Spain.

Dicen que es **"una vida de perros"**, y me inclino a pensar lo mismo. En lo que **estamos todos de acuerdo**, sin embargo, es en la definición: “Una existencia infeliz, llena de problemas y de tratamiento injusto”, según el diccionario de Oxford. Pues no todos los perros son iguales, desde luego, y en ningún sitio es más cierto que en España.

Figure 4: DualTEXT interface for reading Spanish and English versions of an article side-by-side. [19]

Another example of an app that aligns texts of different languages together is paralleltext.io, which offers western classics like *Alice in Wonderland* or Sherlock Holmes books. Upon entering the site, users are prompted to fill in the sentence: "I speak [language options] and would like to get better at [language options]." After inputting two languages in, the website lists the different book options it has for those languages. After choosing a book, users are bought to the reading interface. This site offers two viewing modes. One is the split view, where the two language texts are side-by-side sentence-aligned. Clicking on a phrase would read aloud the selected text. Another viewing mode is the compact view, where it shows only the text in one language; for each sentence, there is a button to change the language individually. The displayed primary language is the language the user indicated that they want to get better at. For example, Figure 5 is the first chapter of *Alice in Wonderland* by Lewis Carroll. The languages entered in the beginning were English and French, respectively. In the figure, the text is in compact view. The text was originally all in French, but because the "switch" symbol (the cyclic icon) was clicked for the first sentence, the sentence turned into English. While this app is a great language-learning tool that uses the parallel text element, the language choices only include English, Polish, French, etc. It currently does not offer any texts in Asian languages. [2]



paralleltext.io

Click on the sentence to hear it spoken

Click on the ⟳ left of the paragraph to see translation or choose split view

< 1/84 >

CHAPITRE PREMIER. AU FOND DU TERRIER.

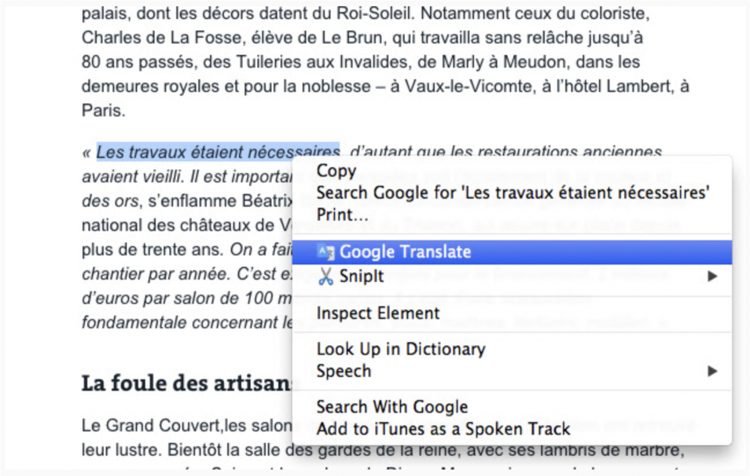
⟳ Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, 'and what is the use of a book,' thought Alice 'without pictures or conversation?'

Elle s'était mise à réfléchir, (tant bien que mal, car la chaleur du jour l'endormait et la rendait lourde,) se demandant si le plaisir de faire une couronne de marguerites valait bien la peine de se lever et de cueillir les fleurs, quand tout à coup un lapin blanc aux yeux roses passa près d'elle.

Figure 5: Paralleltext.io's compact view interface for English readers learning French. [2]

2.0.3. Google Translate Chrome Extension

As a widely-used translation tool, Google Translate has a Chrome extension that allows users to look up translations for the phrases they highlight [7]. First, users highlight the word or phrase they want to be translated. Then, when they right-click, they will see the option for "Google Translate" (Figure 6). Once they click on that, a popup will appear with the translation and audio option (Figure 7). However, while this tool is quite simple to use, it is an external tool that users have to download, and it is only available for Chrome users on desktop and Android. Additionally, getting one translation is a two-step process of having to first click on the "Google Translate" option after right-clicking. While it is only two steps, it is two steps per translation for a user who would probably need translations often.



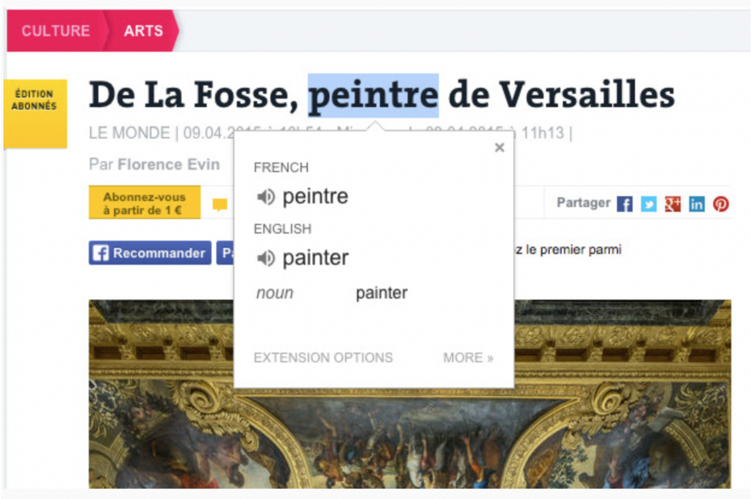
palais, dont les décors datent du Roi-Soleil. Notamment ceux du coloriste, Charles de La Fosse, élève de Le Brun, qui travailla sans relâche jusqu'à 80 ans passés, des Tuileries aux Invalides, de Marly à Meudon, dans les demeures royales et pour la noblesse – à Vaux-le-Vicomte, à l'hôtel Lambert, à Paris.

« *Les travaux étaient nécessaires, d'autant que les restaurations anciennes avaient vieilli. Il est importa*
des ors, s'enflamme Béatri
national des châteaux de V
plus de trente ans. *On a fai*
chantier par année. C'est e
d'euros par salon de 100 m
fondamentale concernant le

La foule des artisan

Le Grand Couvert,les salon
leur lustre. Bientôt la salle des gardes de la reine, avec ses lambris de marbre,

Figure 6: Imagine demonstrating how "Google Translate" is an option when right-clicking on a highlighted phrase [7].



De La Fosse, peintre de Versailles

LE MONDE | 09.04.2 … 11h13 |

Par Florence Evin

z le premier parmi

Figure 7: The resulting pop-up from the Google Translation extension [7].

3. Approach

Using the related works as inspiration, the approach of this project is to build a mobile-friendly web application for language learning through web novels. Called CH-ENG for "Ch"-inese and "Eng"-lish, the key novel idea of this application is to leverage the existing texts from novel-reading websites to generate parallel texts with both the Chinese and English versions. The Chinese texts will be from Qidian, one of the largest publishers of Chinese web novels [3]. The English texts will be from Webnovel, which is the example used in section 2. This is an online publisher for English translations of Chinese, Korean, and Japanese web novels. Owned by Qidian, Webnovel was chosen as the source of English texts because their translations are "official" and the origin of many of their English translations can be found on Qidian. For showcasing both Chinese and English texts, CH-ENG will draw inspiration from the aforementioned related works in parallel reading for the two-column format. However, the distinction is that the English texts were not made to be aligned with the Chinese text. CH-ENG will need backend processing to align the data.

As fundamentally a novel-reading application, CH-ENG will also draw inspiration from other online novel-reading websites for its interface. However, the distinction is that it will be a tool built for learning a language rather than for casual enjoyment by already fluent readers. Recommendations will be by reading level, and two languages, rather than one, will be displayed for each chapter. Additionally, CH-ENG will also offer Google Translate as an aid when reading. The key novel idea for this is to provide convenience with a self-contained translation aid. This means Google Translation will the built into the app so that users wouldn't need to have a specific device or browser. In addition, looking up words will be an even more convenient, one-step process; constantly needing a translation wouldn't be a problem anymore.

4. Functionality

4.1. Overview of Content and Main Features

CH-ENG consists of 6 different pages/types of pages:

1. The Home Page
2. The Features Page
3. The Library Page
4. Book Pages
5. The Recommendations Page
6. Chapter Reading Pages

Collectively, these pages allow CH-ENG to serve as both a novel reading application and a language learning tool with the following main features:

1. Paragraph-Level Alignment for Parallel Reading
2. Text Highlighting for Google Translate with Highlighting on Both Sides
3. Book Recommendations and Suggestions based on Reading Level
4. User Personalization with Google Sign-in

and the following sub-features:

1. Site-wide Navigation Bar
2. Sidebar for Chapter Navigation
3. Back-to-Top Shortcut

This section will go into detail about each page and how the features are integrated within.

4.1.1. The Home Page

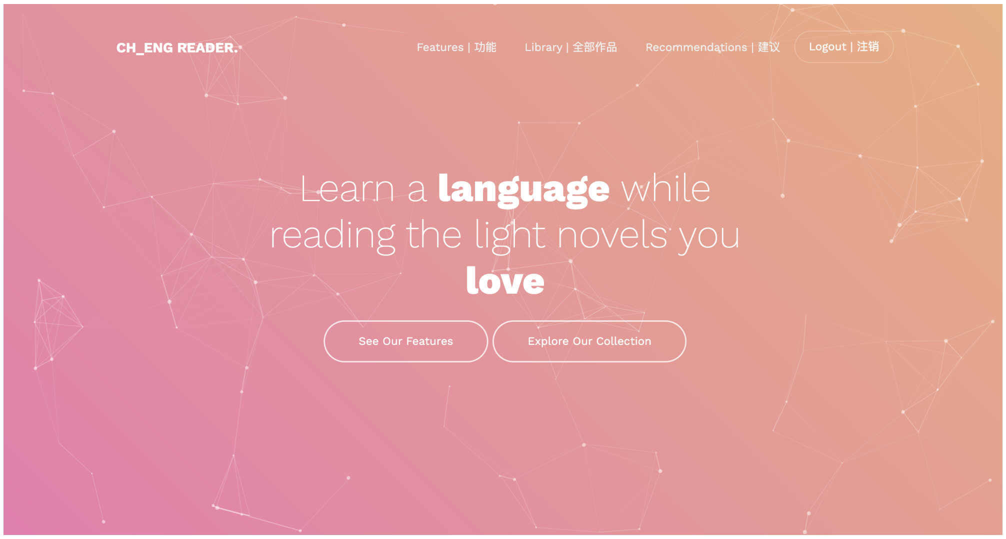
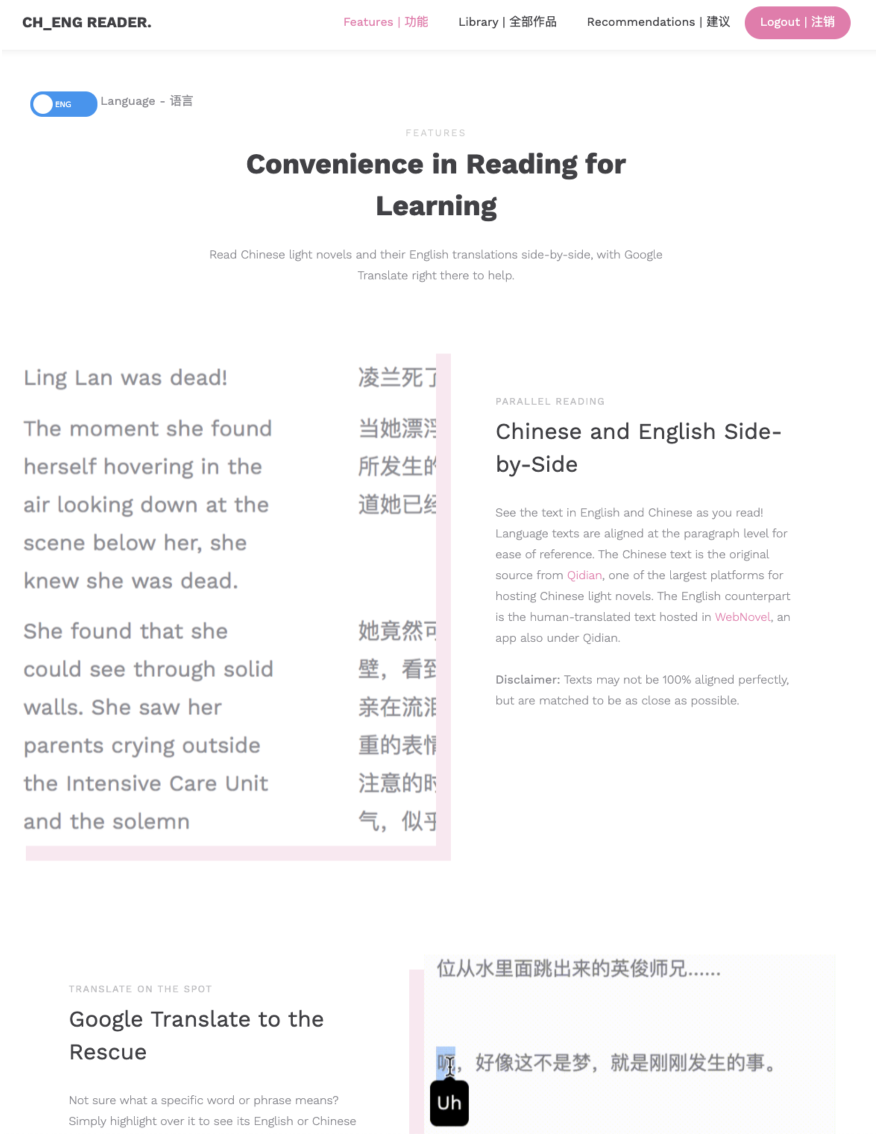


Figure 8: CH-ENG's Home page in English.

The Home page, which can be accessed through https://ch-eng-reader.herokuapp.com, is the first page new users are expected to see upon entering the app. As the introductory page, this page serves to give a short pitch or "mission-statement" about the app to intrigue the users. The language of this page is English by default. However, if the user has Chinese as their first browser language, then they will see the Chinese version. This system of basing the language off the browser settings differs from the rest of the app, where the Chinese and English texts are both available together. This exception was placed to preserve the aesthetics and simplicity of the page.

Figure 8 shows the English version of the Home page, where the slogan "Learn a language while reading the light novels you love" is displayed prominently in the middle. Below are two buttons: "See Our Features" and "Explore Our Collection." The former directs the user to the Features page (to be explained in section 4.1.2), while the latter takes the user to the Library page (to be explained in section 4.1.3).

4.1.2. The Features Page



CH_ENG READER.

FEATURES

Convenience in Reading for Learning

Read Chinese light novels and their English translations side-by-side, with Google Translate right there to help.

Ling Lan was dead!	凌兰死了
The moment she found herself hovering in the air looking down at the scene below her, she knew she was dead.	当她漂 所发生 道她已
She found that she could see through solid walls. She saw her parents crying outside the Intensive Care Unit and the solemn	她竟然 壁，看 亲在流 重的表 注意的 气，似

PARALLEL READING

Chinese and English Side-by-Side

See the text in English and Chinese as you read! Language texts are aligned at the paragraph level for ease of reference. The Chinese text is the original source from Qidian, one of the largest platforms for hosting Chinese light novels. The English counterpart is the human-translated text hosted in WebNovel, an app also under Qidian.

Disclaimer: Texts may not be 100% aligned perfectly, but are matched to be as close as possible.

TRANSLATE ON THE SPOT

Google Translate to the Rescue

Not sure what a specific word or phrase means? Simply highlight over it to see its English or Chinese

位从水里面跳出来的英俊师兄......

呃，好像这不是梦，就是刚刚发生的事。

Figure 9: CH-ENG's Features page in English.

Figure 8 shows the top half of the Features page in English. The Features page details the main features of the app with a picture example for each. This page comes in two languages, and the language choice is controlled by a toggle on top. The purpose of having two languages is to account for the symmetry-aspect mentioned in section 1.2, where users are Chinese and English learners. For this page, only one language is shown at a time (as opposed to showing both together) because there are already many words on the page for one language. Placing two versions together would have made the page busier and more cluttered. English is the default state of the toggle since the primary users are English-fluent. The toggle is blue for English and orange for Chinese.

4.1.3. The Library Page

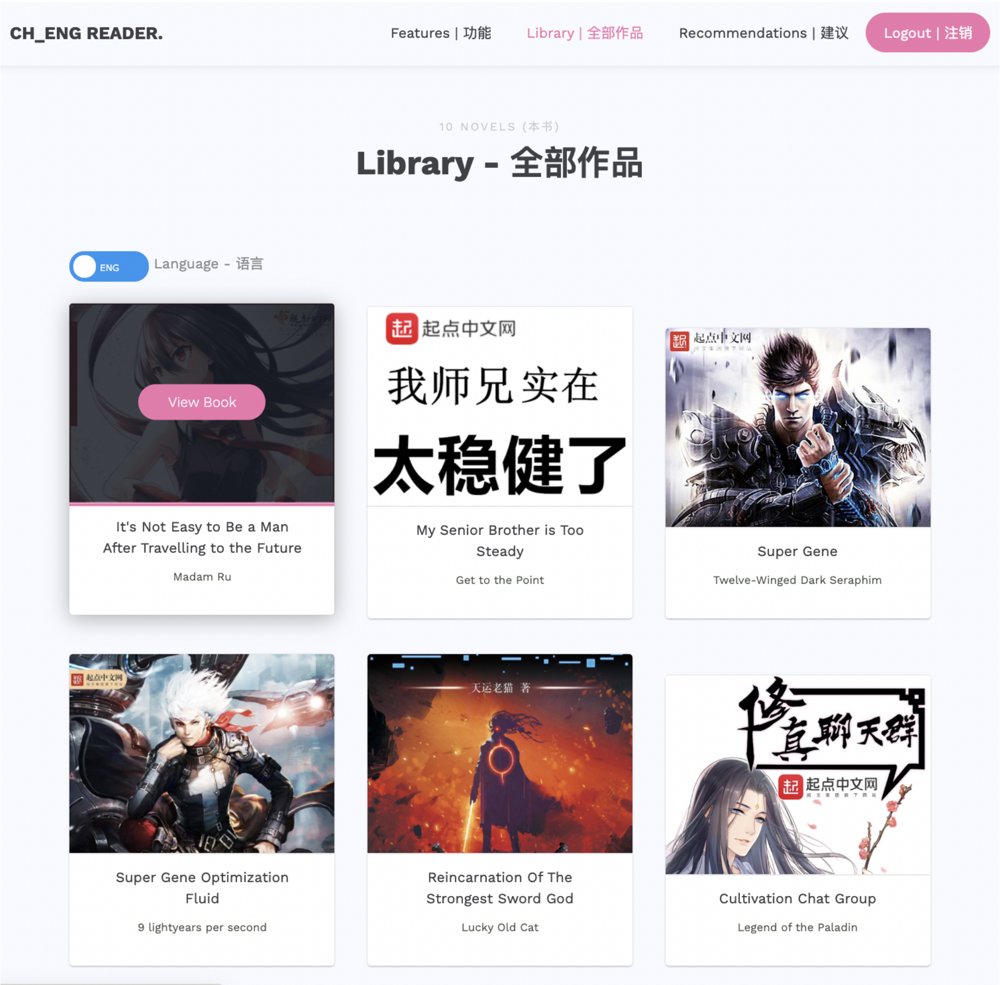
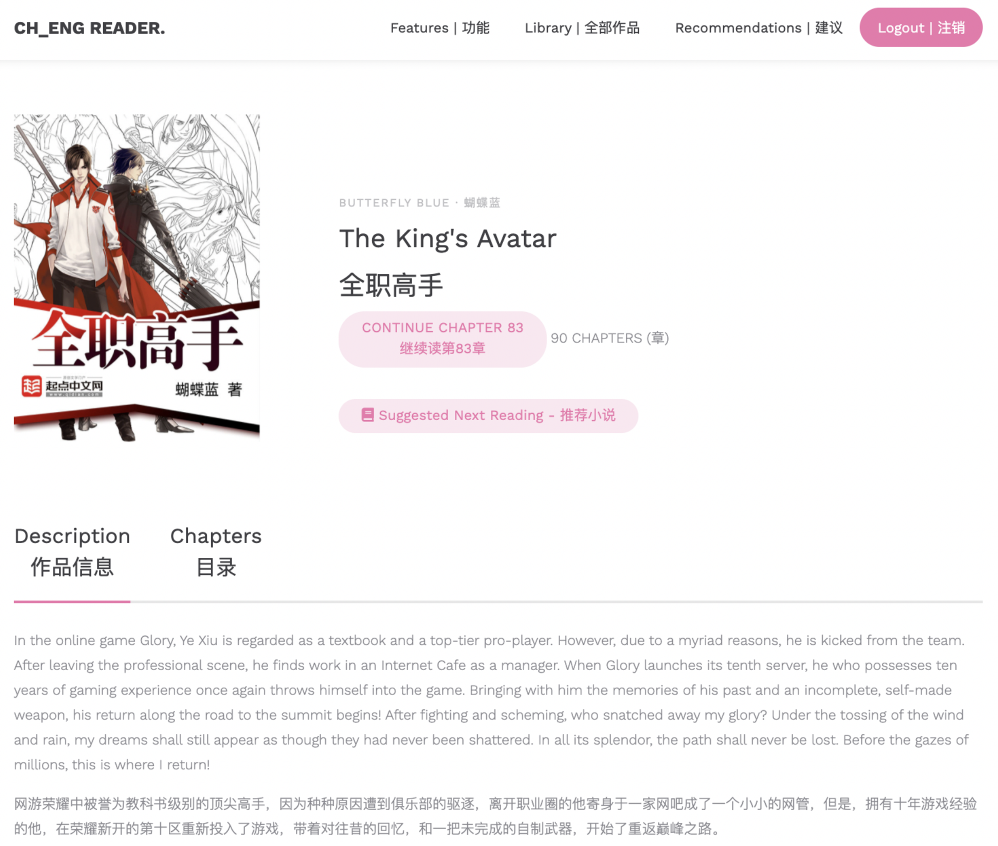


Figure 10: CH-ENG's "Library" page, which features all the books the site offers.

The Library page, shown in Figure 10, is where the user can see all the available books in the app. Each book is a "card," and hovering over one would make a "View Book" button appear inside. Clicking on the button would take the user to a page with more specific information about the book (a Book Page, to be explained in section 4.1.4). Figure 10 shows the state where a user is hovering over the first book. Users can also go to the same Book page by clicking on the title. On each card is the Chinese cover of the novel, followed by the title and author. Like the Features page, there is a language toggle at the top of the page, which changes the language of the cards. This is to conserve space by making the information displayed less "cluttered." In the Chinese version, the English title will be changed to the Chinese title, the author's English pen name will be changed to their Chinese pen name, and "View Book" will become its Chinese translated equivalent. Like the previous toggle, this toggle is set to English by default.

4.1.4. Book Pages

Each book has a page dedicated to displaying information about it, and this paper will refer to each as a Book page. The format of the Book page is the same across all books. It can be divided into three sections. Figure 11 shows the top section of a sample Book page. In the top section, the Chinese cover of the book is featured prominently next to the title and author (in both Chinese and English). Directly below the text is a button that either says "Read Chapter 1" or "Continue Chapter *x*," where *x* is some integer greater than 1. This button is part of the user personalization feature, which will be explained in section 4.1.7. Next to this button is the number of total chapters the book has. Under this is a button called "Suggested Next Reading," which is the title of the bottom section of the page. When a user clicks on this, they will be brought to the bottom section of the page. The purpose of this button is to ensure that users wouldn't miss the bottom section if they don't scroll down enough.



BUTTERFLY BLUE · 蝴蝶蓝

The King's Avatar

全职高手

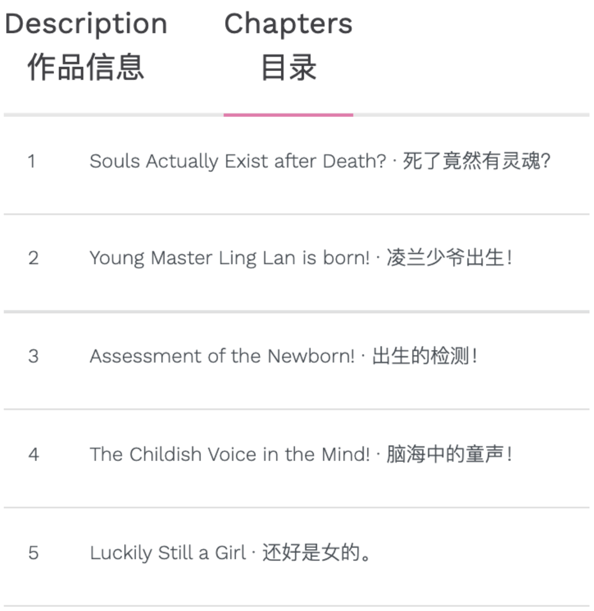
90 CHAPTERS (章)

In the online game Glory, Ye Xiu is regarded as a textbook and a top-tier pro-player. However, due to a myriad reasons, he is kicked from the team. After leaving the professional scene, he finds work in an Internet Cafe as a manager. When Glory launches its tenth server, he who possesses ten years of gaming experience once again throws himself into the game. Bringing with him the memories of his past and an incomplete, self-made weapon, his return along the road to the summit begins! After fighting and scheming, who snatched away my glory? Under the tossing of the wind and rain, my dreams shall still appear as though they had never been shattered. In all its splendor, the path shall never be lost. Before the gazes of millions, this is where I return!

网游荣耀中被誉为教科书级别的顶尖高手，因为种种原因遭到俱乐部的驱逐，离开职业圈的他寄身于一家网吧成了一个小小的网管，但是，拥有十年游戏经验的他，在荣耀新开的第十区重新投入了游戏，带着对往昔的回忆，和一把未完成的自制武器，开始了重返巅峰之路。

Figure 11: Example of the top of the Book page for a sample book.

The middle section features two tabs that users can click on: "Description" and "Chapters." Figure 11 shows an example of the "Description" tab. "Description" is the default tab, and it provides a summary of the novel in both English and Chinese. Both descriptions may not necessarily be the same, as each of these descriptions is from a different source; the English translator may have made their description different from the Chinese version. Clicking on the "Chapters" tab will show all the chapters of the book listed out in the following format: "[Chapter Number] [English Chapter Title] · [Chinese Chapter Title]." An example can be seen in Figure 12. Clicking on any of the chapters will bring the user to the page where they can read the contents of the chapter.



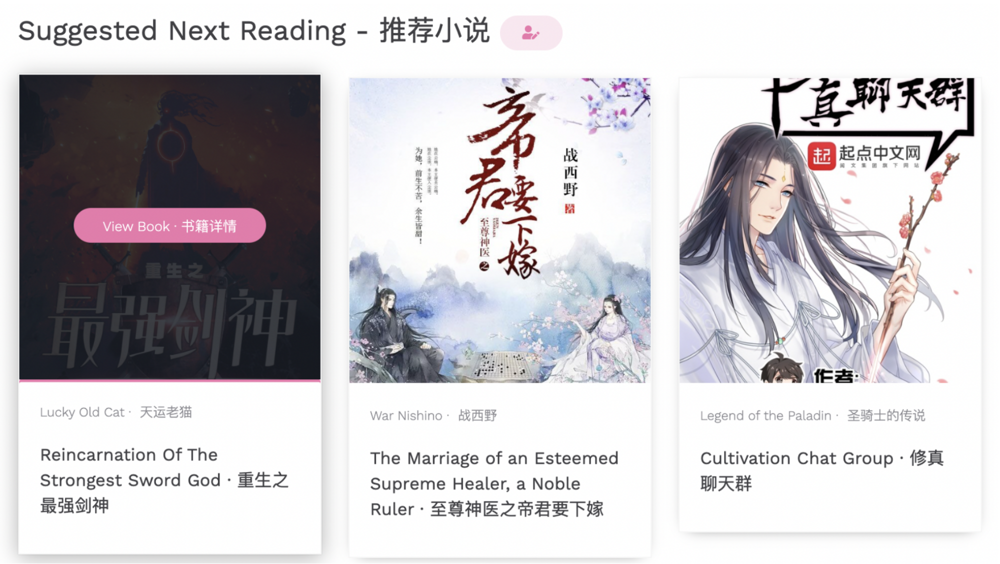
Description
作品信息

Chapters
目录

1	Souls Actually Exist after Death? · 死了竟然有灵魂?
2	Young Master Ling Lan is born! · 凌兰少爷出生！
3	Assessment of the Newborn! · 出生的检测！
4	The Childish Voice in the Mind! · 脑海中的童声！
5	Luckily Still a Girl · 还好是女的。

Figure 12: Example of the chapter list for a sample book.

As mentioned earlier, the bottom section is for the "Suggested Next Reading." This is one of the main features of the app. Here, the app recommends three books for the user to read next (in relation to the book of the page they are currently on) based on reading level. The algorithm behind this will be described in depth in section 5.3.3. An example can be seen in Figure 13. Like the Library page, the suggested books are in a "card" format. Users can go to the Book page of the corresponding book by either hovering over the card and clicking "View Book" or clicking on the title. Figure 13 shows the state where the user is hovering over the first book. Users can also give their own recommendations, which will be dynamically factored into the books the app suggests. From this page, users can give their recommendations by clicking on the icon with a person and pencil. Located next to the heading of the "Suggested Next Reading" section, it will say "Recommendations" when the user hovers over it. Clicking on this button will bring the user to the Recommendations page, which will be covered in the next section (4.1.5).



Suggested Next Reading - 推荐小说

Lucky Old Cat · 天运老猫

Reincarnation Of The Strongest Sword God · 重生之最强剑神

War Nishino · 战西野

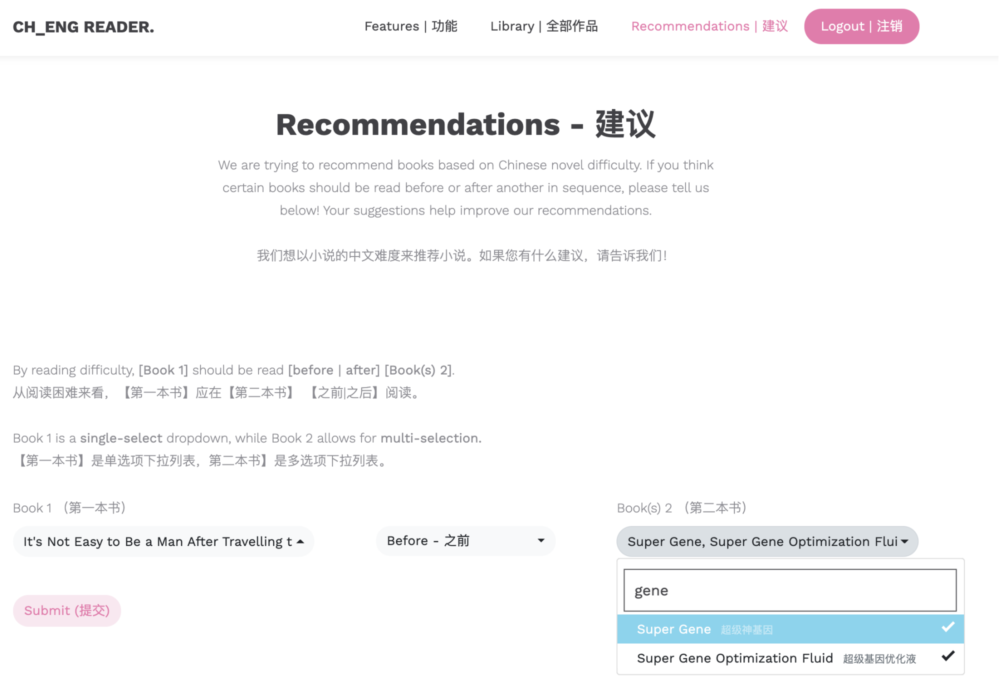
The Marriage of an Esteemed Supreme Healer, a Noble Ruler · 至尊神医之帝君要下嫁

Legend of the Paladin · 圣骑士的传说

Cultivation Chat Group · 修真聊天群

Figure 13: Example of the "Suggested Next Reading" for a sample book.

4.1.5. The Recommendations Page



Recommendations - 建议

We are trying to recommend books based on Chinese novel difficulty. If you think certain books should be read before or after another in sequence, please tell us below! Your suggestions help improve our recommendations.

我们想以小说的中文难度来推荐小说。如果您有什么建议，请告诉我们！

By reading difficulty, [Book 1] should be read [before | after] [Book(s) 2].
从阅读困难来看，【第一本书】应在【第二本书】【之前|之后】阅读。

Book 1 is a single-select dropdown, while Book 2 allows for multi-selection.
【第一本书】是单选项下拉列表，第二本书】是多选项下拉列表。

Book 1（第一本书）: It's Not Easy to Be a Man After Travelling t

Before - 之前

Book(s) 2（第二本书）: Super Gene, Super Gene Optimization Flui

gene

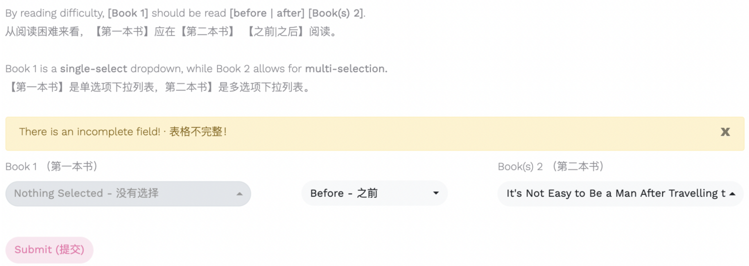
Super Gene 超级神基因 ✓

Super Gene Optimization Fluid 超级基因优化液 ✓

Figure 14: CH-ENG’s Recommendations page, where the user is filling the form.

The "Recommendations" page, shown in Figure 14, offers a way for users to give their own recommendations for "Suggested Next Reading." On this page is a form with three fields. The first field, labeled "Book 1," is a drop-down of all the book titles the app offers. The drop-down has smart search built-in, where the user can filter the titles by keywords. For this field, the user can only select one book. The third field, labeled "Book(s) 2," is also a drop-down like the first field, except users can select multiple books. A check-mark will appear next to the options the user selected, as shown in Figure 14. The second field is a drop-down that indicates the relation of "Book 1" to "Book(s) 2." There are two options: "Before" and "After." "Before" indicates that the user giving the feedback thinks that readers should read "Book 1" before they read those in "Book(s) 2" based on reading difficulty. "After" means that readers should read "Book 1" after those in "Book(s) 2." By default, "Before" is selected for convenience. In summary, the fields are made such that the user will fill in the following sentence: "By reading difficulty, [Book 1] should be read [before | after] [Book(s) 2]." The information collected from this form is used to dynamically update the books recommended in the "Suggested Next Reading" section for the relevant books. Detail on how exactly this data is factored in will be in section 5.3.3.

There are some checks the app does to validate the form inputs. Invalid inputs will cause a yellow alert banner to appear. The banner will automatically fade away after a few seconds, but it can be manually clicked away by the user too. Example invalid inputs include leaving one field empty when clicking the "Submit" button. This is shown in Figure 15. The app proactively ensures that the user does not select the same book in "Book 1" and "Book(s) 2." When the user clicks on a book for "Book 1," that book will no longer be an option for "Book(s) 2." Similarly, any book selected in "Book(s) 2" will be removed from the "Book 1" options. If the user selects all books in "Book(s) 2" while "Book 1" is still empty, then the app will throw a warning banner. When the form receives a valid submission, a green alert banner will appear with "Submission Successful!".



By reading difficulty, **[Book 1]** should be read **[before | after] [Book(s) 2]**.
从阅读困难来看，【第一本书】应在【第二本书】【之前|之后】阅读。

Book 1 is a **single-select** dropdown, while Book 2 allows for **multi-selection.**
【第一本书】是单选项下拉列表，第二本书】是多选项下拉列表。

There is an incomplete field! · 表格不完整！

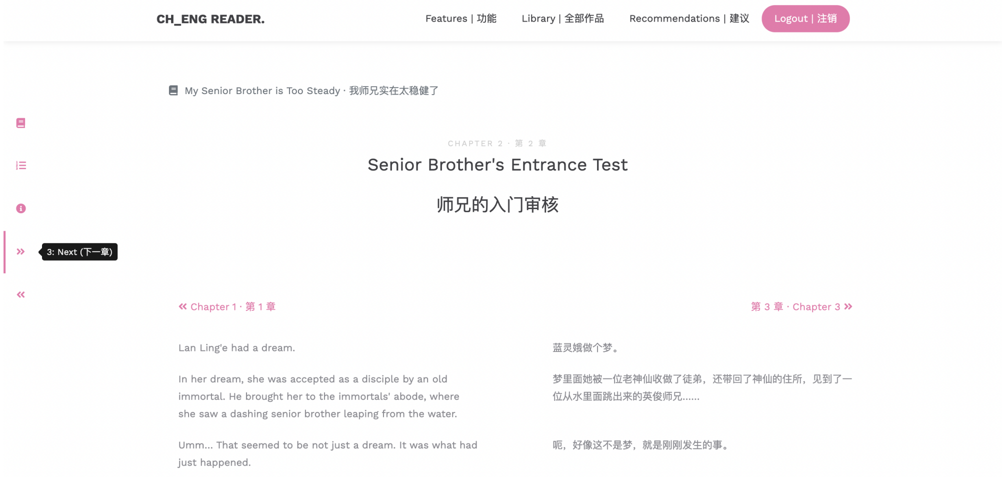
Book 1 （第一本书）: Nothing Selected - 没有选择

Before - 之前

Book(s) 2 （第二本书）: It's Not Easy to Be a Man After Travelling t

Figure 15: Example of a warning banner for the user recommendation form.

4.1.6. Chapter Reading Pages



My Senior Brother is Too Steady · 我师兄实在太稳健了

CHAPTER 2 · 第 2 章

Senior Brother's Entrance Test

师兄的入门审核

Lan Ling'e had a dream.

蓝灵娥做个梦。

In her dream, she was accepted as a disciple by an old immortal. He brought her to the immortals' abode, where she saw a dashing senior brother leaping from the water.

梦里面她被一位老神仙收做了徒弟，还带回了神仙的住所，见到了一位从水里面跳出来的英俊师兄……

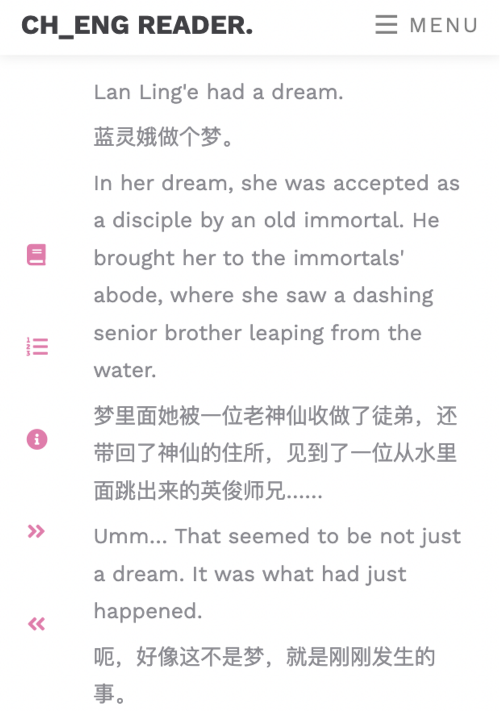
Umm... That seemed to be not just a dream. It was what had just happened.

呃，好像这不是梦，就是刚刚发生的事。

Figure 16: A sample Reading page for chapter 2 of a web novel.

Books are divided into chapters, and each chapter has its own web page that displays its contents. This paper will refer to each as a Reading page since these pages are designed for reading purposes. They all have the same interface. Figure 16 is a sample Reading page for chapter 2 of a web novel. Upon loading this page, users will probably first notice the chapter name centered on top in a big font (relative to the font size of other words on the page). The chapter name is in both Chinese and English. Just above the name is the chapter number, in a smaller font. Further above this, non-centered, is the book icon and the title of the novel. This is a link for the user to go to the Book page for the book.

Many of the core features of CH-ENG reside on this page. One is the paragraph-level alignment for parallel reading. The chapter contents are displayed in two columns, the English version on the left and the Chinese version on the right. The two texts are paragraph aligned, where each paragraph is vertically spaced out. In a narrow window where there isn't enough horizontal space for two columns, it will be one column of alternating English-Chinese paragraphs. This usually happens when users are reading on their phones. An example is shown in Figure 17.



Lan Ling'e had a dream.

蓝灵娥做个梦。

In her dream, she was accepted as a disciple by an old immortal. He brought her to the immortals' abode, where she saw a dashing senior brother leaping from the water.

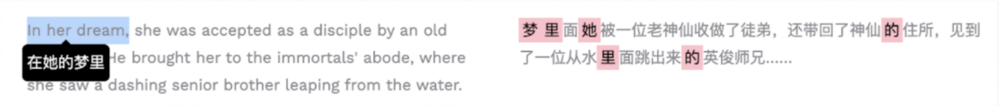
梦里面她被一位老神仙收做了徒弟，还带回了神仙的住所，见到了一位从水里面跳出来的英俊师兄......

Umm... That seemed to be not just a dream. It was what had just happened.

呃，好像这不是梦，就是刚刚发生的事。

Figure 17: A mobile view of the Reading page, where paragraphs are alternating in language.

Another main feature on this page is text highlighting for Google Translate. If the user highlights an English word or phrase, a tooltip will appear near the highlighted section with the Chinese translation. The tooltip is a small black popup with white text inside, as shown in Figure 18. Likewise, highlighting a Chinese word or phrase will show the English translation. This feature is for users who want to know the translation of a specific word or phrase in a paragraph. This feature helps make Google Translate easily accessible and convenient for users, as they only need to highlight with their mouse for the translation. A feature that supplements this is highlighting-on-both-sides. If the translation is in English, then every time a word in the translation appears in the corresponding English paragraph, the word will be highlighted. Similarly, if the translation is in Chinese, then every time a character in the translation appears in the Chinese paragraph, the character will be highlighted. For example, in Figure 18, because the translation is “在她的梦里," each time those characters appear in the corresponding paragraph, they are highlighted pink.



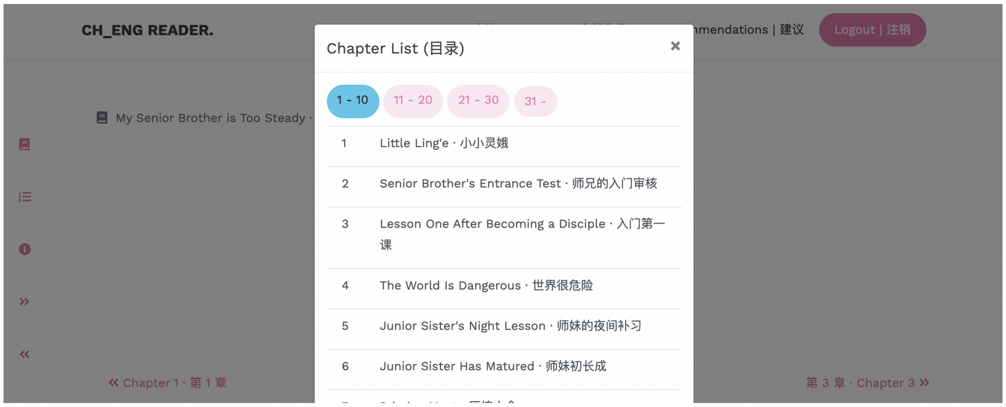
In her dream, she was accepted as a disciple by an old He brought her to the immortals' abode, where she saw a dashing senior brother leaping from the water.

在她的梦里

梦里面她被一位老神仙收做了徒弟，还带回了神仙的住所，见到了一位从水里面跳出来的英俊师兄……

Figure 18: Example of the tooltip showing after highlighting a phrase and the resulting highlighting-on-both-sides.

A sub-feature on the page is the sidebar, which is fixed to the left side of the page. This feature provides shortcuts that would make chapter navigation easier, allowing for a better reading experience. Typically, the sidebar will show five icons in the following order: book for "Description," list for "Chapter List," i for "information," right arrow for "Next," and left arrow "Previous." The icon descriptions are shown when the user hovers over the icons. For example, in Figure 16, the user is hovering over the right arrow, so " 3: Next (下一章) " appeared next to it as a tooltip. The 3 means that the next chapter is the third chapter.



Chapter List (目录)

1 - 10 | 11 - 20 | 21 - 30 | 31 -

1	Little Ling'e · 小小灵娥
2	Senior Brother's Entrance Test · 师兄的入门审核
3	Lesson One After Becoming a Disciple · 入门第一课
4	The World Is Dangerous · 世界很危险
5	Junior Sister's Night Lesson · 师妹的夜间补习
6	Junior Sister Has Matured · 师妹初长成

Figure 19: Sample chapter list after clicking the list icon on the sidebar.

The book icon will bring the user to the Book page for the book. The list icon will open a pop-up module with the list of chapter options. This is designed such that users don't have to leave the page to see the chapter list. An sample is shown in Figure 19. To account for long lists, the chapters are paginated such that ten chapters are displayed at a time. Buttons for each range are on the top. Note that the first-selected range correlates with the chapter the user is currently on. For example, if the user is on chapter 23, the "21-30" range will be the default selection when viewing the list. The information icon is a pop-up with instructions about the highlighting for translation feature. In particular, it talks about the limitations of the tooltip and how a user can fix it if it is misaligned.

The right arrow takes the user to the next chapter, and it will only appear if there is a next chapter. This means the last chapter of a book won’t have this button. Similarly, the left arrow takes the user to the previous chapter, and it will not appear on the first chapter of the book.

4.1.7. Site-Wide Features

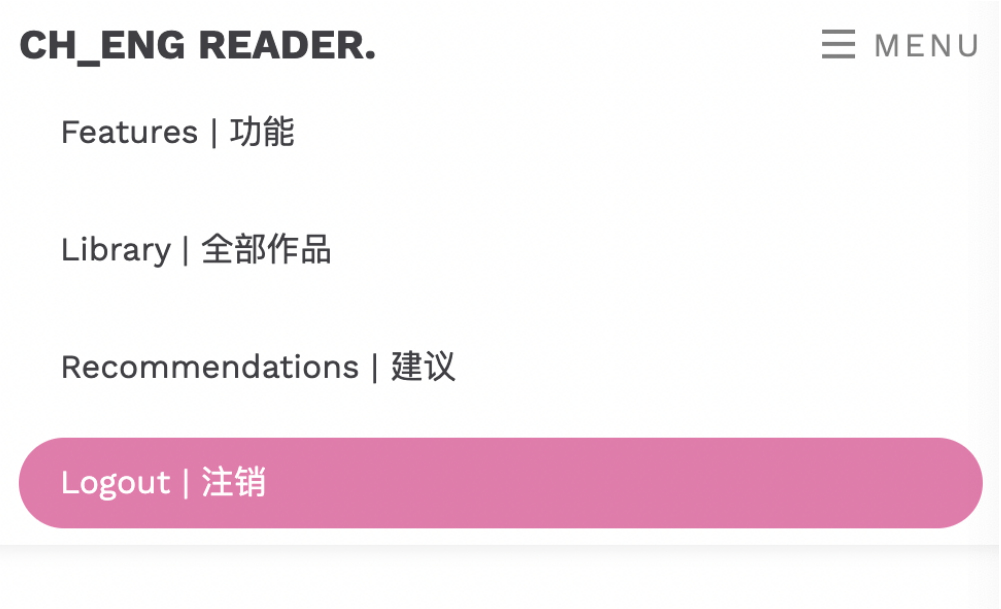


Figure 20: Navbar on a narrow window size.

Some features are always visible to the user, regardless of the page. One such feature is the site-wide navbar on the top, which adjusts according to the window size. Figure 20 shows how the navbar looks on a narrow window size. There are five available actions the user can do on the navbar. The first is to go back to the Home page, which can be done by clicking on the "CH_READER." logo. The second, third, and fourth actions are to go to the Features, Library, or Recommendations page by clicking on the respective name. The final action is to sign in to a Google account, which is part of the user personalization feature. This is a main feature, where the app saves the reading progress of users who are logged in through Google. When users click on the "Login" button in the navbar, they will be redirected to the Google interface to sign in. Upon signing in, the user will be redirected back to the page they were on originally in CH-ENG. After logging in, the "Logout" button will be displayed on the navbar, rather than "Login." After logging in, the app will now remember the last chapter for each book the user has read. On a Books page, a "Read Chapter 1" button is displayed if the user is not logged in or has not read the book before. The button will then direct the users to chapter 1. Note that if a non-logged-in user reads up to some chapter (not one) and goes back to the respective Book page, they will still see the "Read Chapter 1" button. A "Continue Chapter" button will be displayed with a number following it if the user is logged in and has read the book past chapter 1 on the site before. This number is the chapter the user last left off for the book. Clicking the button will lead the user to the chapter indicated.

A sub-feature that is also site-wide is the back-to-top shortcut, which is a button that appears on every page whenever the user scrolls down. Located on the bottom-right, it has an "up" icon, and clicking on it will automatically take the user back to the top of the page. This button is placed for user convenience when the pages are long, like the Reading pages.

4.2. Sample Use Cases

This section will review some sample use cases. Suppose Pat is a new user of CH-ENG and is learning Chinese at an advanced level in school. As a fan of reading English-translated web novels, she wants to improve her Chinese by reading the Chinese version of her favorite novels. She is given a link that leads her to the Home page. Suppose she wants to learn about this app and its features. Then, she can either click on "See Our Features" on the Home page or click on "Features" on the navbar. This will bring her to the Features page, where she can read about the main features of the app or look at the pictures for reference. Suppose she wants to know if the app has her favorite books. Then, she can click on "Library" in the navbar and scroll down the Library page to see if her books are there. Suppose, while browsing the books, Pat finds a book that piques her interest because the picture looks cool. She wants to read a summary of the book. Then, on the Library page, she can hover over the image of the book and click on "View Book." This will bring her to the Book page, where she can read the description under the respective tab.

Suppose that Pat is interested in this book after reading the description. She wants to read immediately! Then, from the Book page, she can either click on "Read Chapter 1" or go to the chapter list tab and select the first option. This will bring her to the Reading page, where both the Chinese and English texts will be displayed. As someone who wants to learn Chinese, she challenges herself to read the Chinese version, but encounters a phrase she doesn't know. From here, Pat can highlight the phrase for its translation. Suppose she thinks the English translation doesn't make sense given the context of the text. Then, she can try to find out where the phrase is in the corresponding English paragraph. She can do this by highlighting the unknown Chinese phrase and looking at the parts that are highlighted in pink in the English paragraph. She can try to deduce where is the phrase is by looking at where the pink highlights are clustered together. Alternatively, upon encountering the Chinese phrase, Pat could have just read the corresponding English paragraph to infer the meaning of the phrase.

Suppose Pat has read up to chapter 7 of this book. She quits the app for a reading break, but after going back on the site, she has forgotten what chapter she has read up to. She doesn't want to always have to remember what chapter she left off reading. To solve this problem, Pat can first log in to a Google account by clicking on "Login" in the navbar. Then, she can navigate back to the Book page of the book she was reading by going to the Library and clicking on the book again. Then, she can estimate the last chapter where she left off and click that on the chapter list. If she guessed wrong, then she can use the sidebar to easily go to the next or previous chapter until she finds chapter 7 again. If she goes back to the Book page, there will be a button that says "Continue Chapter 7." Now, when she reads, her progress will automatically be saved whenever she is logged in.

Suppose Pat has finished the book and wants a new book to read. She doesn't want to choose a book that would be too challenging compared to the book she just read. In that case, she can go to the Books page for that book and scroll down to the "Suggested Next Reading," where she will find suggestions for what to read. Suppose, after reading a couple of books on the app, Pat has her own thoughts about what books should be read before or after others. She wants to express her opinion. Then, she can go to the Recommendations page by clicking on "Recommendations" in the navbar. On the page, she can fill in the form and submit it. Now, her opinion will be part of the decision process when the app makes recommendations in the future.

4.3. Notable Design Decisions

Many design decisions had to be made in regards to the user interface and user experience of the app. One decision was on how to structure the Chinese-English symmetry of the app. As mentioned in section 1.2, the app is for Chinese and English users. Hence, the text in the app should be available in both languages. This proved to be difficult due to space constraints when considering overall aesthetics. In the early stages of development, the app had a language toggle on the navbar, where the toggle would control whether the text of the app is English or Chinese. The default language of the toggle was dictated by the user's first browser language and the app would remember the toggle's setting whenever it changed. However, from the first round of user evaluations (elaborated in section 6.0.3), the toggle proved to be confusing to use and users have expressed that they prefer the two languages to be together. Hence, the app was changed such that there would no longer be a site-wide language toggle, and all pages, besides the Home, Features, and Library pages, would have both texts displayed together.

In addition, due to the symmetry aspect, each book originally had two cover photos. One was the original Chinese cover in Qidian, and the other was the English cover used in Webnovel. The toggle would change the image according to its respective language. With the removal of the toggle, having two cover photos proved to be difficult when designing the app Additionally, users themselves commented that they didn't care about the English cover – just showing the original Chinese cover was sufficient. Hence, the app now only shows the Chinese cover of the books.

Another design decision was on adding the Recommendations page. Initially, in the "Suggested Next Reading" section, the "Recommendations" button would cause a pop-up modulo to show up. In the modulo, a form would show up that asks for next reading suggestions specific to the book featured on the current Books page. However, upon realizing that suggestions can happen in both ways, where a user could suggest a book to be before or after another, the form was changed to take, as input, two book fields and a field to indicate the direction of the relationship. The "Book(s) 2" field was made to be a multi-select drop down so that users wouldn't have to constantly fill the form if they wanted to select multiple books. The "Book 1" field does not allow multi-select, as having two multi-select fields would make the exact relationship the user is trying to indicate confusing. Lastly, the form now has a dedicated page because the form can be generalized to all books.

The sidebar for chapter navigation was also a feature added in a later iteration. Initially, the app only had "Next," "Previous," "Chapter List" hyperlinks at the top and bottom of the chapter contents. However, multiple users suggested on a sidebar during the first round of evaluations, so it was added.

Another design decision was on whether user sign-in should be optional or mandatory. Ultimately, it was made to be optional because the information from the sign-in is mainly to keep track of the user's reading progress. It is not necessarily for the language-learning features of the app and would simply be there as an option to enhance the user's convenience.

5. Implementation

Hosted on Heroku, the implementation of CH-ENG can be divided into three main components:

1. Text Gathering and Parsing
2. System Architecture and Design
3. Web Application Features

5.1. Text Gathering and Parsing

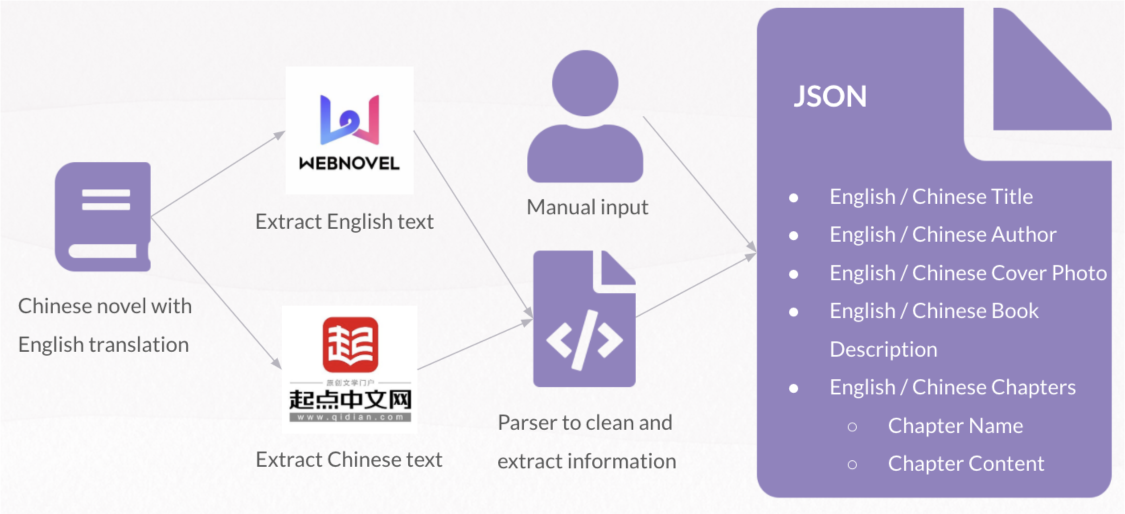

Chinese novel with English translation
WEBNOVEL
Extract English text
起点中文网
www.qidian.com
Extract Chinese text
Manual input
Parser to clean and extract information
JSON
English / Chinese Title
English / Chinese Author
English / Chinese Cover Photo
English / Chinese Book Description
English / Chinese Chapters
Chapter Name
Chapter Content

Figure 21: Diagram of the process to fetch and parse data on one book.

For the app to be possible, there had to be some collection of books that users could read. Each book needed to have the following information:

1. Title (in Chinese and English)
2. Author (in Chinese and English)
3. Cover Photo (Chinese and English versions)
4. Book Description (in Chinese and English)
5. Chapter Content (in Chinese and English)

Unfortunately, there was no one big dataset where all the information was readily available. Hence, for each book, the information had to came from aggregating data from different sources. The first step in this process was choosing which books to offer in the app. These books each needed to have an English version on Webnovel and a Chinese version on Qidian. In addition, the genres had to vary. Ultimately, ten novels were chosen, and their genres ranged from romance to science-fiction. It was for developmental purposes that only ten novels were chosen. The focus of this project was to develop the app features, so scaling the app to accommodate more books was not a priority.

For each book, the English text was extracted from Webnovel through a GUI web scrapper called Novel Grabber [14]. It is a tool that takes in the URL link of a book on Webnovel and converts the chapter contents of free chapters into an EPUB file. Along with the chapter content is also the English description of the book. For a more parse-able format, the file was converted into TXT by Zamzar, an online file converter [4]. Similarly, the Chinese text was extracted from Qidian by FictionDown, a tool that takes in a novel name and scrapes Qidian for the free chapter contents [10]. The information is then downloaded directly as TXT. In addition to the Chinese text, the file also contains the Chinese title, author, and book description.

Once the English and Chinese files are available, they are input into a parser written in Python. This parser cleans and exports the relevant information into an organized JSON file. It is in the parser that, for each chapter of the book, the English and Chinese paragraphs are matched together. The exact algorithm will be detailed in section 5.3.1. It is important to note that the number of chapters a book can have in CH-ENG is bound by the minimum number of free chapters available in Qidian and Webnovel.

Once the scrapped information is converted into the JSON file, several fields are still missing, namely the cover photos, English title, and English author. The title and author are manually inputted. The English and Cover cover photo is manually downloaded and stored on the Heroku file system. Then, the JSON will be updated with the file name of the images. The two cover images are from when the app still differentiated an English cover from a Chinese cover (referenced in section 4.3). While the English cover photo is currently not used, it is kept for potential future uses.

The data was formatted into JSON files because the data didn't have many applicable relational connections. Instead, the data was more suited to being organized as items in a collection. For example, each book rarely had direct relations with another book. Books weren't often by the same author, and even then, knowing that two books shared an author was not too useful for the app. Instead, the more important aspect was that the books were individual elements of a "library" of books, so the book data was better suited to be a part of a list of books instead. The is furthered detailed in the database schema section (5.2.2).

5.2. System Architecture and Design

5.2.1. Architecture Overview

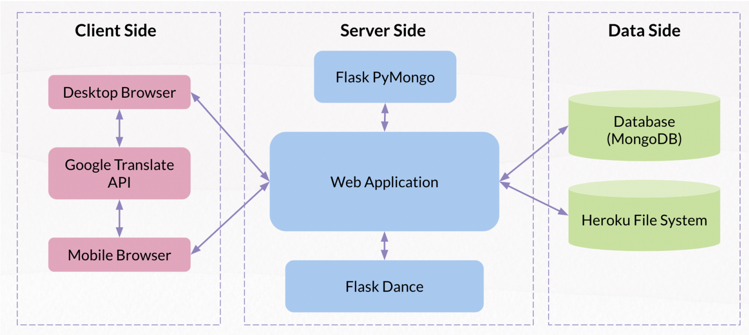


Figure 22: Diagram Overview of System Architecture

The architecture of the application can be divided into three sections: client, server, and data storage. Figure 22 shows a overview of the system architecture. On the client end, CH-ENG is a web app built primarily for the desktop browser. However, it is also mobile-friendly and compatible with the mobile browser. Developed using Bootstrap, JavaScript, Ajax, HTML, and CSS, the app also communicates with the Google Translate API on the front-end through HTTP requests. The usage of this API is for the translation feature, detailed in section 5.3.2.

The client-side uses the Jinja templating engine to complement the Flask framework on the server-side. On the server end, data (such as book information) is passed to HTML templates to be rendered. It is also in charge of handling user-form inputs and Google authentication. The latter is done with functions from the Flask Dance external library (described more in section 5.3.4). Additionally, the server-side has to fetch data from the database. CH-ENG uses MongoDB. As such, communication from the server to the database is facilitated through Flask PyMongo.

To use MongoDB in production, a global cloud database service called MongoDB Atlas is used. As the "best way to deploy, run, and scale MongoDB in the cloud," it allows MongoDB to be deployed across AWS, Google Cloud, and Azure [8]. For CH-ENG, the MongoDB database is hosted on AWS, the cloud provider recommended by Atlas. Using the connection string provided by Atlas, CH-ENG can configure the server-side to connect to the database. Once connected, the server-side can read, write, and update data to the database with Flask PyMongo.

MongoDB was chosen as the database because MongoDB is very suitable for handling JSON, the format of data in CH-ENG. In the beginning, when the JSON files didn't have to be edited, the Heroku file system was used to store all the JSON files (rather than MongoDB). However, once the information had to be updated to accommodate the dynamic recommendation feature with user feedback, an alternative was needed since Heroku doesn't handle updating file data well. I considered AWS S3 storage (a cloud file system), but objects stored are immutable. Hence, just updating a file would involve fetching the file, changing the file, deleting the original file, and uploading a new file. This would require more processing, and hence be slower, than updating data using MongoDB. Another factor in favor of MongoDB was that it had its own API for working with JSON data, making it more suitable than S3.

In addition to MongoDB, the Heroku file system is still used as a supplement to store image files. Initially, the app retrieved the images through an external URL on the client end. However, having a copy of the image stored and loaded was deemed to be more secure, as the images wouldn't depend on an external source. Hence, it is now directly stored in the file system. It is not stored in MongoDB, because images would have to be converted to bytes. Rendering images from their byte representation can be very messy as images aren't always translated to bytes perfectly, causing images to not be shown if the bytes are messed up or truncated.

5.2.2. Database Design

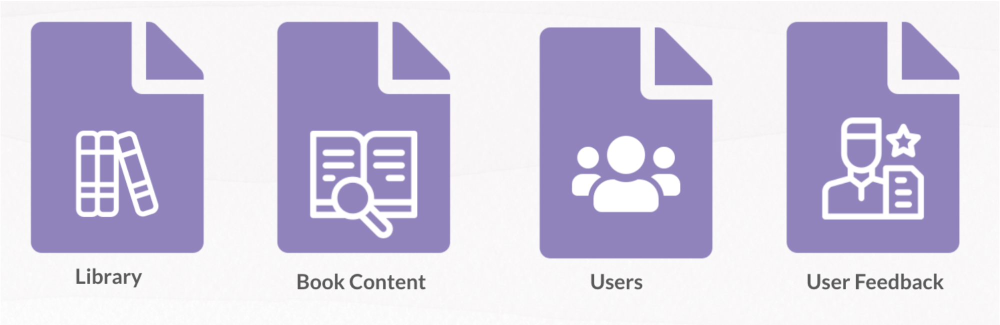
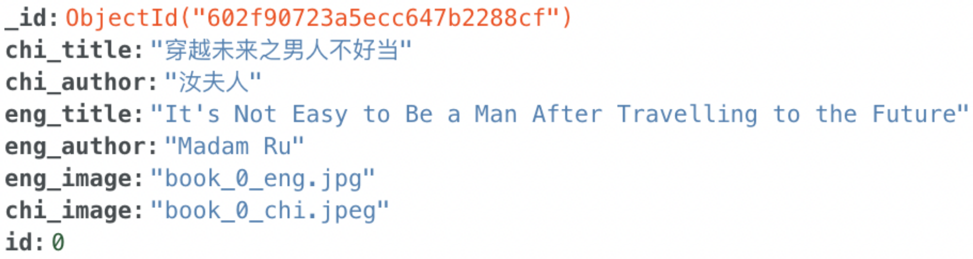


Figure 23: High-level Overview of Database Schema

As depicted in Figure 23, the database organizes the data into 4 different collections called "Library," "Book Content," "Users," and "Feedback". The "Library" collection is a list of objects that save the basic information about a book. This includes the title (in Chinese and English), author (in Chinese and English), and the filename to its Chinese and English cover image. Additionally, each book is assigned two ids. One is of type ObjectId called "_id" and this a unique identifier assigned by MongoDB to every object when added. Another is "id," an integer assigned by the app to uniquely and easily differentiate the books. This id is generated by the order the books were added. An example of an object in the "Library" collection can be seen in Figure 24.



```
_id: ObjectId("602f90723a5ecc647b2288cf")
chi_title: "穿越未来之男人不好当"
chi_author: "汝夫人"
eng_title: "It's Not Easy to Be a Man After Travelling to the Future"
eng_author: "Madam Ru"
eng_image: "book_0_eng.jpg"
chi_image: "book_0_chi.jpeg"
id: 0
```

Figure 24: Example of an element of the "Library" collection.

Another collection is for "Book Content," where the bulk of information for each book is stored. Like the "Library" collection, each element represents a book. It contains the following fields:

1. id
2. _id
3. English Description
4. Chinese Description
5. Chapter Content

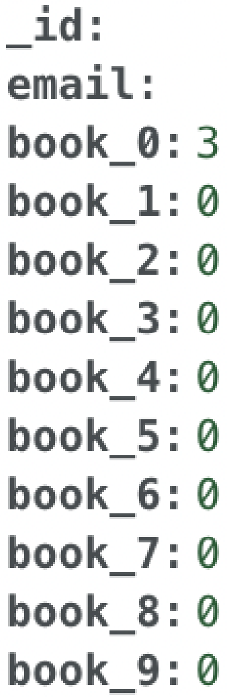
For a book, the "id" field is the same as the one in the "Library" collection. Note that the MongoDB identifier "_id," however, is not the same. Each object contains the book's English and Chinese description and the chapter content in an array. The array is a list of objects, where each represents a chapter. The chapter objects each contain the Chinese and English titles of the chapter and an array of paragraph objects. These paragraph elements each contain a Chinese paragraph, the machine-translated English paragraph (to be explained in section 5.3.1), and the English paragraph.

This arrangement is to store the Chinese paragraphs such that each is already aligned with its English counterpart. One paragraph object would represent a pair of matched English and Chinese paragraphs. The hierarchy within a chapter object can be summarized as so:

1. _id
2. Chinese Title of Chapter
3. English Title of Chapter
4. Array of Paragraphs
    (a) Chinese paragraph
    (b) Machine-Translated English paragraph
    (c) English paragraph

The reason that the "Library" and "Book Content" collections are not combined into one, even though each element is also about a book, is that the app sometimes doesn't need so much information about a book in one query. For many books, the data for the chapter contents can be quite large. Oftentimes, the app just needs a list of book titles. In such situations, the library collection would be a light-load to fetch and parse through; it would minimize the occasions where big chunks of data are passed around.

In addition to book data, the app also has to store user data. In particular, the app has to save a user's reading progress for each book; this is done in the "Users" collection. Each object represents a user, which contains the mandatory "_id" field, email, and numerical progress for each book. The email is the one they used to sign in to the app. Incidentally, this means that if a user used two emails to sign in to the app, each of the emails would be treated as a unique user. For the progress data, there is a field for each book in the object, referenced by the book id from the library collection. Because the app currently only has ten books, there is a field each for books 0 to 9. Each book field is associated with an integer that represents the last chapter the user has read; this is initialized to 0. An example of an object in the library collection can be seen in Figure 25, where the "_id" and email fields are omitted.



```
_id:
email:
book_0: 3
book_1: 0
book_2: 0
book_3: 0
book_4: 0
book_5: 0
book_6: 0
book_7: 0
book_8: 0
book_9: 0
```

Figure 25: Example of an element of the "Users" collection.

Finally, the user feedback results also have to be stored. For this, the information is arranged such that each book has its own collection. Let *x*, *y*, *z* be three books. For the feedback collection of *x*, there is a list of objects where each represents a non-*x* book. For example, one object would be about *y*. It would contain the "id" of book *y* (same as the one associated with *y* in the "Library"), the "original score" of the book (to be explained in section 5.3.3), and the number of "likes." "Likes" is the number of times users have indicated that *y* should be read after *x*. Similarly, there is also an object for *z* in the feedback collection of *x* with the same fields.

In general, the drawback to this database schema is that it does not scale well when adding new books. To add some book *x*, a new entry has to be added to the "Library" and "Book Content" collections. In addition, a new field has to be made for each user in the "Users" collection. There will also need to be a new collection for *x* to store its user feedback. Hence, to add *x*, almost every aspect of the database has to be updated. However, for this project, the expectation is that books would not be added often, since the focus is more on the features rather than the data collection. As such, scalability was not too big of a priority in the short term.

5.3. Web Application Features

This section will delve into the implementation of the main, core features of the app:

1. Paragraph-Level Alignment
2. Highlighting for Translation
3. Recommendation System
4. User Personalization

5.3.1. Paragraph-Level Alignment

One of the major challenges encountered during the development process was that the Chinese and English text did not come in a 1:1 paragraph matching. This means that the $i-th$ Chinese paragraph did not necessary match with the $i-th$ English paragraph in a given chapter of a book. This is because the translators did not necessarily preserve the paragraph format. Some decided to combine multiple Chinese paragraphs into one English paragraph or split one Chinese paragraph into multiple English paragraphs. As such, an algorithm had to be made to actually align the texts at the paragraph level. This algorithm had to be run once for each chapter of a book. That is because the chapters are aligned even though the paragraphs are not. A translator would not make up new chapters or combine chapters into one. So, the paragraph alignment could be done in batches by chapter and any error from one chapter would not impact the next.

Before running the algorithm, the entire Chinese text had to be converted into a machine-translated (MT) English version. This was done through DocTranslator [5], which can take a Chinese text file and output the MT English file such that the MT English is still paragraph aligned with the original text. The algorithm then takes, as input, the MT English and human-translated (HT) English texts for a chapter. Let the MT paragraphs be numbered as $m_0...m_n$ and the HT paragraphs as $e_0...e_q$, where n is the number of MT paragraphs and q is the number of HT paragraphs. Note that n and q are not necessarily equal. The algorithm first iterates through each MT paragraph. For each MT paragraph, the algorithm matches it with the best-fit HT paragraph. It does so by looking at the HT paragraphs within an index range of 3 above and below the last matched HT paragraph. For example, suppose it is trying to match MT paragraph m_i. Let l be the index of the last HT paragraph (initialized to 0 if no HT paragraph has been matched yet).Then, for m_i, it considers each HT paragraph in range $[e_{l-3}, e_{l+3}]$ to be potential match candidates. The range of 3 was chosen after experimentation. A bigger range would lead to more chances for mis-matches, while a narrow range would not adequately account for the extent the actual HT match can be displaced from e_l.

Each candidate e_p (where $l-3 \leq p \leq l+3$) is given a score, and the candidate with the highest score is matched with m_i. The score considers the following factors multiplied together:

1. percentage of common words shared between m_i and e_p
2. word count difference between m_i and e_p
3. index difference between i and p

In general, the higher the percentage of common words, the more likely e_p is a good match. Let x be the number of common words between m_i and e_p. Let y, z be the number of words in m_i and e_p respectively. Then, the percentage of common words is calculated by $\frac{\frac{x}{y}+\frac{x}{z}}{2}$. This is the average percent of common words when taking both y and z as denominators. For word count differences, the lower the difference, the more likely e_p is a good match. Taking this into consideration, $\frac{min(y,z)}{max(y,z)}$ is used to account for the word count difference factor. Similarly, for index differences, this is factored into the score as $\frac{1}{|p-i|+1}$. This is because the lower the difference, the better it is for e_p. So, taking the inverse of $|p-i|+1$ would reflect that. Incidentally, the $+1$ is to ensure that the denominator is not 0.

Combining all the factors together, the score for e_p is calculated by $\frac{\frac{x}{y}+\frac{x}{z}}{2} * \frac{min(y,z)}{max(y,z)} * \frac{1}{|p-i|+1}$. Once the best-fit HT paragraph is chosen, l is then updated with that HT paragraph's index, which will be used to match the next MT paragraph. Note that there are actually many different ways to account for the individual factors in the score. For example, I could have scored the word count difference factor as $\frac{1}{|y-z|+1}$ or put some constants to weight certain factors more than others. The formulas I used above are just those that I settled with after some informal experimentation and trial-and-error.

While the scoring system would try to find the best-fit English paragraph for a Chinese paragraph, it would not proactively address the issue of when multiple paragraphs are actually linked to one. For example, if the content of t Chinese paragraphs is condensed into one English paragraph, the one English paragraph would be repeated t times. This makes it very repetitive for the reader. If one Chinese paragraph contains the content for multiple English paragraphs, then only 1 English paragraph would actually be linked to the Chinese paragraph; the others would be ignored. This would effectively cause missing content on the English side.

To proactively address these issues, the algorithm first detects for the "multiple English to Chinese" case by checking if there are skips in the matched English index. For example, if the previous Chinese paragraph t was matched with the English paragraph at index i and the next Chinese paragraph was matched with $i+2$, paragraphs i and $i+1$ are both probably for t. In that case, i and $i+1$ are combined into one giant paragraph, and that will be matched with t. To detect for the "multiple Chinese to English" case, the algorithm checks if multiple successive Chinese paragraphs are matched to the same English paragraph. An example is if Chinese paragraphs at index i and $i+1$ are both matched with the same English paragraph t. Then, i and $i+1$ are joined and t will be matched with the combined paragraph.

5.3.2. Highlighting for Translation

This feature uses the Google Translation API under Google Cloud as the translation service. Google's API was chosen because it is a very popular translation service that has great support for the Chinese language and allows for "fast, dynamic results" [6]. In particular, getting fast translations for this feature is important because the translation should keep up with whatever the user is highlighting in real-time. Users can change what they highlight very easily and quickly, and the translation has to change just as quickly too. Additionally, the main goal of this feature is convenience. While the highlighting aspect only requires one step for users to get a translation, making users wait for the translation would break their reading process and therefore make it less convenient.

To ensure quick translations, CH-ENG communicates with the API directly on the client end through HTTP post requests. First, the app uses JavaScript to detect if the language in the highlighted text is either Chinese or English. This is done with RegEx for the English alphabet and matching with the Unicode for Chinese characters. As a restriction, the translation will not be given if the highlighted text has both languages (ex: "Information 信息"). This is to limit users from misusing this feature when highlighting across multiple paragraphs. Checking for both languages is a good indicator that the user is highlighting multiple paragraphs because the highlighting will always go horizontally from the English paragraph to the Chinese paragraph, rather than vertically downwards. This means that the user cannot highlight two consecutive English paragraphs a, b without highlighting the Chinese paragraph matched with a. This limitation stems from how the paragraphs alternate in language when the interface displays paragraphs on a narrow screen (Figure 17). This was not considered to be a major issue, as the feature was not meant for a big block of text, and the translation of a whole paragraph is readily available in the other column.

After detecting the language in the highlighted text, the app forms the request to Google by specifying the text, the target language (the other language not in the highlighted text), and the app's API key. This API key is set up through Google Cloud's system, where they can associate the request with this app. It is only after the API key is set up that the app can fetch translations from Google. After the request is sent, the result is then put into a tooltip.

For this feature, accurately positioning the tooltip was tricky. In addition to finding where the highlighted text is in the window, it had to account for users resizing the window while the tooltip was still present. Resizing the browser window meant that the tooltip had to dynamically change its position as well. The positioning was all done in JavaScript, and this issue mostly involved fiddling with numbers and calculating coordinates. Currently, the positioning can still be a little off when the user resizes the window. However, this can easily be fixed if the user highlights again. As mentioned in section 4.1.6, there is a disclaimer on the page about this issue in the information pop-up.

To supplement the tooltip, the app attempts to find the highlighted phrase in the corresponding paragraph. The difficulty in this feature was in trying to know exactly how something will be phrased in the corresponding paragraph, with only knowing the meaning of what was highlighted.

This is due to synonyms and the numerous different ways of conveying something. Take English, for example. "in her dream" can be "inside her dream" or use some synonym like "in her sleep" to convey the same or similar meaning. So, it was hard to predict how exactly something will be phrased in the corresponding paragraph, especially in another language. In consequence, this made it difficult to locate where, for example, "in her dream" is in the Chinese paragraph.

The work around this app did was to use the Google translation as a frame of reference. This was referred to as highlighting-on-both-sides in section 4.1.6. Let x be the phrase that is highlighted, where x can either be Chinese or English. Let x' be the output translation of x by Google. If x' is in Chinese, then for every character in x', the app will search for it in the corresponding Chinese paragraph and highlight it in pink. If x' is in English, then for every word in x', the app will search for all instances of the word in the corresponding English paragraph and highlight it. With this approach, there is the limitation that what Google outputs may not necessarily be in the corresponding paragraph. For example, it may look like scattered blobs or perhaps nothing is highlighted at all. However, the scattered blobs (like in Figure 18) would at least help visually pinpoint the user to a general direction whenever an area has more pink clusters. As such, while this feature may be limited, it is an attempt at tackling the problem and can provide some degree of helpful information.

5.3.3. Recommendation System

One of the project's goals was to develop a recommended reading sequence based on reading difficulty for users. Here, "reading difficulty" is about readability (extensive reading) rather than learnability (intensive reading). The focus is simply if the users can read and understand the text, not focusing on if they can apply what they learn from the text. The approach to this was to algorithmically suggest the top three books to read after some book x. So, after users read book x, they could look at the suggestions to see what books are of similar or slightly-higher reading difficulty. The challenge to this feature was in figuring out how to actually determine the difficulty of a book. The approach to this was to, for each book y (that is not x), give a score that would represent the relative reading difficulty to x. The lower the score, the lower the difficulty. The app would, for the "suggested next reading" of book x, recommend the lowest 3 books. The score is calculated by the product of several factors: average character frequency rank of y * average sentence length of y * number of new characters in y (relative to x) * $\frac{1}{\text{number of users who suggested } y \text{ after } x + 1}$. Each of the factors are meant to encapsulate one of the following:

1. Relational Lexical Difficulty: New Characters
2. Lexical Difficulty: Character Complexity
3. Syntactic Difficulty: Sentence Length
4. User Feedback

Relational Lexical Difficulty: New Characters

To capture the reading difficulty of some book y relative to x, I used the number of "new characters" in y. That is, using the total character set in x as the baseline, the "new characters" in y are the characters that are part of y, but not in x. Note that if a character c in y doesn't appear in x, then every appearance of c in y is counted as one new appearance for simplicity. This would put more weight on "new characters" that appear more often, meaning that users would actually have to remember these characters more. The underlying assumption behind this metric is that the more new characters y has compared to x, the more difficult it will be for the user to read y next because there are more characters that the user potentially doesn't know.

Let $NNC(x,y)$ refer to the number of new characters in y relative to x. NNC provides a directional relationship, where $NNC(y,x) \neq NNC(x,y)$. To explore this metric, I calculated the NNC of every pair of books in both directions. Figure 26 is a scatter plot of the character difference (NNC) results. All dots for some x value represents the NNC of each book (not x) relative to x. The graph shows that the range of NNC for each book varies, with some clustered in a small range and some scattered over a larger range. This indicates some degree of diversity in NNC scores. However, upon examining pairs of books with higher NNC, I observed that the chapter difference of the pair of books potentially impacts the NNC. To determine the correlation, I calculated the chapter difference of every pair and compared it to the NNC. To determine the chapter difference, I subtracted the number of chapters in x from y if $NNC(y,x)$ and the number of chapters in y from x if $NNC(x,y)$.

This means that the chapter difference is also a directional relationship. If x has more chapters than y, then comparing y relative to x would result in a negative difference. Otherwise, comparing x relative to y would result in a positive difference. Figure 27 shows a graph of the chapter difference versus the character difference. The red line marks a positive trend line, suggesting that the greater the chapter difference, the greater the character difference. This indicates that the chapter difference may be a confounding factor in the calculation of the NNC.

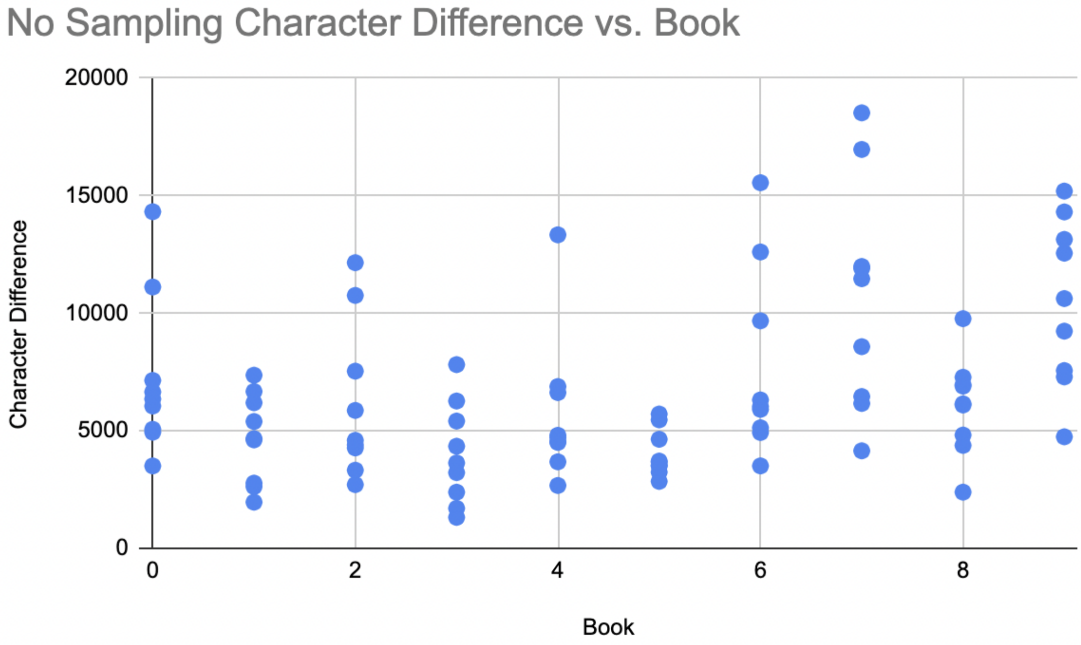


Figure 26: Scatter graph showing the possible character differences for each book with no random sampling.

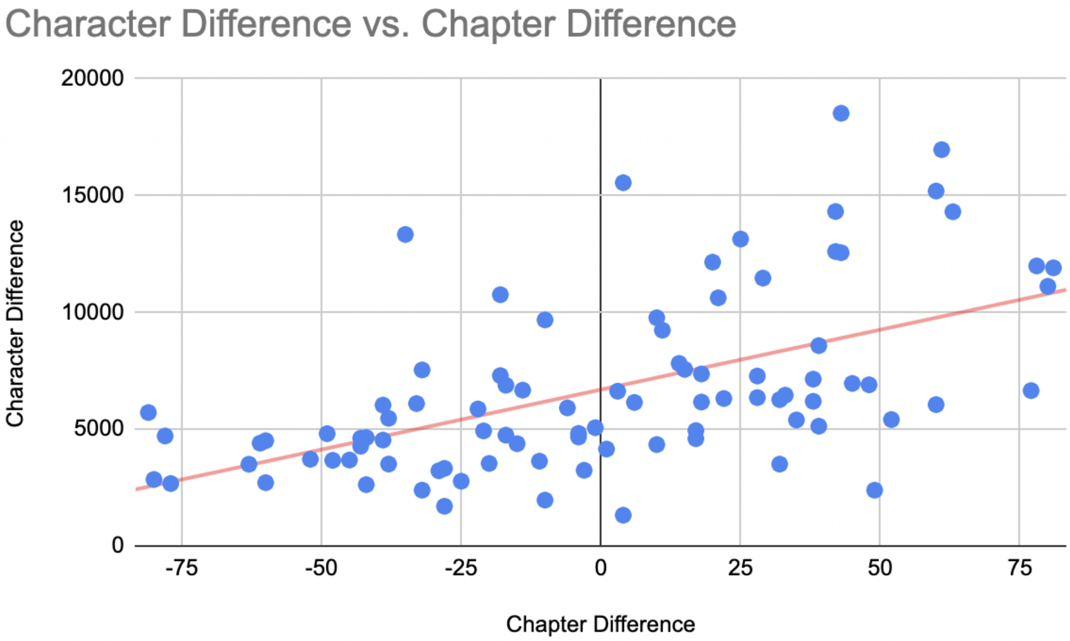


Figure 27: Graph of the correlation between chapter difference and character difference.

To account for the chapter difference, I experimented with taking a fixed random sample of chapters from each book (as opposed to using all the chapters) when calculating the *NNC*. I set the sample size to be the minimum total chapters out of all the books. In the app's sample of 10 books, the sample size came out to be 51 chapters. Using this number, I recalculated the NNC of every pair of books in both directions, taking a random sample of 51 chapters from each book. The random sampling had an impact on the range of values for the NNC. Figure 28 is a scatter plot of the character difference (NNC) results with sampling. Compared to figure 26, the books still have diverse ranges of NNC scores. However, upon examining the y-axis, the range of values have actually proportionally shrunk, where 60% of the books had a change in the minimum possible NNC value.

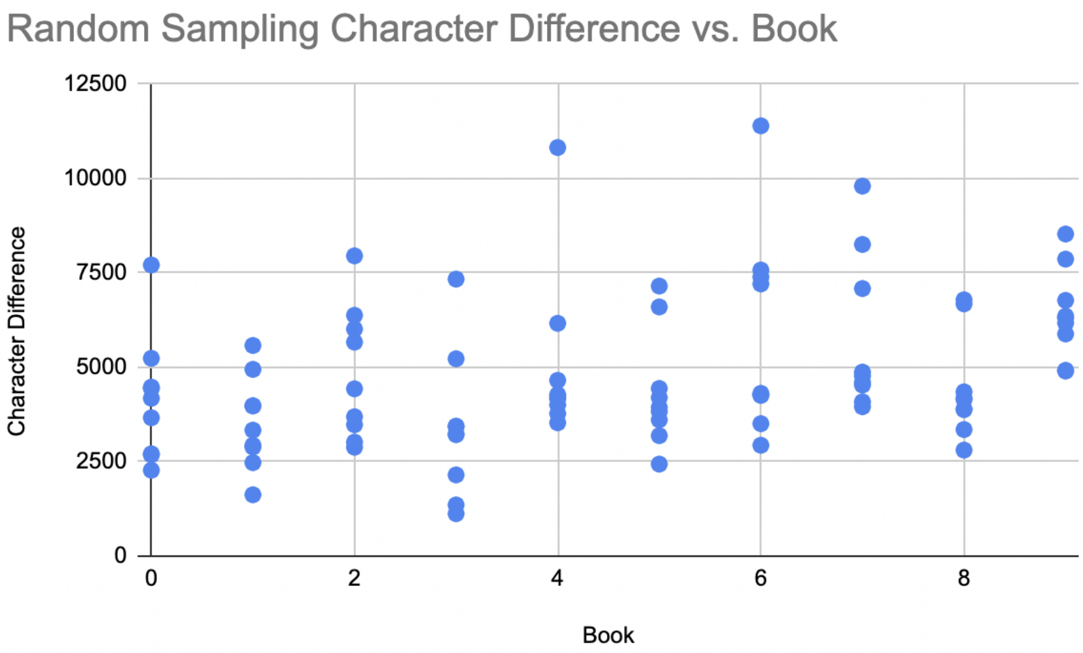


Figure 28: Scatter graph showing the possible character differences for each book with random sampling.

To observe if the sampling helped account for the noise from chapter differences, I calculated the difference in NNC by taking the absolute value of the difference between the first NNC value (without sampling) and the second NNC value (with sampling). This will be referred to as the "difference of differences." The absolute value was taken because the focus was on the degree of change on the NNC value, rather than the positive/negative direction of change. I then graphed this against the total chapter difference, which can be seen in Figure 29. From the graph, the line of best fit indicates a positive correlation, meaning that the greater the chapter difference, the more random sampling changed the NNC value. This suggests that the random sampling helped "correct" the impact of chapter difference on the NNC. As such, NNC with random sampling was preferred over NNC with no sampling for the scoring metric of relational lexical difficulty.

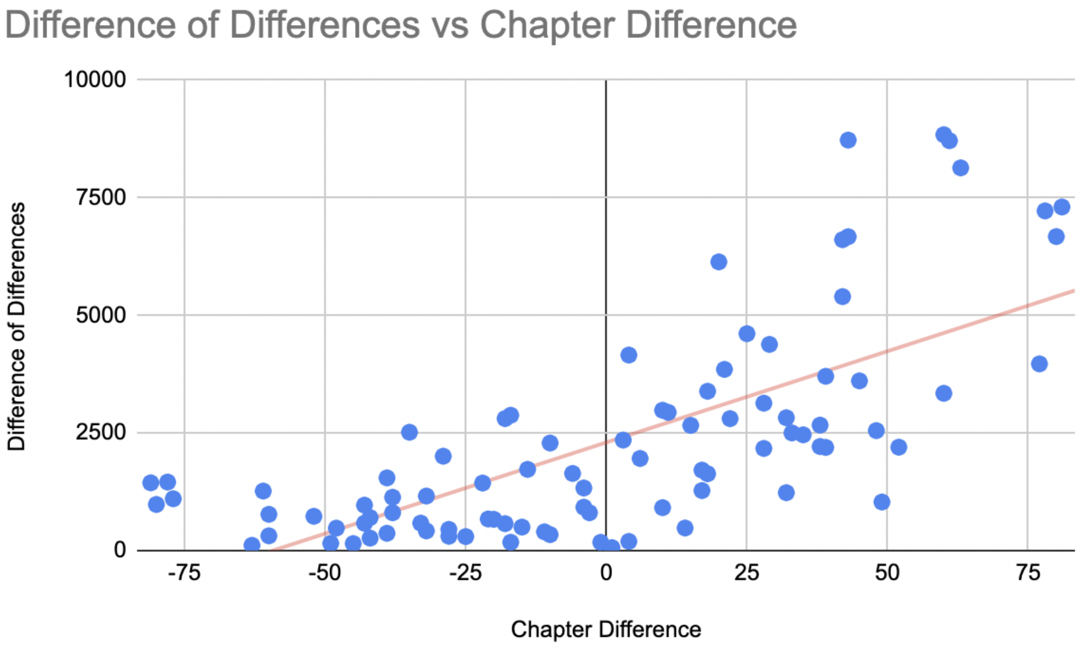
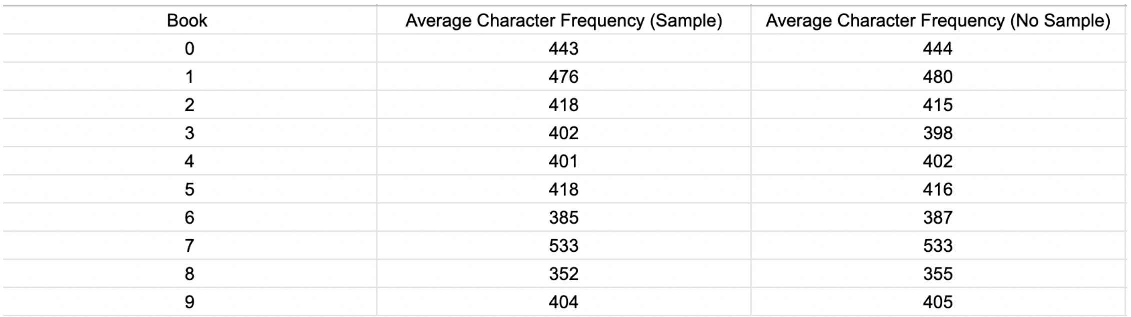


Figure 29: Graph of the correlation between chapter difference and difference of differences.

Lexical Difficulty: Character Complexity

According to a linguistics study on assessing Chinese readability, word frequency can measure word difficulty, and the frequency of the Chinese characters in a text has a positive impact on readability [12]. From this, I explored ways to incorporate frequency into the reading difficulty score. For this project, "frequency" is defined as how common a character is used or seen. Luckily, there is an existing dataset with such information from hanziDB, where they have a list of characters ordered by frequency rank [9]. The higher the rank, the more common the word. For example, the rank 2 character is more common than the rank 5 character. Using this data, I decided to use the metric of average frequency rank as the scoring metric. This is calculated by adding the frequency rank of all characters in the novel and dividing that by the number of characters. For this, random sampling was not used as there is no significant difference in the averages with and without sampling, as shown in Figure 30.



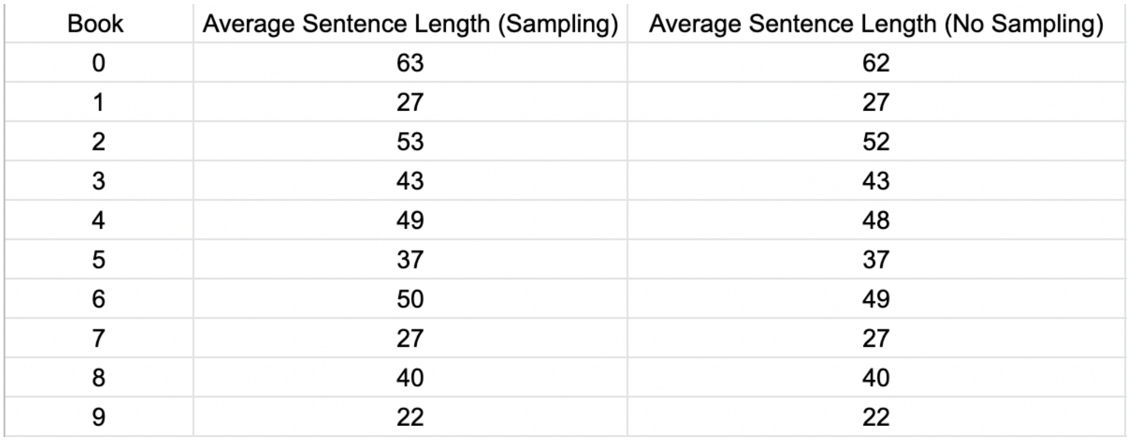
Book	Average Character Frequency (Sample)	Average Character Frequency (No Sample)
0	443	444
1	476	480
2	418	415
3	402	398
4	401	402
5	418	416
6	385	387
7	533	533
8	352	355
9	404	405

Figure 30: Average character frequency rank with and without sampling for each book.

Another metric that was explored was stroke count. According to a study about the effect of pattern complexity, the number of strokes is a common measure of character complexity. Crowding, which is the space between adjacent characters, increases with complexity; this limits visual span. They define visual span as "the number of letters that can be recognized without moving the eyes" [20]. While the same hanziDB data set from above does have data on the number of strokes per character, this metric was ultimately not used as stroke count is more applicable to a short text excerpt, rather than a whole novel. Additionally, limited visual span does not necessarily inhibit the reader from ever understanding the text, and the app focuses on if the user can understand the text even if they read it slowly.

Syntactic Difficulty: Sentence Length

In a study about CRIE, a tool that uses machine learning to analyze Chinese texts, the paper mentions that there is a positive correlation between sentence length and syntactic difficulty: "Complex sentences are usually longer, structurally intense, and impose a higher cognitive burden on the reader" [18]. This led to the idea of average sentence length, which is included in the app's scoring system. This metric is calculated by summing the length of all sentences in a novel and dividing that by the total number of sentences. Like average character frequency, sampling does not have much impact on average sentence length, as shown in Figure 31. Hence, random sampling is not used.



Book	Average Sentence Length (Sampling)	Average Sentence Length (No Sampling)
0	63	62
1	27	27
2	53	52
3	43	43
4	49	48
5	37	37
6	50	49
7	27	27
8	40	40
9	22	22

Figure 31: Average sentence length with and without sampling for each book.

User Feedback

In addition to linguistic factors, the score also accounts for user feedback with $\frac{1}{\text{number of users who suggested } y \text{ after } x+1}$. Users can recommend books to be read after book x and this would dynamically update the score of those recommended to account for the suggestion. The inverse is used, because the score is "better" the lower it is. User feedback is included because if the app gives a poor recommendation, there is a chance for it to be "corrected" through user opinion. User opinion is also very valuable because it is ultimately other similar users who will read the books and use the app as well.

Implementation of Algorithm

Besides the user feedback factor, the other aspects of the score will never change. Hence, to save on repeated calculations, the following makes up the "original score": average character frequency rank of y * average sentence length of y * number of new characters in y (relative to x). This is pre-computed and stored in the database under the feedback collections, as mentioned in section 5.2.2. Then, whenever the user loads the page for book x, the app will calculate the scores of all books y that are not x. This is done by multiplying the original score with $\frac{1}{\text{number of users who suggested } y \text{ after } x+1}$. Then the app ranks the scores, outputting the three books with the lowest scores.

As described in section 4.1.5, the user feedback is done through filling the form on the "Recommendations" page. They essentially use the form to complete the following sentence: "By reading difficulty, [Book 1] should be read [before | after][Book(s) 2]." Suppose the user inputs: "Book *a* should be read after books *b* and *c*." Then, upon submission, the app will fetch the feedback collections for both *b* and *c*. Finding the entry for *a* in both collections, the app will update the "likes" of *a* by 1 in both. Suppose the user inputs this instead: "Book *a* should be read before books *b* and *c*". Then, the app will fetch the feedback collections for *a*. For both the *b* and *c* entries in the collection, the app will update the "likes" by 1. With this database structure, the app will only need to fetch the feedback collection for *x* to calculate the scores when the user loads the Book page for *x*. Overall, this implementation helps save the processing on data that can be pre-computed, while automatically maintaining the dynamic aspect of the scoring system from the user feedback.

5.3.4. User Personalization

Currently, users can sign in with their Google accounts to save their reading progress. While not about language learning, this feature is designed for CH-ENG to give users a better reading experience as a novel-reading application. Google sign-in was chosen for account authentication because there was the underlying assumption that many users probability have Google accounts. For CH-ENG, the app only needs the email of the user to identify them. Whenever a new user logs in, the app checks if the user is already in the database. If they aren't, then the app will add a record into the "Users" collection. Once the user has logged in, the app will automatically save the user's progress by storing, per book, the number of the last chapter they read. Whenever the user navigates to a different chapter for a book, the app will update the user's position in the database. This means that if the user jumps from the first to last chapter for a book, then the app will indicate the last read chapter as the last chapter. Likewise, if the user goes back from the last chapter to chapter one, the app will say that chapter one is the last read chapter. As mentioned in section 4.3, the last saved point can be seen on the Books page. When the logged-in user loads the page, the app will automatically check if the user has already read the book. If so, it will fetch the chapter they were last reading and provide a "Continue" button. If not, then the app will show a "Read Chapter 1" button.

To enable Google authentication, I first had to register the app with Google to obtain a client id and secret code. Then, on the server end, the app uses the Flask Dance API to connect with Google's

OAuth system. Flask Dance is a general library that allows a Flask app to connect with OAuth for many providers such as Google, Github, and Facebook [11]. In CH-ENG, Flask Dance helps create a "blueprint" for the Flask session to request and obtain authorization from Google when a user clicks "Login." The blueprint contains the app's client id, secret code, scope, and redirecting path. The scope refers to the type of information being requested from the user logging in, such as their name. The redirecting path is the page the user should be redirected to after they log in. Once Google gives authorization, the app can use functions in the Flask Dance library to obtain allowed information about the signed-in account, such as the email address. Once the user logs out, their session is cleared, and the app will no longer show any reading progress data.

6. Evaluation

For this project, there were two types of evaluation. The first was to evaluate the accuracy of the algorithms, namely the paragraph-alignment and book recommendation algorithms. The second dealt with getting user feedback on the app itself.

6.0.1. Evaluating the Paragraph-Level Alignment Algorithm

To evaluate this algorithm, I took a random sample of five chapters. I ensured that they were each from a different book (chosen randomly). This was to account for the fact that an algorithm may be performing well for certain books over others. Then, for each selected chapter, I evaluated if each Chinese paragraph was matched with the correct English paragraph. This was manually done by looking at the English machine-translated (MT) version of the Chinese paragraph and comparing the MT version to the matched English paragraph. Using my best judgment, I looked at the words and general meaning expressed in both paragraphs to determine if they were a good match. Note that "paragraphs" in web novels are typically very short, sometimes even just one sentence. This made it much easier to compare the paragraphs. In addition, typically both paragraphs would either be very similar or very different– no in-between. An example of a good match is the MT paragraph:

> "With red eyes, Lan Luofeng said firmly: 'Don't worry, Ling Xiao, I will take care of our

> home.’ She put Ling Xiao’s hand on her abdomen, and said with shame: ’In more than eight months, you will I’m going to be a father.’"

and the corresponding English paragraph:

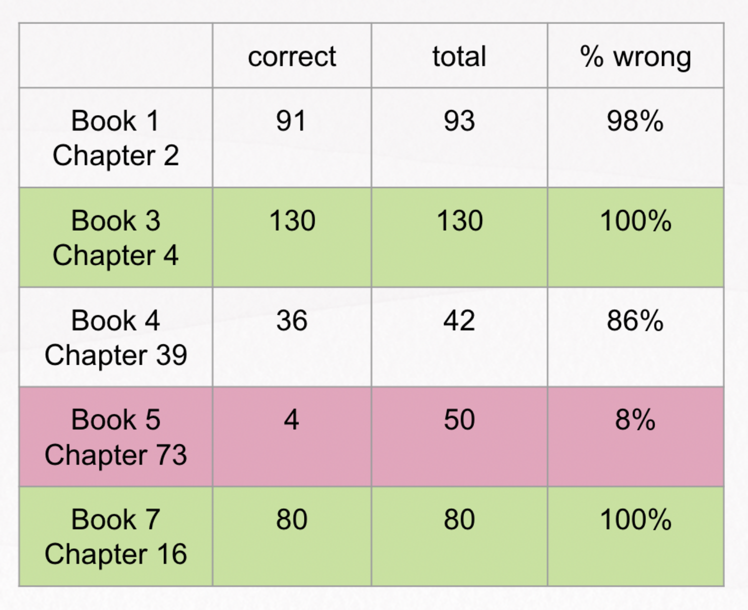
> "With red-rimmed eyes, but a firm voice, Lan Luofeng said, ’Don’t worry, Ling Xiao. I will take good care of our household.’ She placed Ling Xiao’s hand on her abdomen, and said shyly, ’In another eight months or so, you are going to be a father’[17]."

An example of a bad match would be the MT paragraph:

> "This man is not easy to provoke!"

and the corresponding English paragraph:

> "’Chu Liuyue.’ She had no intention of hiding her identity. After all, he looked like a man of many means. Lying would only cause more problems [15]."



	correct	total	% wrong
Book 1 Chapter 2	91	93	98%
Book 3 Chapter 4	130	130	100%
Book 4 Chapter 39	36	42	86%
Book 5 Chapter 73	4	50	8%
Book 7 Chapter 16	80	80	100%

Figure 32: Evaluation Results for Paragraph-Alignment Algorithm

For each of the five sample chapters, I calculated the percent accuracy by dividing the number of correct alignments by the total number of paragraphs. Overall, the average accuracy per chapter is 78%. The result for each chapter is shown in Figure 32. There are two chapters with 100% accuracy, highlighted in green in the figure. Two other chapters, while not perfect, also have a relatively high accuracy of 98% and 86%. In these chapters, the incorrect matches are in the middle of the chapter, and because the impact of the inaccuracy was small, the algorithm was able to self-correct the displacement. This means that although there are inaccurate matches in the middle, the chapter still ends with correctly matched paragraphs.

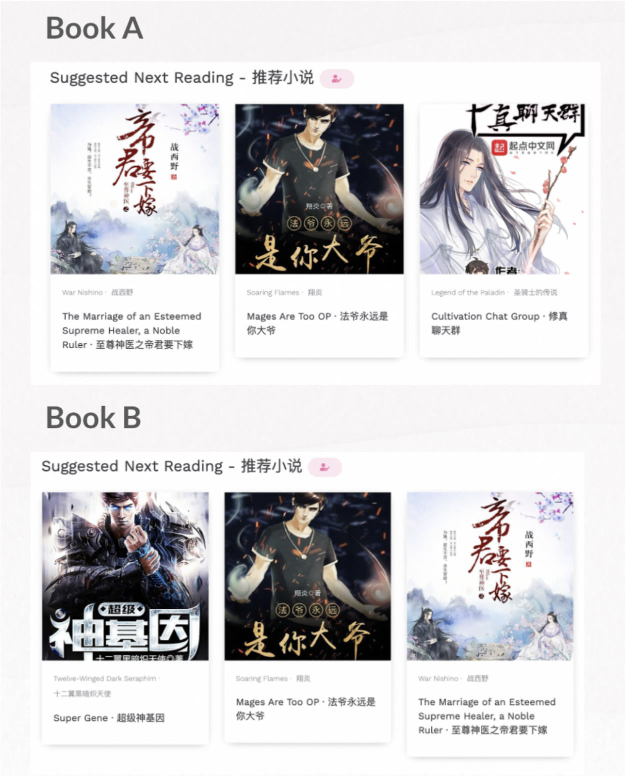
However, one chapter has only an 8% accuracy. Upon inspection, this is because, at the beginning of the chapter, there are a couple of instances where the algorithm isn't able to match an MT paragraph to the correct English paragraph due to synonyms and different phrasing. For example, the MT paragraph "The monks are also very busy, except for the scattered people in Beihe" should be matched with "Cultivators were quite busy... other than Northern River's Loose Cultivator" [16]. However, the algorithm isn't able to connect them because it isn't able to associate "Northern River's Loose Cultivator" with "Beihe" and "Cultivators" with "monks." Instead, it matched the MT paragraph with "Inside the dorm, Song Shuhang went to bed very early" due to the score settings. The mismatched English paragraph caused a relatively large displacement that was outside the tolerance range of three (set from the algorithm). Due to other such instances early in the chapter, the algorithm ultimately isn't able to correct itself since the beginning, and this error is carried until the end of the chapter. As such, the four correct paragraphs are the very first four paragraphs of the chapter, before any displacement started.

Overall, from the evaluation results, the algorithm often performs well. It was able to perfectly match two of the five chapters, proving that it does have the potential to be 100% accurate. Additionally, in the two chapters that aren't perfect but have high accuracy, the algorithm shows that it can self-correct small displacements. The biggest drawback is that the algorithm can perform very poorly for some occasional chapters. However, the algorithm has much potential to improve. By experimenting with different ranges and ways to calculate the score, the algorithm can potentially increase its fault tolerance to account for major errors (like the one from the evaluation).

6.0.2. Evaluating the Book Recommendation System

Evaluating this algorithm was tricky because there was difficulty in finding "truth" to these recommendations without numerous user feedback. However, getting user feedback was difficult due to the nature of what was being evaluated. For example, let book s be recommended as a follow-up to book p. It wouldn't be known if s is a good recommendation until multiple users, who have read all of p, read s and give feedback about this recommendation. For this project, that was a bit too much to ask of users.

An alternative evaluation method was to observe the books being recommended. The observed trend was that different books have different suggestions, but due to the small sample size, many share the same suggestions. For example, Figure 33 shows the app's suggestions for two books. Note that the suggestions are shown such that the left-most of the three is the top suggestion and the right-most is the top third. These two books share two of the same suggestions, although the rank is not necessarily the same. The top suggestion for book A is the top third for book B, and the top second for both books is the same. This implies that the scores are actually varying per book, but certain books are nevertheless still consistently scoring well in the algorithm.



Book A

Suggested Next Reading - 推荐小说

War Nishino · 战西野

The Marriage of an Esteemed Supreme Healer, a Noble Ruler · 至尊神医之帝君要下嫁

Soaring Flames · 翔炎

Mages Are Too OP · 法爷永远是你大爷

Legend of the Paladin · 圣骑士的传说

Cultivation Chat Group · 修真聊天群

Book B

Suggested Next Reading - 推荐小说

Twelve-Winged Dark Seraphim · 十二翼黑暗炽天使

Super Gene · 超级神基因

Soaring Flames · 翔炎

Mages Are Too OP · 法爷永远是你大爷

War Nishino · 战西野

The Marriage of an Esteemed Supreme Healer, a Noble Ruler · 至尊神医之帝君要下嫁

Figure 33: Comparison of the suggested books for two books.

The reason for the similar recommendations may be that two of the four factors in the scoring system are not relationship dependent. In particular, these factors are the average character frequency rank and average character count per sentence. No matter what book another book is compared to, these numbers are finalized and never changed. So, its score mostly fluctuates from book to book by the new character and user feedback factor. As such, a possible way to improve the algorithm is to make these factors relationship dependent. For example, suppose the algorithm is finding suggestions for book p. Let the average character frequency rank of a book p be $CFR(p)$. When calculating the score for some book s (that is not p), the algorithm could subtract the CFR of s from p ($CFR(p) - CFR(s)$). Recall that a higher CFR means that the books have more uncommon words. Then, a positive difference would mean that p has a higher CFR, perhaps indicating that p is harder than s. Likewise, a negative difference may mean that s is harder than p. Factoring some modification of $CFR(p) - CFR(s)$ into the score, rather than just $CFR(s)$, may make the scores more unique, and therefore vary the recommended results more. Similarly, another possibility could be to explore what the difference in average character count per sentence can reveal, and this can also help make the suggestions more unique.

Other reasons for the similar recommendations may be due to the small sample size of books and the low number of user recommendations. Currently, the app does not have a sufficient user base that actively provides recommendations. As such, the "dynamic" aspect of the scoring system is not yet fully utilized. With more books and more active users, the algorithm could output more varied results.

6.0.3. User Evaluations

For this project, there were two rounds of evaluations. The first round tested the MVP (minimal viable product). At the MVP stage, the app did not yet have Google sign-in, book recommendations, highlighting-on-both-sides, and the sidebar when reading. The second round tested the current version of the app without highlighting-on-both-sides; this feature was added after evaluations were done. For both evaluations, there were six users; the same users were used each time. They varied in Chinese reading proficiency. Some users have never learned Chinese, while others were perfectly fluent in reading. They also varied in personal engagement with web novels, with some only knowing about them while others are avid readers themselves. For each evaluation, users were given a document containing a task list and a brief introduction paragraph with instructions. The task list for round 1 and 2 can be found in Appendix Figures A1 and A2 respectively. Users were asked to talk out loud as they walked through the tasks. Additionally, they were encouraged to try to figure out how to do the tasks themselves. While they did the tasks, I took notes on any tasks they struggled with and any feedback they said.

In the first round of evaluations, users were largely able to get through the tasks themselves without much help. There were no issues with page navigation, and completing the tasks was fairly intuitive. Even if they stumbled, they were able to easily adapt. For example, one user initially wasn't sure how to go back to the Books page from the Reading page. However, after exploring the page and finding the link, they said that it became intuitive. Additionally, the highlighting-for-translation feature garnered much appeal from the users. Many spent a relatively long time on the task to play around and see what translations came up. There were also many compliments about the aesthetics of the app, such as the formatting of the "cards" for the books on the Library page.

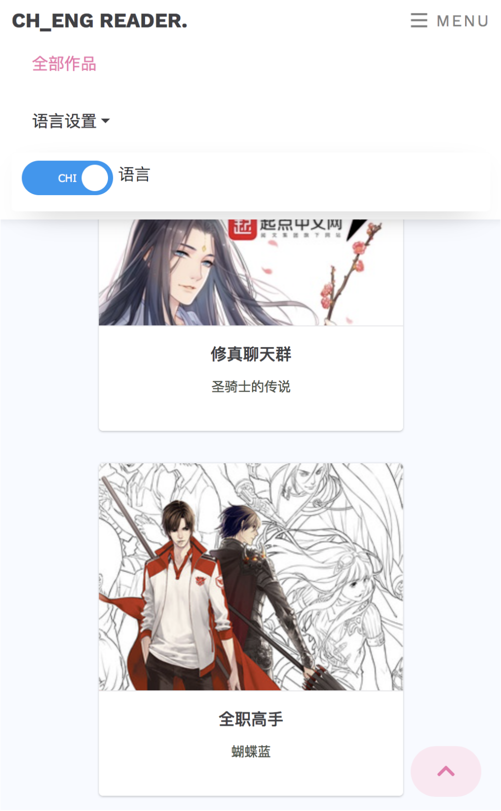


Figure 34: Imagine of the language toggle from an earlier version of the app.

From this round, the primary source of confusion was the site-wide language toggle the app had at the time (mentioned in section 4.3). This toggle would change a text to Chinese or English (if the counterpart wasn't already on the page), and the ordering of texts would change. For example, in the English version, the English paragraphs would be on the left (on the Reading page), and in the Chinese version, the Chinese paragraphs would be on the left. Figure 34 shows how the toggle looked in the app in the earlier version. One issue was the color scheme of grey for English and blue for Chinese. A user commented: "The color choice is very unintuitive. The grey, to me, would indicate disabling." On the pages, there were also inconsistencies with what was translated and what was not. For example, one user was confused why the title of the app and the footer text did not change to Chinese when the toggle was set to Chinese. They were unsure about the extent of the language parallelism in the app. Users were also confused about why the texts were getting flipped: "Don't flip things with the toggle– there's no point. The format should stay the same." As such, multiple users have expressed that the toggle should just be removed and to show the two languages on the page together.

Other feedback was on some user interface or wording aspects. For example, on the Reading page, a user suggested that the size of the text should indicate what's important. In the early iteration, the title of the book was in large prominent letters, while the title of the chapter was in medium-size letters under the book title. They suggested that the chapter title should be given more emphasis; the book title should be smaller and be put in a corner instead. Another example is the user suggestion for sign-post indicators on the home page. In the first version, the Home page contained the contents of the current Home page, the Features page, and a section explaining the book collection. Because the scroll bar was not apparent, having some form of "next" would ensure that users knew there was more information to scroll down to. There were also many user ideas for app additions, from small components such as adding GIFs on the Home page to feature suggestions like a sidebar for chapter navigation (which was later implemented). Overall, the first round of evaluations indicated that the app was heading in the right direction. It was relatively functional and was largely not difficult to use. However, user experience with component sizing or placements could use some work, and the approach to language symmetry in the app had to be reconsidered. Aggregate field notes from the first round of user evaluations can be found in Appendix Figure A3.

For the second round of evaluations, there was no one big issue where multiple users have expressed feedback about (like with the language toggle). However, a user did find a bug where the tooltip is triggered when it's not supposed to be – that has since been fixed. There were also further suggestions on how to improve the user experience. One was to have more proactive error-handling with the recommendation form. In the past, when users selected the same book for multiple fields, there would be an "invalid" warning when the user clicked "Submit." Instead, the suggestion was to not let that possibly happen in the first place by removing options depending on what the user has previously selected. Another suggestion was to have the "card" formats be more consistent. One type of card had a shading animation with non-clickable pictures, while another one had a clickable image with no shading animation. Additionally, a suggestion was to divide the Home page into separate pages. At this point, the Home page still had the contents of the Features page, so it was only after the second evaluation did the features information finally get its own separate page.

Nevertheless, there was much positive feedback in the second round. Multiple users have expressed that they really like the user personalization with Google sign-in, and users still liked to play with the highlighting-for-translation feature. There was also more praise about the user interface and user experience compared to the first round. Aggregate field notes from the second round of user evaluations can be found in Appendix Figure A4. Overall, the app was evaluated to be intuitive, pretty, and works as intended. Some direct quotes are:

- "It very pretty and works pretty seamlessly. JS is responsive for the most part."
- “Pretty and surprising intuitive.”
- "UI is 11/10."
- "Everything is as expected."

7. Conclusion

7.1. Reflection: Lessons Learned

7.1.1. What went well?

Overall, this project was able to result in a usable, intuitive application that provides a proof of concept for the possibility of language-learning with web novels. It met the goals set in the beginning– the paragraph-alignment algorithm achieved a great degree of success, the highlighting-for-translation feature was one of the test users' favorite tools, and the book recommendation system can serve as a basis for future work. In addition, the app was ultimately able to include more features than what was initially expected. In the early planning stages, features such as user personalization with Google sign-in and highlighting-on-both-sides were marked as "stretch goals" – features that would potentially be left as "future work." While not all "stretch goals" were included in the final product, CH-ENG was able to become more multi-functional with such additions.

7.1.2. What didn't go well?

Of course, this project met its fair share of problems during the research and development process. In particular, data gathering and processing proved to be a challenge. As mentioned in section 5.1, the novel data had to be scrapped from Webnovel and Qidian. The scrapped data was often messy and inconsistent in formatting. For example, some text files would repeat the chapter name twice (rather than once); some would also signpost new chapters differently. Take “第一章”， which means Chapter 1, for example. Some would use add symbols, such as a colon (“第一章” versus “第一章:”), and some would use different ":" characters ("章：" versus "章:"). They look the same, but actually have a different Unicode, so they are by default treated as different characters in a program. These inconsistencies made creating a parser program to extract and organize the information difficult, as there were many special cases to account for.

In addition to messy data, a challenge was also the lack of data that could specifically help the app classify the reading difficulty of web novels. If more information were available, such as an already pre-made dataset with difficulty scores for web novels, or concrete data on how certain book elements (i.e genre) can impact reading difficulty, then developing the book recommendation algorithm would not have taken as much guess-work. If backed by web novel specific data, the algorithm results would also have much more "validity."

7.1.3. What could have been done differently?

In hindsight, something that could have been done differently was the database schema. As mentioned in section 5.2.2, it could have been designed to be more scalable. This would not only be helpful when implementing future features, but would also help the app be sustainable in the long term. Another aspect that could have been done differently was spending more time on adding books to the app. While the focus of the project was on the features (rather than the books themselves), having more data would perhaps have made a difference when analyzing algorithm performance or noticing trends. However, these drawbacks have inspired ideas for future work, which will be elaborated in the next section (7.2).

7.2. Future Work

There are four categories of possible future work:

1. Adding More Diverse Books with More Data
2. Improving the Algorithms with ML
3. Features to Improve Reading Experience
4. Features for More User Involvement

7.2.1. Adding More Diverse Books with More Data

As mentioned in section 6.0.2, one of the possible ways to improve the book recommendation algorithm is to have a larger book collection. Currently, the app only has ten books. While genres range from romance to science-fiction, there are also many more genres available that are not included, such as historical fiction. More books can be added to not only expand the amount of data to work with, but also to increase diversity. Additionally, the app can relay more information about the books, such as the genre. With such information, perhaps the relationship of a book's difficulty with its genre can even be explored.

7.2.2. Improving the Algorithms with ML

As discussed in section 6.0.1 and 6.0.2, there is much room for improvement in both algorithms. However, rather than tweaking and experimenting with the scoring systems, a possible direction could be to use machine learning (ML). For the paragraph-alignment algorithm, ML could be used to determine the best-fit English paragraph through training a model on a dataset of "good matches." This may be a better alternative than the current methodology of playing with different weights for the factors, as the model could dynamically determine what factors to prioritize based on the training set. For the book recommendation algorithm, ML could be used to give books a reading difficulty score. By providing a dataset of Chinese passages with a difficulty score, a model can be trained to provide a score for each book (or maybe even for each chapter). With numbers, users can compare the books themselves, understand their reading level, and plan their own reading sequences.

7.2.3. Features to Improve Reading Experience

Ultimately, CH-ENG is also a novel-reading application, and one of the most important aspects of such an app is the reading experience. During the second round of evaluations, the users gave many great suggestions for features that would improve their reading experience. They include:

1. a personal library with the books the user has read or are reading
2. a reading progress bar for a chapter and/or book
3. a smart search bar to easily find books
4. dark mode
5. indicating word count for chapter and book
6. changeable text size

7.2.4. Features for More User Involvement

Currently, users don’t have many ways to express their thoughts or opinions on the app other than through the recommendation form. Hence, more features could be added to have users become more involved in the app and to possibly build up a community of similar users. One such feature is to allow them to comment on books. These comments could provide other users insights on how interesting the book is and perhaps even hint at the reading difficulty. Another potential feature is a user rating system for the book. However, in spirit of the app being for language learning, the rating could be for reading difficulty. With enough users, this metric may be even more accurate than any machine-generated score, as the users would be suggesting from experience. A possible feature that could help with the paragraph-alignment issues is to have users report any passages that are misaligned. This would not only help catch errors, but would also be helpful with adjusting the algorithm if there are trends in what is being reported. Finally, another possible feature could be for users to upload their own content. This would help automate the process of adding content to the app and reduce the weight on the developer's end for picking the books. This would also open doors for more personalized use.

8. Acknowledgements

I would like to acknowledge Professor Dondero, my advisor who graciously met with me weekly to give guidance, feedback, and suggestions for this project. I would also like to thank 吕老师 for being my second reader and showing great interest in the app, giving me feedback and advice about my project. I would also like to thank 欧老师, who met with me to give her thoughts about the app. In addition to Professor Dondero who was also a test user for my app, I would like to thank my friends who participated and gave me emotional support: Zhan Xian Chen, Michael Lee, Anna Hattle, Amy Xu, and Wendy Li. Zhan Xian Chen was also kind and patient enough to help me translate any English in the app to Chinese. Thank you all!

9. Honor Code

This paper represents my own work in accordance with University regulations. - SUKI YIP

References

[1] [Online]. Available: https://www.webnovel.com/
[2] [Online]. Available: http://paralleltext.io/
[3] [Online]. Available: https://www.qidian.com/

[4] [Online]. Available: https://www.zamzar.com/

[5] [Online]. Available: https://www.onlinedoctranslator.com/en/

[6] “Cloud translation nbsp;|nbsp; google cloud.” [Online]. Available: https://cloud.google.com/translate

[7] “Google translate.” [Online]. Available: https://chrome.google.com/webstore/detail/google-translate/aapbdbdomjkkjkaonfhkkikfgjllcleb?hl=en

[8] “Mongodb atlas: Cloud document database.” [Online]. Available: https://www.mongodb.com/cloud/atlas

[9] “Most common chinese characters - ordered by frequency.” [Online]. Available: http://hanzidb.org/character-list/by-frequency

[10] Feb 2020.

[11] D. Baumgold. Available: https://flask-dance.readthedocs.io/en/latest/

[12] Y. T. Chen, Y. H. Chen, and Y. C. Cheng, “Assessing chinese readability using term frequency and lexical chain,” *Computational Linguistics and Chinese Language Processing*, vol. 18, no. 2, p. 1–18, Jun 2013.

[13] R. Cheung, “China’s online publishing industry – where fortune favours the few, and sometimes the undeserving,” *Post Magazine*, May 2018. Available: https://www.scmp.com/magazines/post-magazine/long-reads/article/2144610/chinas-online-publishing-industry-where-fortune

[14] Flamish, “Novel grabber.” Available: https://github.com/Flameish/Novel-Grabber

[15] W. Nishino, *She’s Alive!* Webnovel.

[16] L. of the Paladin, *Trouble Came After All*. Webnovel.

[17] M. Ru, *Souls Actually Exist after Death?* Webnovel.

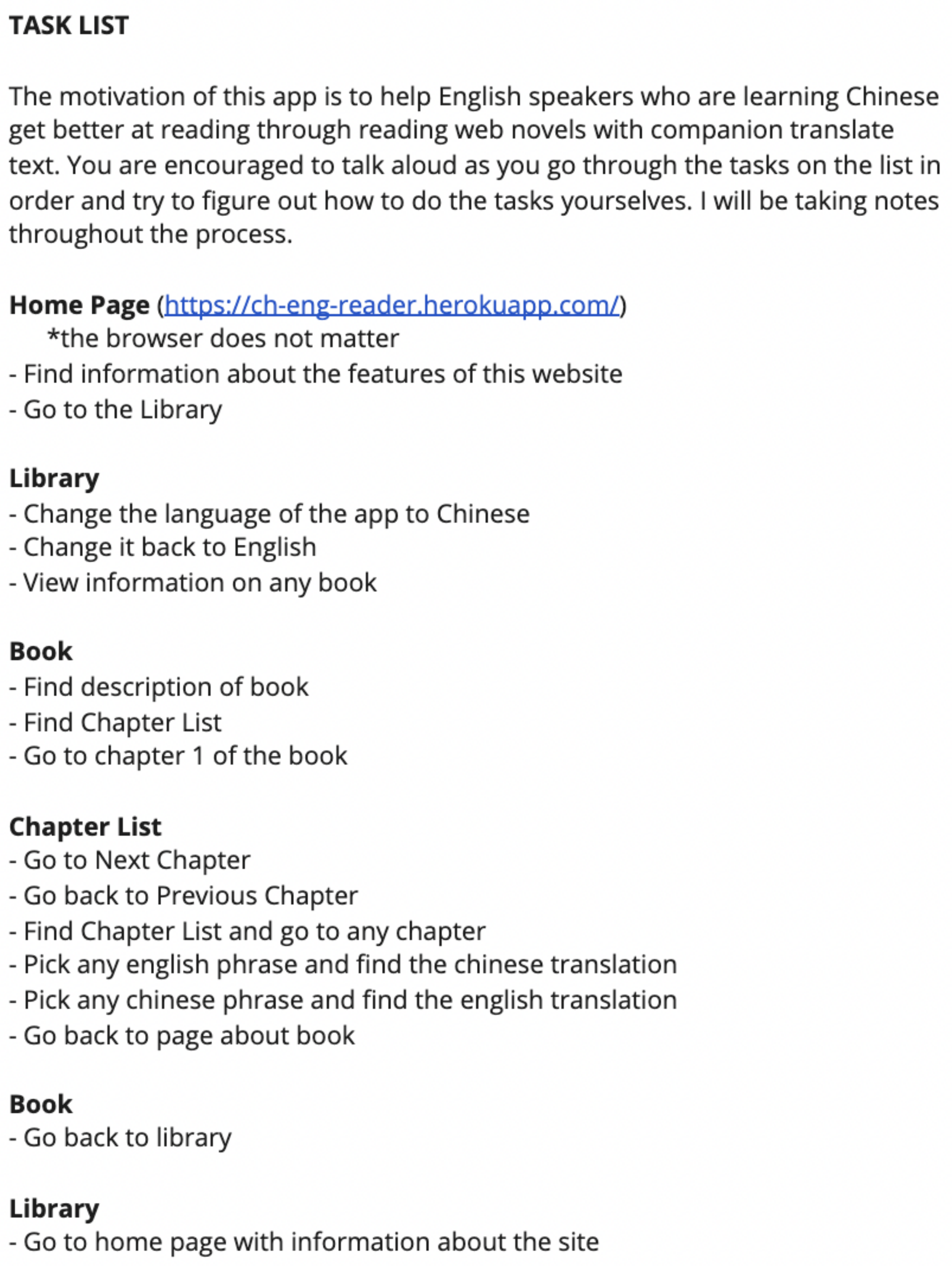
[18] Y.-T. Sung *et al.*, “Crie: An automated analyzer for chinese texts,” *Behavior Research Methods*, vol. 48, no. 4, p. 1238–1251, 2015.

[19] Super, “A dog’s life -the spanish dichotomy.” Available: http://www.dualtexts.com/parallel/316-a-dog-s-life

[20] H. Wang, X. He, and G. E. Legge, “Effect of pattern complexity on the visual span for chinese and alphabet characters,” *Journal of Vision*, vol. 14, no. 8, p. 6–6, 2014.

10. Appendix

Figure A1: The task list for the first round of user evaluations.



TASK LIST

The motivation of this app is to help English speakers who are learning Chinese get better at reading through reading web novels with companion translate text. You are encouraged to talk aloud as you go through the tasks on the list in order and try to figure out how to do the tasks yourselves. I will be taking notes throughout the process.

Home Page (https://ch-eng-reader.herokuapp.com/)

*the browser does not matter

- Find information about the features of this website
- Go to the Library

Library

- Change the language of the app to Chinese
- Change it back to English
- View information on any book

Book

- Find description of book
- Find Chapter List
- Go to chapter 1 of the book

Chapter List

- Go to Next Chapter
- Go back to Previous Chapter
- Find Chapter List and go to any chapter
- Pick any english phrase and find the chinese translation
- Pick any chinese phrase and find the english translation
- Go back to page about book

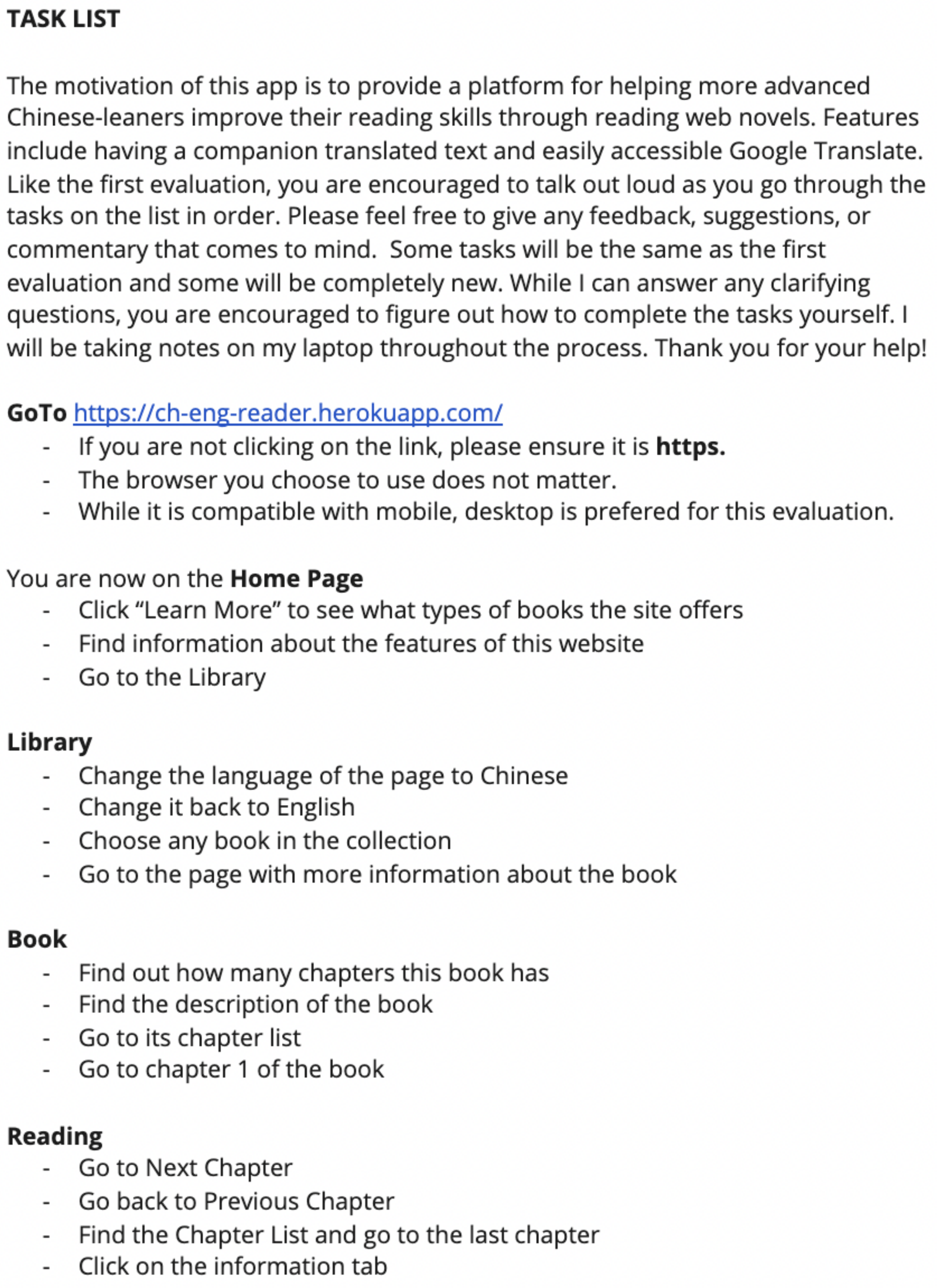
Book

- Go back to library

Library

- Go to home page with information about the site

Figure A2: The task list for the second round of user evaluations.



TASK LIST

The motivation of this app is to provide a platform for helping more advanced Chinese-leaners improve their reading skills through reading web novels. Features include having a companion translated text and easily accessible Google Translate. Like the first evaluation, you are encouraged to talk out loud as you go through the tasks on the list in order. Please feel free to give any feedback, suggestions, or commentary that comes to mind. Some tasks will be the same as the first evaluation and some will be completely new. While I can answer any clarifying questions, you are encouraged to figure out how to complete the tasks yourself. I will be taking notes on my laptop throughout the process. Thank you for your help!

GoTo https://ch-eng-reader.herokuapp.com/

- If you are not clicking on the link, please ensure it is **https.**
- The browser you choose to use does not matter.
- While it is compatible with mobile, desktop is prefered for this evaluation.

You are now on the **Home Page**

- Click “Learn More” to see what types of books the site offers
- Find information about the features of this website
- Go to the Library

Library

- Change the language of the page to Chinese
- Change it back to English
- Choose any book in the collection
- Go to the page with more information about the book

Book

- Find out how many chapters this book has
- Find the description of the book
- Go to its chapter list
- Go to chapter 1 of the book

Reading

- Go to Next Chapter
- Go back to Previous Chapter
- Find the Chapter List and go to the last chapter
- Click on the information tab

- Pick any english phrase and find its chinese translation
- Pick any chinese phrase and find the english translation
- Go back to the page with information about the book

Suggestions

- See the Suggested Next Readings for this book
- Click on any suggested book
- Go to the page for giving a recommendation
- Give a recommendation
- Go back to the library

Library

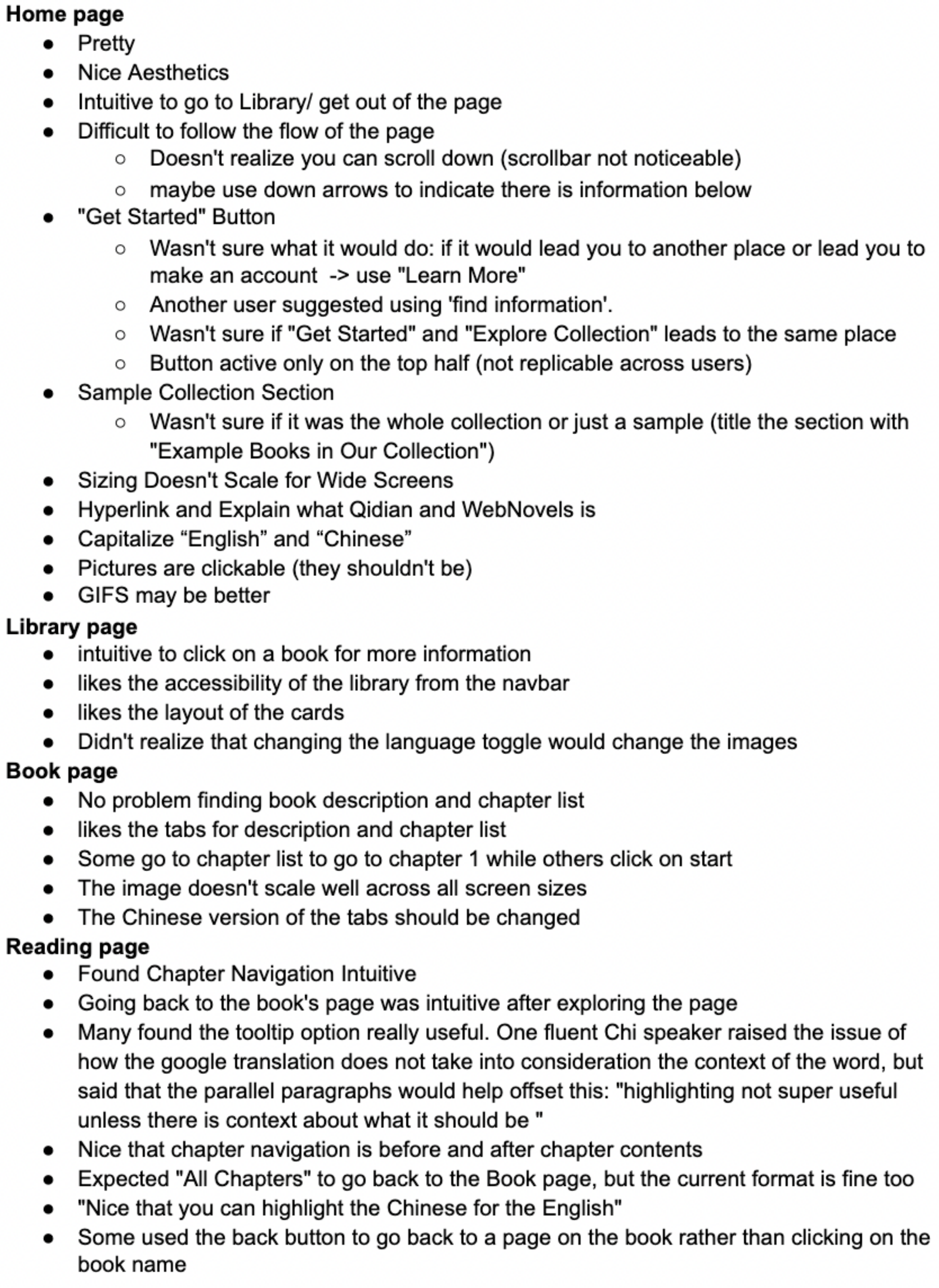
- Go back to the home page (the page with the features listed)

(optional)

Login

- Login to the application with Google Sign-In
- Go to the library
- Pick any book and go to chapter 3 of that book
- Go back to the page with information about the book
- Resume reading from chapter 3
- Logout

Figure A3: Aggregate field notes from the first round of user evaluations.



Home page

- Pretty
- Nice Aesthetics
- Intuitive to go to Library/ get out of the page
- Difficult to follow the flow of the page
  - Doesn't realize you can scroll down (scrollbar not noticeable)
  - maybe use down arrows to indicate there is information below
- "Get Started" Button
  - Wasn't sure what it would do: if it would lead you to another place or lead you to make an account -> use "Learn More"
  - Another user suggested using 'find information'.
  - Wasn't sure if "Get Started" and "Explore Collection" leads to the same place
  - Button active only on the top half (not replicable across users)
- Sample Collection Section
  - Wasn't sure if it was the whole collection or just a sample (title the section with "Example Books in Our Collection")
- Sizing Doesn't Scale for Wide Screens
- Hyperlink and Explain what Qidian and WebNovels is
- Capitalize “English” and “Chinese”
- Pictures are clickable (they shouldn't be)
- GIFS may be better

Library page

- intuitive to click on a book for more information
- likes the accessibility of the library from the navbar
- likes the layout of the cards
- Didn't realize that changing the language toggle would change the images

Book page

- No problem finding book description and chapter list
- likes the tabs for description and chapter list
- Some go to chapter list to go to chapter 1 while others click on start
- The image doesn't scale well across all screen sizes
- The Chinese version of the tabs should be changed

Reading page

- Found Chapter Navigation Intuitive
- Going back to the book's page was intuitive after exploring the page
- Many found the tooltip option really useful. One fluent Chi speaker raised the issue of how the google translation does not take into consideration the context of the word, but said that the parallel paragraphs would help offset this: "highlighting not super useful unless there is context about what it should be "
- Nice that chapter navigation is before and after chapter contents
- Expected "All Chapters" to go back to the Book page, but the current format is fine too
- "Nice that you can highlight the Chinese for the English"
- Some used the back button to go back to a page on the book rather than clicking on the book name

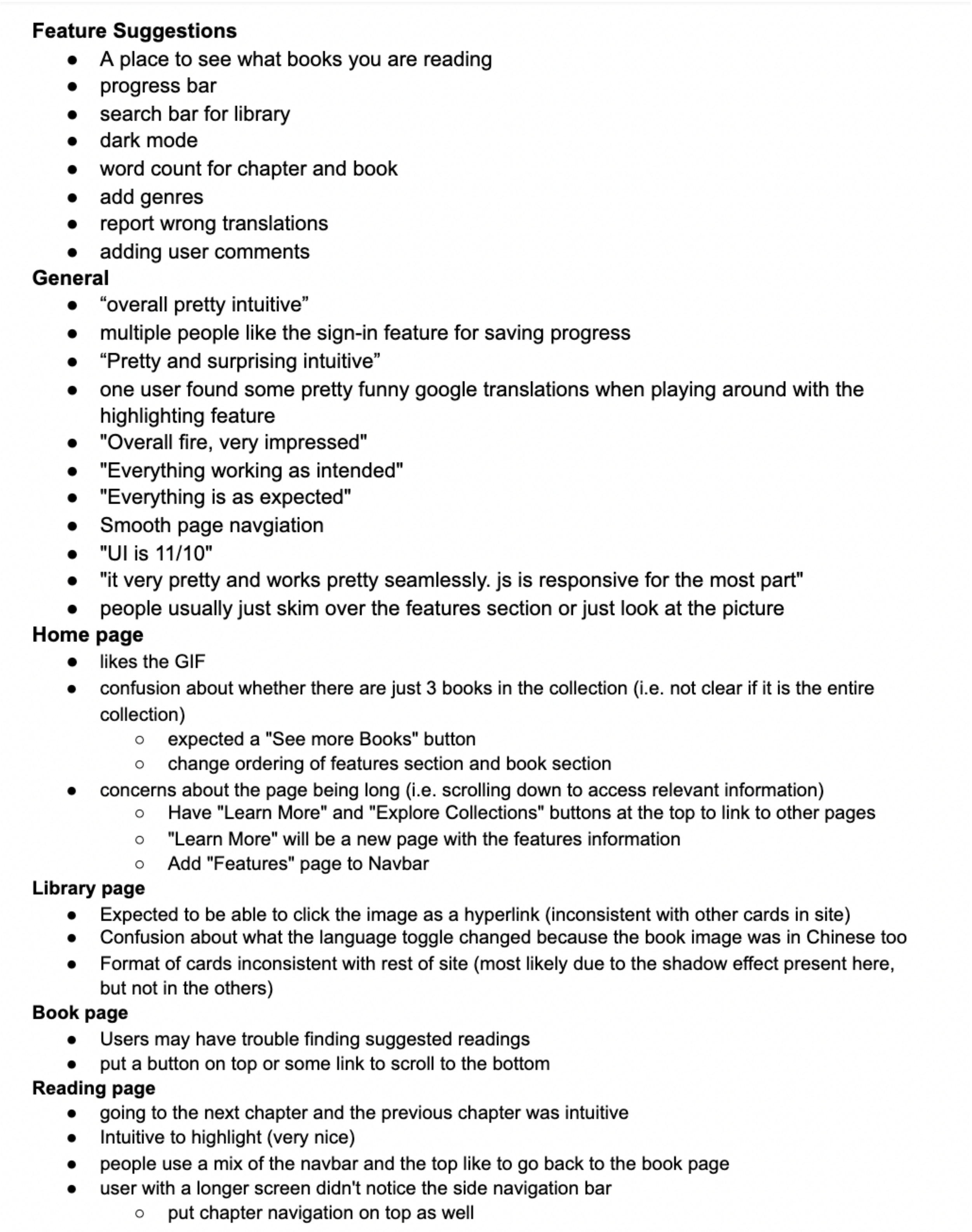
Miscellaneous

- Intuitive that clicking logo will go back to home page
- CH_ENG logo on top can have a better fade (currently blends too well with background)

Language Toggle

- Confused why the Home page, footer, and title of the app are not translated when the toggle switches. "Inconsistency of some parts showing both and some just 1 language"
- "The color choice is very unintuitive. The grey, to me, would indicate disabling."
- radio buttons would be better, or just two buttons on the navbar
- "no point of the button if it just changes layout" -- suggested to just keep the button for the library page and for everything else, explicitly put the two languages together on the page
- The toggle broke when the user played around with chapter navigation (switching it to Chinese would not make it stay on Chinese)
- "Don't flip things with the toggle-- there's no point. The format should stay the same."
- The user was confused about why the button just flips the format for some pages-- did not see the point until explicitly pointed out what changed.
- Sees the button as a translation button
- Confused about the extent the parallelism goes
- Take out the toggle
- Only have a toggle for Library
- Add translations for Missing Parts

Figure A4: Aggregate field notes from the second round of user evaluations.



Feature Suggestions

- A place to see what books you are reading
- progress bar
- search bar for library
- dark mode
- word count for chapter and book
- add genres
- report wrong translations
- adding user comments

General

- “overall pretty intuitive”
- multiple people like the sign-in feature for saving progress
- “Pretty and surprising intuitive”
- one user found some pretty funny google translations when playing around with the highlighting feature
- "Overall fire, very impressed"
- "Everything working as intended"
- "Everything is as expected"
- Smooth page navgiation
- "UI is 11/10"
- "it very pretty and works pretty seamlessly. js is responsive for the most part"
- people usually just skim over the features section or just look at the picture

Home page

- likes the GIF
- confusion about whether there are just 3 books in the collection (i.e. not clear if it is the entire collection)
  - expected a "See more Books" button
  - change ordering of features section and book section
- concerns about the page being long (i.e. scrolling down to access relevant information)
  - Have "Learn More" and "Explore Collections" buttons at the top to link to other pages
  - "Learn More" will be a new page with the features information
  - Add "Features" page to Navbar

Library page

- Expected to be able to click the image as a hyperlink (inconsistent with other cards in site)
- Confusion about what the language toggle changed because the book image was in Chinese too
- Format of cards inconsistent with rest of site (most likely due to the shadow effect present here, but not in the others)

Book page

- Users may have trouble finding suggested readings
- put a button on top or some link to scroll to the bottom

Reading page

- going to the next chapter and the previous chapter was intuitive
- Intuitive to highlight (very nice)
- people use a mix of the navbar and the top like to go back to the book page
- user with a longer screen didn't notice the side navigation bar
  - put chapter navigation on top as well

- harder to highlight the right side
- Nit: Highlighting part of an English word still translates it
- Not sure what was meant by “Chinese” and “English” highlighted together in the information popup
- At the chapter list, user thought they had to scroll all the way down to see the last chapter, but they were able to quickly figure out how the list works
- Thought the book icon was an information tab at first, but highlighting over it cleared the confusion
- BUG: information modal results in a tooltip in the background, "I thought the information tab was a tutorial"

Recommendations page

- Be more proactive, rather than reactive, about error handling
- User may be expecting to give an evaluation for the book by the phrasing "book recommendations"