Erasmus Mundus Master's Degree in

Language and Communication Technologies

MSc in Language Science and Technology, Saarland University

MSc in Natural Language Processing, University of Lorraine

NLP Methods to Automate Language Learning through Extensive Reading

Master Thesis

Author

Siyana Pavlova

Supervisors

Prof. Dr. Günter Neumann

Prof. Dr. Josef van Genabith

October 19, 2020

Declaration of Authorship

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Ich versichere, dass die gedruckte und die elektronische Version der Masterarbeit inhaltlich übereinstimmen.

Declaration

I hereby confirm that the thesis presented here is my own work, with all assistance acknowledged. I assure that the electronic version is identical in content to the printed version of the Master's thesis.

Saarbrücken, 19. October 2020

Signature: 

Abstract

NLP Methods to Automate Language Learning through Extensive Reading

by Siyana Pavlova

A variety of language learning applications exist and are widely used today. However, in most cases they rely on manual content creation, leading to slower expansion of the content covered within each language pair. In an effort to speed up this process, we propose a solution that uses Extensive Reading at its core and a number of methods to collect and process content automatically. A proof-of-concept language learning web application was built for English speakers learning German to test the effectiveness of this solution. The feedback obtained in a pilot user study shows that the platform was well received and users felt it helped them improve their German language skills.

Acknowledgements

Preparing this thesis has been an adventure and I would like to thank all the people who made it possible. In particular:

My main supervisor, Prof. Dr. Günter Neumann, first, for agreeing to the thesis topic I proposed, but more importantly for all the feedback, ideas and resources given and for taking the time to listen to my ideas and review my work.

Saadullah Amin, for taking the time to participate in all the meetings, for all the feedback and ideas, and for providing me with his in-context translation experiment setup (despite it not being used in this current version).

My second supervisor, Prof. Dr. Josef van Genabith for taking the time to review my thesis.

Elitsa, Ena, Kelvin, Mihaela, Yordan and Ziyan for taking the time to participate in my user study and providing me with the first user feedback for this system.

Marta for trying out the system and giving me useful user feedback, but more importantly for all the conversations and shared experiences.

My parents, for their constant support, not only in the past few months, but also throughout all the years of my education.

Finally, Veit, for all the moral support, especially during the more difficult parts in the last few months, and for inspiring me to choose this particular thesis topic in the first place.

Content

Chapter 1

Introduction

1.1 Motivation

Learning a foreign language is an essential skill, a compulsory part of most educational systems' curricula and a required skill for a large amount of jobs. Classroom exercises, however, are not enough and for a long time self-study has been facilitated by books, audio recordings and flashcards, to name but a few. Modern technology has managed to enhance these and make them more efficient. A variety of online language learning platforms exists nowadays. However, most of them rely on content creators or users to create new content, vocabulary lists or reading materials and do not make use of automated tools for collecting and processing such information.

This thesis proposes a proof-of-concept language learning system which combines two powerful methods - Extensive Reading, and Spaced Repetition along with automated content collection and processing to help users in their language learning process.

1.1.1 The benefits of extensive reading and spaced repetition for language learning

Extensive Reading (ER) is defined as the independent reading of a large quantity of material for information or pleasure [1]. According to Day and Bamford, the primary goal of ER programmes is "to get students reading in the second language and liking it". ER can be opposed to Intensive Reading which consists in students spending a lot of time reading, analysing and discussing short, difficult texts under the supervision of a teacher, the emphasis being on the students developing reading and language skills. Steve Kaufmann, founder of LingQ[1], argues in his book [2] that learning a foreign language should not be the primary goal, but rather being exposed to the language

[1] `https://www.lingq.com/`

through other activities (e.g. learning about another subject, but in the target language). A study by Mason and Krashen [3] has demonstrated the benefits of ER to foreign language acquisition, and Bell [4] even shows that ER is more beneficial to reading speed and reading comprehension than Intensive Reading is.

Spaced Repetition is a memorising technique which relies on the concepts of spacing and lag effects: a piece of information fades from memory if it is not recalled for a certain period of time. Spaced Repetition is the process of recalling that piece of information just before it fades from memory. It has been shown to be beneficial to second language acquisition [5]. A number of spaced repetition methods exist, including the Pimsleur Method [6], the Leitner System [7] and there is ongoing research on new techniques which use the advancements of Machine Learning [8]. Duolingo, for example, uses Half-Life Regression (inspired by psychological theory and modern machine learning techniques). Newer methods include the Ebbinghaus model [9] a.k.a the *forgetting curve*, ACT-R [10] and Multiscale Context Model [11].

1.1.2 The need for automated content for language learning

While a number of language learning applications exist, most of them concentrate on the most popular language pairs where there is a large amount of speakers, learners and content. However, for native speakers of smaller languages and for people who want to learn smaller languages, finding resources can often be difficult. The amount of possible language pairs is also vast and creating content can be very time-consuming, especially for smaller languages. However, automation could lift off a lot of the workload necessary from content creators. Texts in different language pairs can be collected and processed by automated tools and then only reviewed and edited manually, instead of being collected and processed manually too, thus speeding up the process of making content available for more language pairs.

1.2 Contributions

The main contributions of this thesis are as follows:

- Compared current language learning system that use Extensive Reading and/or Space Repetition as their core features.
- Proposed a new language learning application which uses Extensive Reading and Spaced Repetition, and which relies on automated methods for collection and processing of the content.

- Developed a proof-of-concept system which implements parts of this solution.
- Carried out a preliminary user study, which showed that such a system is well received by users. Users also stated that they were rather likely to use the improved version of this system, which implements the full solution proposed in this thesis.
- Collected 1,200 articles in English from Wikipedia across two difficulty levels (easy and standard) and six different topics, 519 articles in standard German from Wikipedia and 300 articles in easy German from Hurraki, both across the same six topics. Each article has been annotated with its difficulty level and topic. The annotated data has been made publicly available[2] on GitHub.
- The code for the proof of concept system has also been made publicly available[3]

1.3 Marker's Guide

In chapter 2, we discuss the motivation for this project, present the related work and outline the background frameworks/knowledge.

In chapter 3 we describe the system architecture and present the design choices and implementation.

In chapter 4, we introduce the measures and experiments used for evaluating the system. We then present the results of the experiments.

In chapter 5 we discuss the results obtained by the system and analyse its performance.

We summarise our work and outline some areas worth exploration in chapter 6.

[2] `https://github.com/siyanapavlova/thesis-dataset`
[3] `https://github.com/siyanapavlova/master_thesis_django`

Chapter 2

Background and Related Work

In this chapter we summarise and compare some existing language learning systems and present the related work.

2.1 Language Learning Applications

A number of language learning systems exist, relying on extensive reading or spaced repetition and, in some cases, a combination of both. Some systems from each group are presented here.

2.1.1 Systems Using Extensive Reading

2.1.1.1 LingQ

LingQ[1] was founded in 2007 by Steve Kaufmann and his son Mark and currently has over 1,000,000 users. The system is based on Steve Kaufmann's personal experience learning foreign languages and relies on Extensive Reading. Users can select from texts in about 40 different languages and at varying levels. The system allows users to highlight unknown words in the texts they read, get translations for them and save them to a vocabulary list. Words that have been added to the user's vocabulary can be revised in the form of quizzes or flash cards.

There are two main shortcomings to LingQ. Firstly, content is created entirely manually. While this ensures high quality content, it means that the expansion in the resources available for each language, and more importantly, expansion towards smaller languages and smaller language pairs is slow. Secondly, LingQ, like most existing language learning platforms, offers a free and a premium version. However, in LingQ's case, the free version

[1] `https://www.lingq.com/`

allows a user to save only up to 20 words in their vocabulary. When a user's vocabulary list is full, the user is no longer provided with translations for unknown words in the texts, thus rendering the system useless, unless they are willing to pay for the premium version.

2.1.1.2 Bliu Bliu

Bliu Bliu[2] was another language learning system based on Extensive Reading. However, it has been discontinued, but for completeness' sake it is presented here. Bliu Bliu offered short texts across a number of topics aimed to help users read in their target languages. To help users remember new words, Bliu Bliu used a mode in which it showed short easy sentences with no other difficult words so that users can see how these words are used in the language.

2.1.1.3 Leaning with Texts

Learning with Texts (LWT)[3] is another Extensive Reading based platform. It is a free and open-source platform, which can be used as a desktop application. It allows users to upload texts in their target language. The system highlights words that the user has not come across yet. When the user clicks a word, they are presented with a dictionary entry for that word and can add it to a list of words they want to study. These lists can then be uploaded to a Spaced Repetition system (e.g. Anki, described below) to help users learn it.

The main shortcoming of LWT is that installing the software is rather time-consuming and technical and thus may not be appropriate for users who do not have a strong Computer Science or technical background.

2.1.2 Systems Using Spaced Repetition

2.1.2.1 Memrise

Memrise[4] is a language learning application (both web and mobile). It was founded in 2010 and currently has about 40 million users worldwide. Memrise uses Spaced Repetition to help its users learn vocabulary and useful phrases in their target language. Words and phrases are organised in courses. Each course can contain any number of

[2] `https://www.youtube.com/watch?v=SDMhO-cIjkI`
[3] `https://sourceforge.net/projects/lwt/`
[4] `https://www.memrise.com/`

words and/or phrases (which we will call "items" hereafter). Courses can be generic (e.g. frequency lists) or cover specific topics and can cover different language proficiency levels. Memrise offers a number of official courses created by their own content creators for a number of language pairs where both languages are languages with a large number of native speakers. However, most of the courses offered on Memrise, are created by users. Thus, another powerful feature offered by Memrise is that it lets users create their own courses and make them publicly available if they wish so.

2.1.2.2 Duolingo

Duolingo[5] is probably the most well-known and widely used language learning platform. Duolingo offers its users a level-based system where they can develop new "skills" (vocabulary and grammar related) in their target language by doing a variety of tests and quizzes - tapping tests where users have to organise the words in a sentence, translation tests, typing tests, etc. Duolingo uses Machine Learning approaches to generate content and tailor it to its users. This often results in funny sentences, such as "The giraffe went to the bar", which has been criticized by users. However, Duolingo claim that it is done on purpose, because it helps users remember these structure better. The main feature that Duolingo does not offer and we are interested in developing as part of our system is Extensive Reading.

2.1.2.3 Anki

Anki[6] is a free, open-source language learning platform that uses spaced repetition. It is very similar to Memrise in the way it organises new words and phrases in vocabulary lists. However, the content on Anki is entirely manually created. Anki is supported by a large community of users who create their own courses and make them available to anyone. Unfortunately, Anki does not provide an Extensive Reading component, which is the main component considered in our system.

2.1.2.4 Lingvist

Lingvist[7] uses a spaced repetition method in which it presents users with short sentences containing their target vocabulary, across a variety of topics. Vocabulary items are then reviewed periodically, requiring the user to type the word they have learnt in the context

[5] `https://www.duolingo.com/`
[6] `https://apps.ankiweb.net/`
[7] `https://lingvist.com/`

	LingQ	**Bliu Bliu**	**LWT**	**Memrise**	**Duolingo**	**Anki**	**Lingvist**	**Glossika**	***This system***
Extensive Reading	Yes	Yes	Yes	No	No	No	No	No	Yes
Vocabulary Builder	Yes	Yes	Yes	Yes	No	Yes	Yes	No	Yes
Spaced Repetition	Yes	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Reasonable free version	No	-	Yes	Yes	Yes	Yes	No	No	Yes
Automated Content Creation	No	Unk	No	No	Yes	No	Unk	Yes	Yes

TABLE 2.1: Comparison of requirements of similar systems.

of a sentence. However, Lingvist is a paid application and does not yet offer a large variety of languages.

2.1.2.5 Glossika

Glossika[8] is a language learning platform offering courses over in 60 languages. Glossika uses a spaced repetition technique for its users to learn words and short sentences in their target language and offers a variety of modes for practice - listen and repeat exercises, multiple choice quizzes and typing tests. However, the system does not offer Extensive Reading as a feature. Furthermore, Glossika is a paid system, offering only a 7-day free trial to new users.

2.1.3 Comparison

The goal of this project was to create a language learning platform which uses Extensive Reading and Spaced Repetition at its core and aims to automate as large a part as possible of the content collection and processing pipeline.

Furthermore, in order to ensure accessibility, this system should be completely free for all users, or offer a reasonable amount of its features to non-paying users (e.g. paying users may be able to get more personalised content or track their statistics).

While all of the systems described above provide at least one of the five main requirements that we want our system to have, none of them provide all, as can been seen in table 2.1

2.2 Text Difficulty

Having too many unknown words or too complex grammar structures in a text can be frustrating for a new language learner. Therefore, it is essential that the difficulty level of a text can be determined.

[8] `https://ai.glossika.com/`

The difficulty of a text, as it stands, can be influenced by a number of factors:

- Lingustic features:
  - Some morphemes or morphological structures are more complex
  - Some syntactic structures are more complex than others
- Average sentence length
- Amount of unknown words
- Length of unknown words

In order to determine text difficulty, these factors need to be addressed. This can be done using a set of morphological and grammar rules, similarly to iREAD's domain model [12]. Furthermore, heuristics on the amount of unknown words, average word length and average sentence length can be used.

Text difficulty is also dependent on the source and target languages of the user. For example, if an Italian speaker is learning Spanish and German, a text in Spanish will be easier for them to read than the same text in German. A text difficulty module should take the similarity between language pairs into account too. Such a similarity can be based on the number of cognates [13] that the two languages share and the number of cognates present in the text.

2.3 Topic Analysis

The two most commonly used techniques for Topic Analysis are Topic Modeling and Topic Classification.

Topic Modeling is an unsupervised ML technique which can scan a collection of text documents, detect patterns in the words and phrases that appear in these documents, and cluster word groups that characterise the document. Topic Classification, on the other hand, is a supervised technique, which works with a predefined set of topics that are said to appear in a set of documents.

A number of algorithms exist for both techniques. The most popular Topic Modeling algorithms are Latent Semantic Analysis [14] and Latent Dirichlet Allocation [15]. Topic classifiers can be built using rule-based systems, ML techniques (Naive Bayes, Support Vector Machines (SVM) or some Deep Learning techniques) or a combination of both.

Chapter 3

Design and Implementation

In this chapter we discuss the various stages of development of the system.

3.1 Functional Requirements (FR)

The starting point for developing the system was to define a set of functional requirements (FR) around which to build it. The requirements were split into *Core requirements* and *Additional requirements*. The proof-of-concept version of the system implements only the former. However, the design decisions taken with respect to the system (e.g. database architecture and overall system architecture) were taken by keeping both Core and Additional requirements in mind. Furthermore, the Additional requirements will play a central role in prioritising different future work aspects. Thus, it is important that both are outlined here.

3.1.1 Core requirements

FR1. User should be able to read texts in their target language. These texts should:

- **a.** Be appropriate for their language level.
- **b.** Correspond to their interests.

After a user has been assigned a level (or has selected one) and chosen a set of topics they are interested in, they should be able to get texts in their target language ranked in relevance to their preferences and level.

FR2. User should be able to mark unknown words. When presented with a text in their target language, invariably the user will find that some of the words in this text are unknown. The system should allow for the user to mark these words as unknown.

FR3. User should be able to get translations for unknown words. When a user sees an unknown word in a text, after clicking on it, they should be provided with a translation of this word in their source language.

FR4. User should be able to revise new vocabulary. Words that the user has marked as unknown should become part of the user's vocabulary. The system should provide a module where these words can be revised at specific time intervals determined by a spaced repetition algorithm.

3.1.2 Additional requirements

FR5. System should have an infrastructure which makes it possible to learn from users and update models accordingly. This learning will cover what texts users find easy or difficult, what texts they find to be appropriate for the topics they have chosen, what words and translations they find difficult to learn.

FR6. User should be able to view and edit their vocabulary.

- **a.** Add new words.
- **b.** Propose different translations.

When provided with a translation that the user does not find correct or helpful, they should be allowed to propose different translations, which are then visible to other users too. In their own vocabulary lists, users should be allowed to make edits and add new words.

FR7. User should be able to add their own texts. The system should allow the user to upload their own texts that are then processed by the system (tokenized and translated) and can be read by the same user as other texts available on the system.

FR8. System should be able to determine user's language level/skill based on an initial pre-test. When a user registers, or starts a new language on the system, they should be offered to do a pre-test. This pre-test should require the user to read some texts and mark unknown words in them. Based on the amount of unknown words, the system should be able to determine the user's language level.

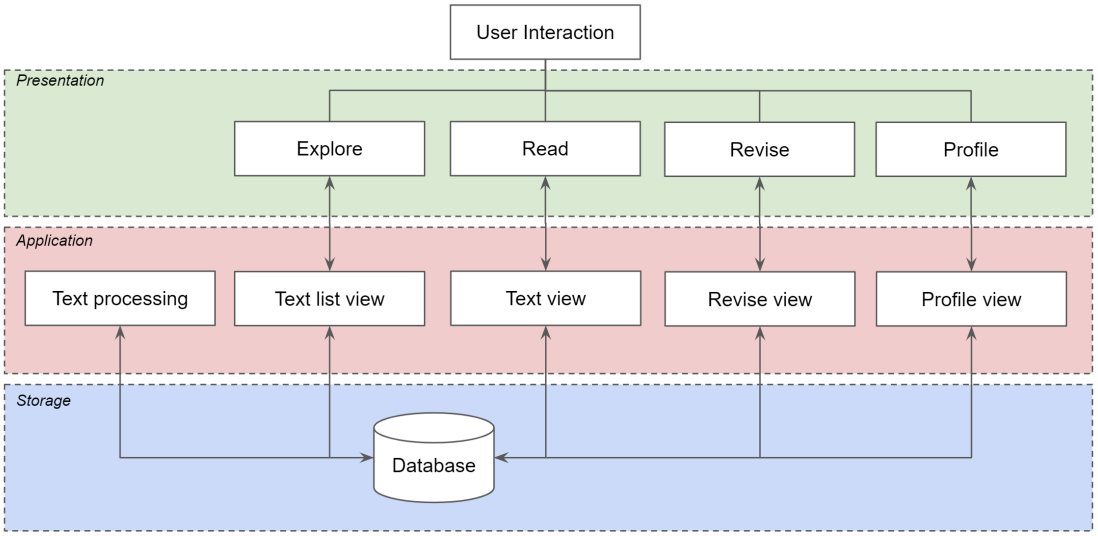

FIGURE 3.1: System Architecture.

3.2 System Architecture

The architecture of the web application was designed based on the functional requirements outlined in section 3.1. The three-tier architecture presents possible user interactions in the Presentation layer, the corresponding back-end functions in the Application layer, and finally the database in the Storage layer. A figure outlining the core system architecture can be seen in figure 3.1.

3.2.1 User Interaction

From the user's perspective, apart from the Login and Signup pages, which are standard for any user-based platform, there are four core page templates: "Explore", "Read" (called "Read" here for convenience, it represents the detailed page for each text),"Revise" and "Profile". A few additional pages (detailed in 3.2.1.5), outside of the core requirements, are also available.

3.2.1.1 Explore

This page was designed to fulfill **FR1**. Here users can explore texts available in the database. These texts are sorted according to the level and topics selected by the user in their profile. This means that the first texts that a user sees are the ones that are most appropriate for their level and match at least one of the topics they are interested in. An example of the *Explore* page for a user who has selected "Science and Technology" as a topic of interest is shown 3.2.

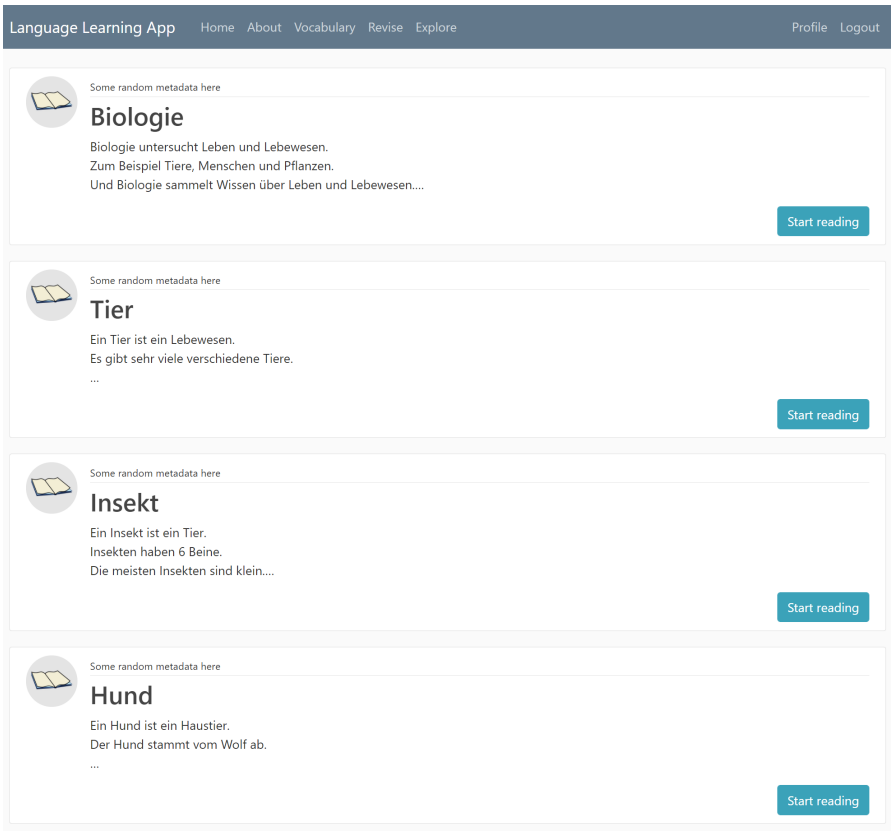

FIGURE 3.2: *Explore* page for a beginner with "Science and Techonology" as topics of interest.

3.2.1.2 Read

The text pages, where a user can see the whole contents of the text they opened, are designed to fulfill **FR2** and **FR3**. Each word in the text that is not in the user's vocabulary, is highlighted by the system and is clickable. When a user clicks such a word, its highlight colour changes and a translation of the word is shown. If a user wishes to change the colour back, they can do so by clicking the word again. Once a user has finished reading the text, they can signify that by pressing a `Done` button. When this happens, the words that the user has clicked are added to their vocabulary as new words that the user should learn. The remaining highlighted ones (that were not marked by the user) are added to their vocabulary as known, so that they are not highlighted if they appear in future texts. An example of the text page, with a few highlighted words for the text "Katze" is shown in figure 3.3.

3.2.1.3 Revise

This page is designed to fulfill **FR4**. Here users are presented with multiple choice quizzes where they have to select the correct translation for a word they have added to

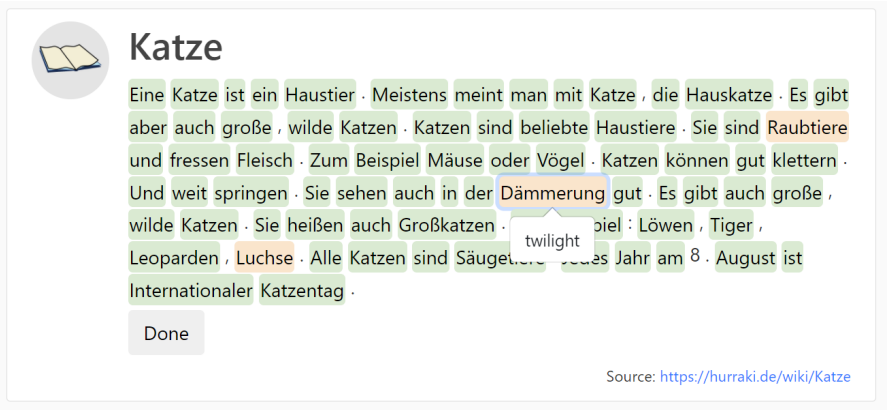

FIGURE 3.3: *Read* page for the text "Katze" for a new user.

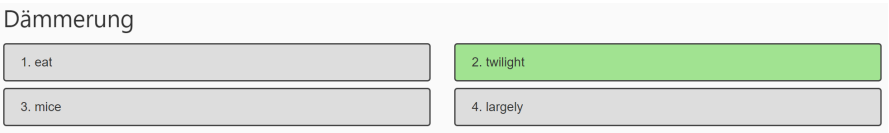

FIGURE 3.4: Entering correct answer to quiz.

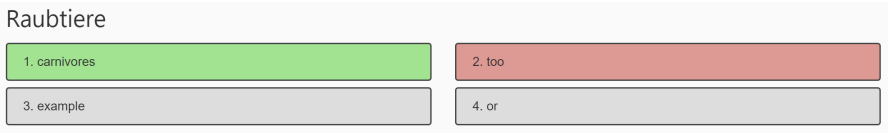

FIGURE 3.5: Entering wrong answer in quiz.

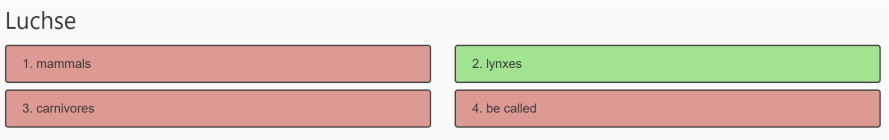

FIGURE 3.6: Pressing "Enter" in quiz.

their vocabulary. The multiple choice quiz includes four options - the correct answer and three distractors. A user can select an answer by clicking on it or by typing its corresponding number. If a user is unsure of the answer and does not want to guess, they can press *Enter* to see the correct answer.

Once a word has been revised, its next revision time is updated. A spaced repetition mechanism is used to determine the next revision time. More details on this are available in subsection 3.2.2. A few sample revision pages, showing the correct answer, wrong answer, and no answer behaviours are shown in figures 3.4, 3.5 and 3.6.

3.2.1.4 Profile

The profile page is where a user can change their profile details. To fully fulfill **FR1**, we need to know the user's language level and the topics they are interested in. The profile

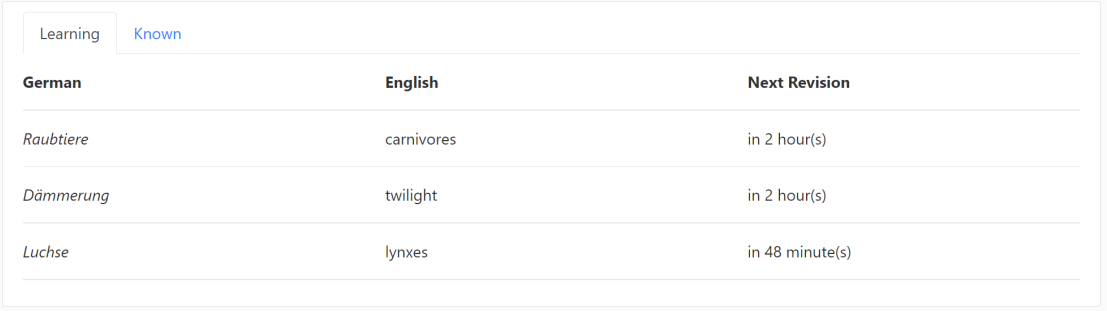

FIGURE 3.7: Vocabulary for a user who has recently read and revised some words.

page is where a user can edit these.

3.2.1.5 Additional pages

Apart from these four core pages, users also have access to two additional pages to make their user experience more pleasant - *Home* page and a *Vocabulary* page.

The *Home* page is where a user can see all the text that they have already read or have started reading. Upon initial registration this page is empty and prompts the user to go to *Explore* to select their first text to read.

The *Vocabulary* page is where a user can see all the words that are already in their vocabulary. The words are split in two groups - `Learning` and `Known`. Additionally, for the words in the `Learning` group, the remaining time until each word is due for revision again is shown. An example of the vocabulary page for a user who has just read the "Katze" text from figure 3.3 above is shown in figure 3.7.

3.2.2 Back-end

Once the necessary pages from a user's perspective were considered, a number of design decisions concerning the back-end of the system had to be taken too, concerning behaviour of the *Explore* page, generating questions along with distractors for each question and choosing and implementing a spaced repetition algorithm for the *Revision* page.

3.2.2.1 Explore

For the *Explore* page, it was important to decide whether to prioritise appropriate level or appropriate topics when there were conflicts. In the current version of the system, topic relevance has a higher priority. This means that if a user has selected, "Beginner"

as their level and "Arts and Culture" and "Sports" as their topics of interest, between two articles, one at a "Non-beginner" level on "Sports" and one at a "Beginner" level on "Food and Drink", the former will be shown first. The reasoning behind this logic is that if a user is interested in a topic, even if a text is too difficult, their curiosity to keep reading about that topic will keep them engaged despite the text difficulty. If the text is too easy, but the topic interesting, the risk of the user disengaging is lower. It is important, however, to keep in mind that text difficulty mediates the effects of topic interetest so there is invariably a tradeoff [16].

3.2.2.2 Read

In the *Read* page, as seen in subsection 3.2.1, a user is presented with a text. We saw that the *Read* page requires that each word is clickable and needs to behave differently depending on whether it is present in the user vocabulary or not. Thus, to comply with this requirement, the back-end logic was designed in such a way that, instead of presenting the text as a whole, it is presented as a sequence of words (tokens), each of which characterized with a feature specifying whether it is a part of the user vocabulary or not.

3.2.2.3 Revise

To generate questions for the quizzes, a number of distractors had to be chosen. Distractors are answers different from the correct answer. The number of distractors was chosen to be three, thus making the number choices in the multiple choice quiz four. When selecting the distractors, it was important to consider the set from which they were chosen. One option was to choose from all the words available for that language pair. However, this means longer processing time and potential confusion for the users if they are presented with translations they have never seen. Therefore, it was decided for the distractors to be chosen among the words that are already in the user's vocabulary (regardless whether the are in **Known** or **Learning**).

The other important factor to consider for the revision page was the spaced repetition algorithm to be used. As seen in subsection 1.1.1, a number of spaced repetition techniques exist. For this proof-of-concept version of the system, the goal was to choose a light, but efficient such technique. As shown in [8], the Leitner System [7] fulfills both of these requirements and it was therefore chosen.

The Leitner system was proposed in 1972 and was intended for use with flashcards back then. The idea behind it (see figure 3.8) is that a student has a deck of flashcards

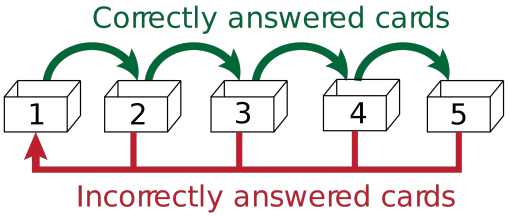

FIGURE 3.8: The Leitner System.

which they keep in a few different boxes or piles, each of which corresponds to a practice interval. All flashcards start with a one-day practice interval. If the student remembers the word on a card correctly, that card gets "promoted" to the two-day interval pile. Next time it is revised (two days later), if correct again, it gets "promoted" to the four-day interval pile, etc. In the same manner, if a student does not remember the word on a card correctly, that card goes to the lower interval pile. In this way, words that the student gets correctly need to be revised less often and words that are difficult for them are revised more often.

In our implementation of the system, whenever a word is added to a student's vocabulary, its next revision time is set to the current time. Once a question for this word has been answered correctly in a quiz, its next revision time is set for one hour from the current moment, then two, then four, etc. Whenever a user gives a wrong answer, the interval is divided by two (e.g. from eight hours to four hours). The minimum interval is set to one hour. The reason for such a short starting interval (compared to one day in the original Leitner system) was to allow for the user study (4.1) to be performed in a shorter period of time.

3.2.2.4 Text processing

Apart from the logic that serves each of the pages that users interact with, the back-end also includes the pipeline used for processing the texts in the system - tokenizing them and translating the tokens.

Once texts were in the database, each of them was tokenized. It is important to note that it is essential to keep punctuation within the tokens. This is strictly necessary in order to build the text in the *Read* page as described in subsubsection 3.2.2.2. If punctuation is omitted during tokenization, the texts presented in the *Read* page will consequently contain no punctuation.

Once a text has been tokenized, its tokens need to be translated in the languages available in the system. For the proof of concept system, the simplest possible translation has been used - the most frequent translation of each word (token) is used. The PenLex Lite corpus[1] was used to obtain these translations.

3.2.3 Database

The database was designed in such a way that it contains all the models necessary for the current version of the system (the core system, **FR1 - 4**), but it also includes features that will allow an easy expansion into the additional requirements (**FR5 - 8**). The most important models are outlined here.

- **Profile**. This is the user profile model. This model captures the topics and language level of a user and it is essential for fulfilling **FR1**.
- **Language**. Each language exists as an item in the database and a number of other models, as shown below are linked to this.
- **Topic**. This model captures all the topics that users can choose from and that texts are about.
- **Text**. This model represents a text in the database. It contains the title and contents for a given text, as well as the URL address from where it was collected. Original URLs were collected in order to be able to track back to where texts came from and to be able to comply with Hurraki's license (CC-BY-SA 3.0). Furthermore, each text is linked to a Language, a set of topics and has a field for difficulty. Finally, to allow for tokenization after the text has been added to the database, a text has a boolean field signifying whether is has been tokenized yet or not.
- **Token**. This model represents a token in the system. A token is characterised by a *form* - the orthographic representation of the token - and a language. This is not to be mistaken with TokenAppearance (below). A token can appear in many texts, whereas a tokenAppearance is linked to a specific location in a specific text.
- **Meaning**. It is important to note that an orthographic word (represented by the Token model in the database) can have a number of different meanings in a language (e.g. "book" can mean "a written text that can be published in printed or electronic form" or "to arrange to have a seat, room, performer, etc. at a particular

[1] `https://dev.panlex.org/panlex-lite/`

time in the future"[2], to name but a few). Thus, the Meaning model was designed to capture the different possible meanings of each orthographic token. This model has not been used in the current implementation of the system, but it will be essential once a different translation algorithm is implemented. An instance of the Meaning model is linked to an instance of the Token model (the orthographic word it is represented by) and includes a field for a definition and a field for a string ID. The string ID for a meaning can be the string by which a meaning is represented in some of the existing sense banks (e.g. PropBank[3] for English where a word sense is represented by its orthographic value, followed by a dot, followed by a sequence number, e.g. "book.01").

- **TokenAppearance**. This model is built to represent each appearance of each token. A token appearance is linked to a text and contains fields to show its span within this text - start and end.

- **Translation**. An instance of the Translation model represents a link between two meanings from different languages.

- **TokenAppearanceTranslation**. This model links a token appearance with the translation of its meaning.

- **UserTranslationRelation**. This model links a user to a token. When a user adds a new translation of a token to their vocabulary, the next revision time is saved as a field in this model. Furthermore, the model keeps track of the number of times a token has been revised by this user and how many times the answer was correct. A boolean field shows whether the word is known or not, which is important for the *Revision* and *Vocabulary* pages.

- **UserTextRelation**. This model links a user to a text and is used when generating the content in the *Home* page and the *Explore* page. The next version of this system includes a feature which splits a text across a few pages, so only a few sentences are read and their vocabulary added at a time. To account for this, the UserTextRelation model also has a field which keeps track of the current position of a given user within a given text.

The next two models are not used in the proof of concept system. However, they address some of the functional requirements outlined in section 3.1 and this is the reason why they are also described here. The first one is linked to **FR5**. The second one is useful for improving the content of the system.

[2] `https://dictionary.cambridge.org/dictionary/english/book`
[3] `https://verbs.colorado.edu/~mpalmer/projects/ace.html`

- **UserTextRating**. This model links a user to a text, but unlike the UserTextRelation model, its purpose is to collect information about the user's opinion on a given text. Fields for a difficulty rating (on a scale from one to five) and a relevance rating (on a scale from one to five) are provided. A larger userbase will allow us to make use of the full potential of this model by collecting data on the difficulty and relevance of a text according to a user. This data can then be used to build a recommendation engine that, when recommending texts to a user, takes into account what texts other users with a similar level and interests found appropriate.
- **UserTokenAppearanceTranslationRating**. This model links a user to a token appearance translation and lets the user rate on whether they found the translation helpful or not. This model will help highlight problematic translations which can be taken a closer look at in the future.

3.3 Implementation

In this section the specific implementation details of the proof-of-concept system are discussed. These include details on the technologies chosen for the implementation and information for the data that has been collected and deployment details.

3.3.1 Technologies

Python 3 was used for the development of the main application. Two Python web development frameworks were considered, namely Django[4] and Flask[5], both of which have advantages and disadvantages over each other. Both have been used for a number of high-traffic websites so both offer good scaling opportunities. However, due to their different philosophies one was more appropriate for this project. Flask is a micro and lightweight web framework, whereas Django is a full-stack web framework which does not rely on third-party tools. Therefore Flask is more suitable for smaller or mid-size applications that use static content[6], including single-page applications and forums. Django, on the other hand, is more appropriate for more complex applications. With consideration of the functional requirements and relative complexity of the proposed system, Django was the more suitable choice.

[4] `https://www.djangoproject.com/`
[5] `https://flask.palletsprojects.com/en/1.1.x/`
[6] `http://www.mindfiresolutions.com/blog/2018/05/flask-vs-django/`

3.3.2 Database

SQL vs NoSQL solutions were considered for the databases of the project. However, since Django comes with support for relational databases and seems to be rather unfriendly towards non-relational database solutions, an SQL solution was the more appropriate choice. During the development stages SQLLite was used. In production, this was upgraded to PostreSQL due to the deployment platform chosen for the system.

3.3.3 Deployment

The final proof-of-concept system has been deployed to Heroku[7]. Heroku is a platform which "enables developers to build, run, and operate applications entirely in the cloud", in their own words. It was chosen, because it supports a number of programming languages, including Python and provides a free tier subscription that can be used for small or prototype applications. One disadvantage of Heroku is that it does not store images. Our system, however, lets users upload and update their profile pictures and also allows for each text to have an image (though not currently used in the platform). The solution implemented, as proposed by this tutorial[8] was to use Amazon Web Services' (AWS) S3 service to store images and proved to be an adequate solution for this version of the system.

The deployed system can be found under this link:

`https://learn-with-lily.herokuapp.com/`

3.4 Data

In this subsection, details about the data that was collected and used in the system are provided. The choice of sources was informed by their selection of texts and by the copyright restrictions for each. The data collected covered six different topics - "Art and Culture", "Science and Technology", "Travel", "Food and Drink", "Business, Economics and Politics" and "Sports" and spans across two levels - an easy set of texts ("Beginner (A1-B1)") and a difficult set of texts ("Non-beginner (B2-C2)"). Texts were collected in two languages - English and German. For the purposes of the experiment, only the German texts have been added to the production version of the system.

[7] `https://www.heroku.com/`
[8] `https://www.youtube.com/watch?v=kt3ZtW9MXhw`

3.4.1 Wikipedia

Wikipedia[9], as the largest collection of freely available text in a variety of languages, was an obvious choice for the proof of concept system. Furthermore, there are a number of Python API's available, which made the collection process rather straightforward once the texts within each topic were determined.

English Wikipedia was used to collect texts in English at a standard level. Simple Wikipedia was used for easy texts in English. Finally, German Wikipedia was used to collect German texts at a standard level. From each Wikipedia page, only the summary of that page was collected. A summary on Wikipedia is the text before the contents table. It can vary in length from a few sentences to a few paragraphs.

Wordlists[10] were used in order to select what articles to include in each topic. 100 article titles were included in each topic. The topics were determined only for English. The German articles were collected by using Python's `wikipedia-api`[11] library's feature which allows us to get the same article page in a diffent language. Since some English articles did not have corresponding German articles, this resulted in fewer German texts.

3.4.2 Hurraki

Hurraki[12] defines itself as "a dictionary in simple German". However, in terms of content, Hurraki is comparable to Simple Wikipedia for English. Hurraki currently contains 3,600 articles on varying topics. Each article contains a simple description of the topic it talks about, equivalent terms, and a broader explanation. Articles may vary in length from two-three sentences to more than 20 sentences. The sentences used are simple, rarely longer than 10 words and use mainly basic German vocabulary. All content is under an Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0) licence. Thus, Hurraki was an ideal source for easy texts for German.

300 articles were collected from Huraki in total (six topics, 50 articles per topic).

A summary on the texts and topics collected for each of Standard English, Easy English, Standard German and Easy German is presented in table 3.1.

[9] `https://www.wikipedia.org/`
[10] `https://www.enchantedlearning.com/wordlist/`
[11] `https://pypi.org/project/Wikipedia-API/`
[12] `https://hurraki.de/wiki/Hauptseite`

Topic	Standard English	Easy English	Standard German	Easy German
Art and Culture	100	100	89	50
Sports	100	100	84	50
Science and Technology	100	100	91	50
Travel	100	100	87	50
Food	100	100	81	50
Business, Economics and Politics	100	100	87	50

TABLE 3.1: Summary of the data collected across topics and languages.

3.4.3 Global Voices

Global Voices is a news platform, which offers articles across a variety of topics and languages. All articles there are under a creative commons license.

A number of articles was collected across the aforementioned topics (roughly 100 per topic). However, after the data was collected, we realised that many of the articles included text in languages other than the original language of the article. For example, this article[13] is about the recent protests in Belarus and the text includes part of the lyrics of a song in Belarusian. Removing these parts would have impaired the articles as the foreign language paragraphs are referred to in the rest of the text too. Since this applies to more than just a few articles, it is hard to use Global Voice articles in the system without major editing. The Global Voice articles, however, were used in a Topic Classification experiment described in chapter 4.

[13] `https://globalvoices.org/2020/08/18/how-one-telegram-channel-became-central-to-belarus-protests/`

Chapter 4

Evaluation and Results

In this chapter, we present the experiments and the user study that were performed as a way to evaluate the system.

4.1 User Study

An informal user study was designed to see how the proof-of-concept system was received by users.

The participants were asked to create an account and use the system in the course of three days. The users were asked to first select their corresponding German language level ("Beginner" or "Non-beginner") and select the topics they were interested in. For the duration of the study, users were expected to use the system a few times throughout each day, alternating between reading new texts and practising their vocabulary through the revision module. At the end of the study, the users were asked to reflect on their experience with the system by filling in a short survey with four categories "Use of other language learning platforms", "Appropriate content", "Ease of use" and "Overall".

4.1.1 Survey

4.1.1.1 Use of other language learning platforms

The questions in this category were aimed at finding out more about what other language learning platforms people use and what they like about them. The answers to these questions should give us a better overview of what potential future users might look for in our system.

4.1.1.2 Appropriate content

The questions in this category asked about the participants' opinion about the content that the system provided. The participants were asked:

- Whether the text difficulty was appropriate for the language level they had specified in their profile and given five options to choose from ("Too easy", "Fairly easy", "Just right", "fairly difficult" and "too difficult")
- Whether the texts they were presented with matched the topics they marked as being interested in ("Almost all were relevant", "More than half were relevant", "About half were relevant", "Less than half were relevant", "Almost none were relevant")
- Whether they found that the translations offered were correct ("Almost all were correct", "More than half were correct", "Roughly half were correct", "Less than half were correct", "Almost none were correct")
- How often the translations offered helped them understand the meaning of the text ("Almost always", "More than half of the time", "About half of the time", "Less than half of the time" or "Almost never")
- Whether they felt they had a large enough variety of texts to choose from ("Yes", "Yes, but I would have liked more" or "No").

4.1.1.3 Ease of Use

The "Ease of use" category was designed to see whether participants found the system to be intuitive to use, whether they enjoyed using it and what would have made their experience more enjoyable.

4.1.1.4 Overall

This section was designed to see whether users are likely to continue using the system as it is, whether they would be interested in using an improved version of the system and what additional features they would like to see.

4.1.2 Results

Six participants took part in the experiment. They were all Master's of PhD students or young professionals. All participants were proficient or native English speakers and spoke at least one more language fluently. Their German language level varied from complete beginner to upper-intermediate.

4.1.2.1 Use of other language learning platforms

All the users reported having used other language learning platforms, with five users having used Duolingo, four - Memrise and one user other, more grammar and exam based, applications for German.

When asked about the features they liked in other apps, three of the participants said they liked having audio materials and one pointed out the short video of native speakers provided by Memrise. Additionally, users liked quizzes (in one case time-paced) for vocabulary revision, reminders, and sample sentences showing the usage of the words.

4.1.2.2 Appropriate Content

In terms of content, most users found that the texts they were presented with were fairly easy or just right for their level and that more than half or almost all were relevant to the topics they had selected in their profiles.

Users found they had enough texts to choose from, though some would have liked to see more.

The biggest issue users had with the content were the translations, with two users finding they were correct less than half of the time, while the other four found they were correct more than half of the time or almost always (see figure 4.1). Only half of the users found that the translations helped them understand the meaning of the text more than half of the time (or almost always) (see figure 4.2). The dissatisfaction with the translations was also reflected in the comments related to the content, with four users pointing out that some words were lacking translations. One user pointed out they would like to see more granularity in terms of difficulty levels.

4.1.2.3 Ease of Use

All users found that the platform was intuitive to use or more or less intuitive to use, with four wanting to see some guidelines (such as a walkthrough or pop-ups when they

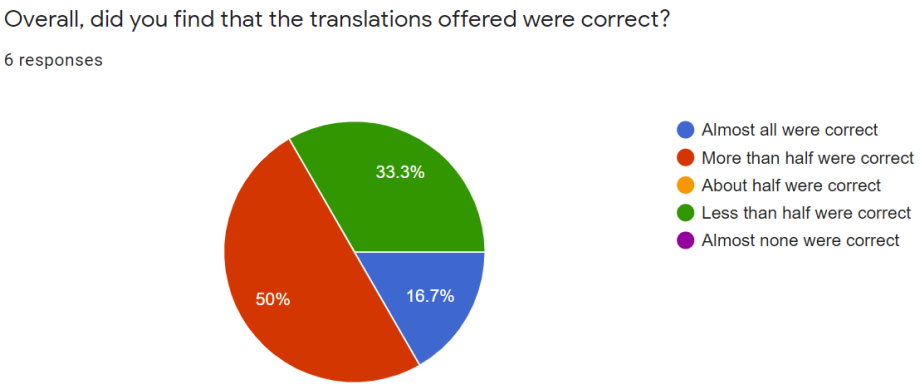

FIGURE 4.1: Responses to "Overall, did you find that the translations offered were correct?"

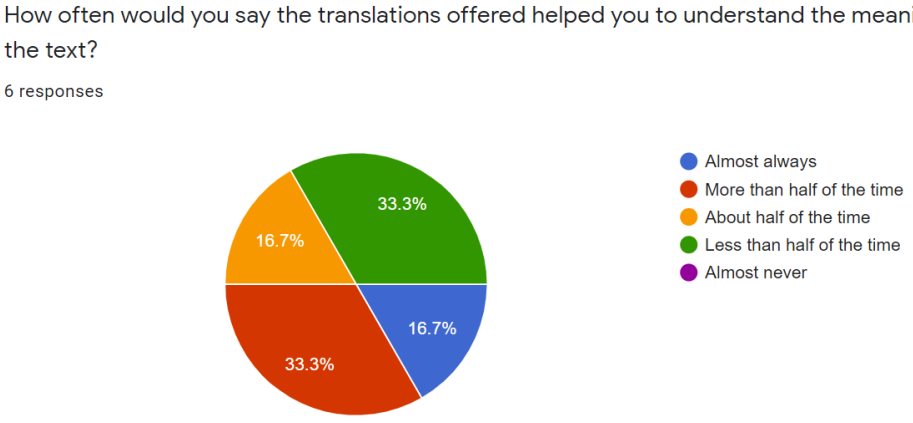

FIGURE 4.2: Responses to "How often would you say the translations offered helped you to understand the meaning of the text?"

try out new features) and two being unsure about the matter. One user enjoyed using the platform and five mostly enjoyed using it.

The current behaviour of the "Done" button on the *Read* page was named as one of the confusing aspects of the system. Faster loading, the possibility to enter notes, more revision modes, and having images displayed along with the text were among the things that would have made the users' experience more enjoyable.

4.1.2.4 Overall

The results from the last two multiple-choice question of the quiz, related to continued use of the system and continued use of the improved version of the system are shown in figure 4.3 and figure 4.4 respectively.

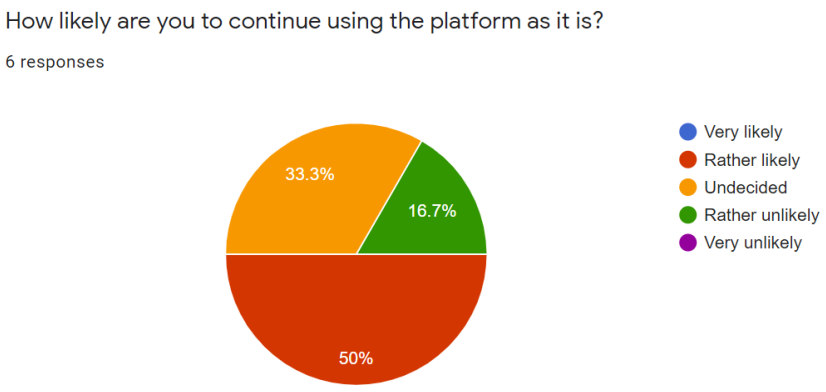

FIGURE 4.3: Responses to "How likely are you to continue using the platform as it is?"

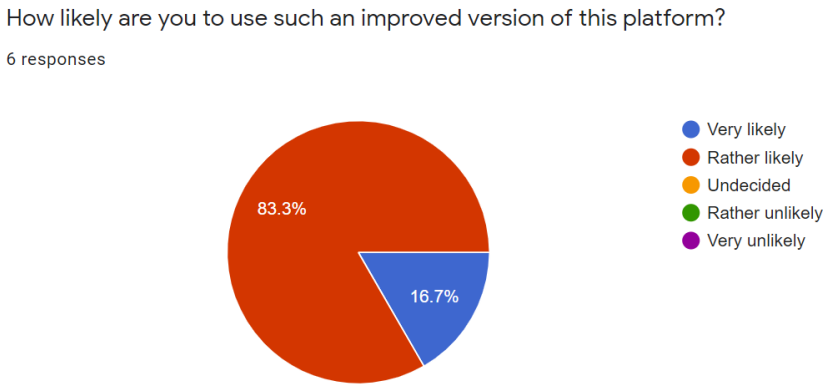

FIGURE 4.4: Responses to "How likely are you to use such an improved version of this platform?"

For the remaining graphs of the multiple-choice questions, please refer to the Appendix.

4.2 Topic Classification

While the topics of the data used in the final version of the platform were manually annotated, an experiment was carried on in the early stages of development of the system on the Standard English data and the data collected from Global Voices to see how good an algorithm trained on this data would be at predicting the topics of new articles that are also from this set of topics.

Dataset	Training Accuracy	Test Accuracy
Global Voices	0.303	0.218
Standard English	0.865	0.844
Combined	0.571	0.542

TABLE 4.1: Topic classification experiment results.

4.2.1 Experiment Setup

The expepriment was inspired by this article[1] on topic classification and was performed in order to better understand the behaviour of the data that has been collected. Following the article's setup, a TF-IDF vectorizer was used with an n-gram range = (1, 2), maximum DF = 1, minumum DF = 10 and maximum features = 300 to vectorize each text. After that an SVM classifier was used to train and test a topic classification model. The train-test split was 85:15.

The data used for this experiment was the English data collected from the Global Voices articles (578 articles balances across the six topics) and the Standard English data (600 articles across the same six topics). The algorithm was first trained and tested on the combined data and consequently on each of the Global Voices and Standard English datasets separately.

Towards the end of the development of this project, the experiment was re-run. This time, confusion matrices were generated for each of the experiments to help better understand the performance.

4.2.2 Results

The results from this experiment are summarised in table 4.1. As can be seen from the table, while the model trained on the Standard English dataset performs well, with accuracy of 0.844 on the test set, the model that was trained and tested on the Global Voices dataset performs very poorly with an accuracy of 0.218 on the test set. Thus, it is not surprising that the model trained on the combined dataset has a performance exactly in the middle between the other two.

To better understand where the problem related to the Global Voices data originated from, confusion matrices were generated at a later stage for each experiment. As it can been seen in figure 4.5, almost all articles in the Global Voices dataset were classified as "Travel". A similar pattern is observed in the combined dataset in figure 4.7, where the mostly correctly identified articles along the diagonal are likely coming from the Standard English dataset and the ones classified as travel from the Global Voices articles.

[1] `https://towardsdatascience.com/text-classification-in-python-dd95d264c802`

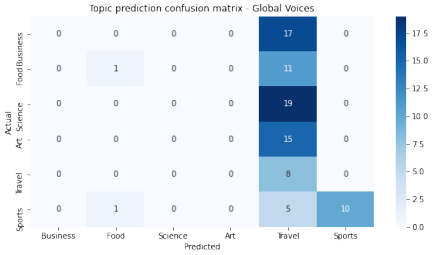

FIGURE 4.5: Confusion matrix for Global Voices dataset on topic classification

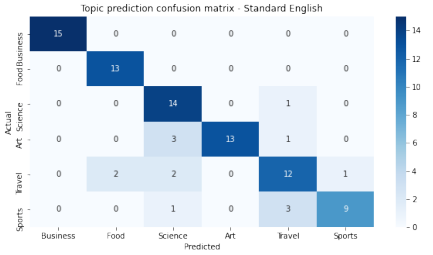

FIGURE 4.6: Confusion matrix for Standard English dataset for topic classification

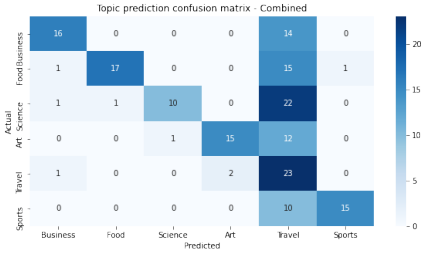

FIGURE 4.7: Confusion matrix for combined dataset for topic classification

Chapter 5

Discussion

In this chapter, the user study and experiment results are discussed. Then, we talk about future work directions for this work, a large amount of which were considered and outlines throughout the development of the system and some of which were inspired by the feedback from the user study.

5.1 Experiment results

In this section, the insights gained from the user study and the impact of the results from the topic classification experiment are discussed.

5.1.1 User Study

The feedback from the user study shows that, despite its shortcomings, the system was well-received as a prototype by the users. The aspects of the systems that the users indicated needed improvement coincide with the expectations the author had, demonstrating a realistic understanding and evaluation of the user needs and requirements.

It is interesting to note that, while the algorithms used to sort the texts shown in the *Explore* page were rather simple, users were generally satisfied with the text difficulty and topics of the texts they got at the top of the page. This poses the question of how necessary it is to invest the computational resources to provide better recommendations and whether this will bring in a reciprocal amount of value in the user experience. Additionally, this also shows that in improving the content in the future version of this system, providing better translations must necessarily have higher priority than updating the algorithms that sort the texts according to difficulty and topic.

5.1.2 Topic Classification

The first direct impact the the topic classification experiment had on this project, was the realisation that something was wrong with the Global Voices data. This led to a closer look at the data and the realisation that many of the articles in this dataset contained text different than English (which was the original language of the articles) as discussed in chapter 3. Therefore, this dataset was put aside for the purposes of the proof-of-concept system and will be revisited at a later stage.

At the moment of writing of this report, we have not been able to identify why almost all articles from the Global Voices dataset were classified at "Travel". To better understand this, it will be necessary to have a closer look at the data that was used for training as well as testing and to try out different hyperparameters for training. These points are subjects to a future work investigation.

5.2 Future Work

In the chapters above, we have outlined the current state of the proof-of-concept system developed as part of this thesis. Throughout the development of this system, a large number of directions for future work were identified that can help bring this work to a more complete web application. Along with many of the already considered future work directions, the participants in the user study also proposed a number of ideas which were not previously considered and are certainly worth exploring.

The most important future work directions are presented in this section.

5.2.1 Content

One of the first future work directions on the priority list, is to improve the content available within the system. This includes a number of aspects.

5.2.1.1 More topics

To allow for better granularity in interests and for more varied texts, it is necessary to expand the current set of topics. There are two aspects to this. Firstly, the current topics, which are somewhat broad, could be broken into finer ones. For example, "Arts and Culture" could be broken down into "Music", "Visual Arts", "Literature", etc. The current articles in the database can be split according to these new topics. Secondly,

more articles could be collected for each of these topics, allowing, again, for a larger variety. Having more articles per topic will also allow us to train better topic classification models.

An important future work direction is also to allow an article to have multiple topics. For example, if our system offers the topic "People" as well, then an article about Pablo Picasso should be tagged with both "Art" (or "Visual Arts") and "People".

5.2.1.2 More difficulty levels

As seen with the topics, providing more difficulty levels could be achieved by collecting more data. However, as experience throughout the development of this project shows, finding copyright free articles at a range of language skill levels is difficult. Therefore, a better approach to expanding the text difficulty aspect of the database could be to use text simplification algorithms. The main idea of this would be to start from a text at a C2 (or native) language level and simplify said text in a few different ways so that it is appropriate for learners at levels A1 through C1 as well.

As the simplified texts are expected to not be completely correct, some manual work by editors will be required. This will not be as fast as having all the work being done automatically, but it is a huge improvement over entirely manual content creation.

5.2.1.3 Better translations

As we have seen, the current version of the system defaults to the most frequent translation of a token when translating texts. This is a heuristic approach and while it works in a large amount of the cases (especially for non functional words), it is far from ideal. A number of statistical, Machine learning and neural algorithms need to be tested in order to find a solution which works well for this platform.

5.2.1.4 Translating phrases

In the current version of the system, the highlighting of a text is done on the token level. Another direction for future work, inspired by LingQ's approach, involves highlighting and providing translations for common phrases and collocations as well. For example, in the "Katze" text we have seen in chapter 3, the German phrase "es gibt" is present in the sentence "Es gibt auch große wilde Katzen". In this context, if we want contextual word translation, "es" will translate to "there" and "gibt" would translate to "are" or even "exist" in the English equivalent of the sentence. These translations would help

the user understand the text, but it would also be useful if along with “es” and “gibt”, the phrase “es gibt” is also highlighted and translated as “there exist”, so that the user can add the whole phrase to their vocabulary list and learn it too.

5.2.1.5 Tailored content for complete beginners

While the system seemed to work well for users with some knowledge of German or at least knowledge of another Germanic language (namely, English) in the cases where they were complete beginners, there is a risk that the system may not work well for languages that are not close to any of the languages that the user speaks. This was considered during the development of the system and was also remarked by one of the participants in the user study.

To tackle this and gain a userbase even among the complete beginners in a given language, there are two main aspects currently being considered as future work directions that can be implemented within the system. The first aspect is for the system to contain very simple texts, likely manually created, tailored specifically for complete beginners that will help them gain the necessary vocabulary. The other aspect would be to have vocabulary lists tailored for beginners which contain basic general vocabulary (e.g. “I”, “good”, “can”, etc.) and basic vocabulary for specific topics. Complete beginners can practice and learn the words in these lists before starting reading.

5.2.2 Additional functional requirements

Each of the additional functional requirement presented in 3.1 is a future word direction in itself.

5.2.2.1 FR5 - Learning from users

As mentioned earlier, provided the user base expands, we will be able to use data from the users, such as what texts they find easy or difficult, what texts they find relevant to their topics of interesting and use these to make better suggestions for new texts to users with similar profiles.

Furthermore, adaptive learning techniques can be used to adapt the content a user is presented with based on their performance. That is, based on the amount of unknown words that a user marks in a text, that text can be deemed too easy or too difficult for the user, so that next time they can be presented with a text more appropriate for

their level. This is important also because, as users progress by using the system, their language level should increase and therefore defer from the level they initially chose.

Finally, as mentioned in 2.2, the difficulty of a text for a specific user is linked to the closeness of the language they are learning and the language(s) they already know. Having data on the languages each user has knowledge of and their respective levels, may be another useful set of features in determining the profile similarities between users and therefore offering more appropriate texts in terms of difficulty.

5.2.2.2 FR6 - Users can edit vocabulary

This feature was considered in the initial stages of development and was mentioned in the feedback by one of the test users. The proposed translations are not always correct now (and some words are even missing translations) due to the heuristic method used for that. While this is due to improve in a future version of the system, automated translation will never be perfect. Therefore, it is essential to allow users to suggest edits. A future improvement for the system would be to allow them to edit their own vocabulary and propose different translations (that will also be available to other users) where they find that the translations offered by the system are incorrect. Apart from making sure other users get the correct translations, this approach would also help the system learn some alternative translation and improve its own translation algorithm over time.

5.2.2.3 FR7 - Users can add texts

Another feature that has been taken into consideration is to allow users add their own texts which can then be processed by the system and read by the user. The potential development of this feature, however, needs to be taken into more careful consideration. The reason for this is that copyright restrictions need to be taken into account as the texts that users add may not be their own. Thus, while this feature remains on the Additional FR list, its priority is not as high as it was at the start of the development of this project.

5.2.2.4 FR8 - Pre-test

As considered in the initial development stages of the project and as suggested in some of the user feedback, determining the user's language level is another future work direction. When a user starts learning a new language on the platform, they will be offered to do

a pre-test for that language. The structure currently considered for a pre-test is to offer progressively harder texts to the user to read and ask them to mark the unknown words, until the correct language level is determined. The pre-test ties well with **FR5**, because having data on how other users perform on different texts, will help determine the level of a new user better.

5.2.3 More revision modes

The system currently offers one revision mode - quizzes. A wider range of revision modes has been considered as a future work direction for this work. The results from the user study only show that this will indeed be helpful, since a few of the test users pointed out that this is one of the aspects they want to see in a future version of the system.

The revision modes that are considered for the future expansion include:

- Typing tests - showing the translation or original word to the user and asking them to type the original word or the translation accordingly;
- "Fill in the gap" exercises - showing a sentence to a user where one of the words in their `Learning` list is missing and ask them to fill in the sentence. This aspect is interesting in terms of NLP work as well, as creating such exercises automatically will involve using some text generation approaches.
- Passive revision - this is where users will be shown texts containing some of the words they have recently learnt as a refresher exercise. It is important to note that this is different from text comprehension exercises (where learners are asked to read a text and then answer questions about it), which are used in Intensive Reading rather than Extensive Reading.

5.2.4 Audio

As pointed out by a few of the participants in the user study, one of the aspects they found most helpful in other language learning systems was the audio accompanying the words or sentences shown in these systems.

This makes sense as it helps users learn the correct pronunciation, which is especially important for languages with highly irregular orthography or languages which are not alphabetic, but rather syllabic or logographic.

Therefore, adding a text-to-speech component to the system is likely to prove popular and helpful among users. The first aspect of this will be to add single word pronunciations, for example, when a user clicks on a word in the text page, and in revision mode. Once this first step is complete, it might be useful for readers to be able to hear whole sentences or even texts read out loud and thus a module to address that will be investigated.

5.2.5 Gamification

Gamification has been used increasingly as a method in increase motivation to use a variety of educational platforms. While some show positive effects from using gamification [17] and others call for more empirical evidence[18], it cannot be denied that gamification is used by the majority of the existing language learning platforms and it is worth investigating whether this platform can also benefit from it.

Gamification aspects can be implemented in a number of different ways, from points and badges systems, through leaderboards where users can compare (and compete) against each other, to quests which ask users to complete certain tasks (e.g. revise a certain number of words or read at least 100 words) to unlock a new text, for example.

Each of these gamification strategies could be applied in our system, thus further investigation into what would work best is necessary and is therefore another direction for future work.

Chapter 6

Conclusion

In this thesis we looked at some existing language learning systems and they features the offer and proposed a new solution, which combines Extensive Reading and Spaced Repetition and proposes a way to automatically collect and process the content for a language learning platform.

The proposed solution was implemented as a proof-of-concept web application for English speaking students learning German. The resulting platform was tested in a user study by six users, whose feedback showed that the platform was received positively.

In the course of the development of this system, about 2,000 articles across two languages, two difficulty levels and six topics were collected and annotated with their topic and difficulty. Seeing that this dataset could be useful for other researchers working on text difficulty, topic classification, or even summarisation tasks, the dataset was made publicly available.

Finally, a number of future work directions were proposed, taking into account the user feedback and the lessons learnt throughout the development, which aim to bring the proof-of-concept system presented in this thesis to a complete application for language learning.

List of Figures

List of Tables

Bibliography

[1] Richard R Day, Julian Bamford, Willy A Renandya, George M Jacobs, and Vivienne Wai-Sze Yu. Extensive reading in the second language classroom. *RELC Journal*, 29(2):187–191, 1998.

[2] Steve Kaufmann. *The linguist: A personal guide to language learning*. The Linguist, 2003.

[3] Beniko Mason and Stephen Krashen. Extensive reading in english as a foreign language. *System*, 25(1):91–102, 1997.

[4] Timothy Bell. Extensive reading: Speed and comprehension. *The reading matrix*, 1(1), 2001.

[5] Richard C Atkinson. Optimizing the learning of a second-language vocabulary. *Journal of experimental psychology*, 96(1):124, 1972.

[6] Paul Pimsleur. A memory schedule. *The Modern Language Journal*, 51(2):73–75, 1967.

[7] Sebastian Leitner. *So lernt man lernen: angewandte Lernpsychologie–ein Weg zum Erfolg*. Herder., 1995.

[8] Burr Settles and Brendan Meeder. A trainable spaced repetition model for language learning. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 1848–1858, 2016.

[9] Hermann Ebbinghaus. Memory: A contribution to experimental psychology. *Annals of neurosciences*, 20(4):155, 2013.

[10] Philip I Pavlik and John R Anderson. Using a model to compute the optimal schedule of practice. *Journal of Experimental Psychology: Applied*, 14(2):101, 2008.

[11] Harold Pashler, Nicholas Cepeda, Robert V Lindsey, Ed Vul, and Michael C Mozer. Predicting the optimal spacing of study: A multiscale context model of memory. In *Advances in neural information processing systems*, pages 1321–1329, 2009.

[12] Eleni Metheniti, Pomi Park, Kristina Kolesova, and Günter Neumann. Identifying grammar rules for language education with dependency parsing in german. In *Proceedings of the Fifth International Conference on Dependency Linguistics (Depling, SyntaxFest 2019)*, pages 100–111, 2019.

[13] Lisa Beinborn, Torsten Zesch, and Iryna Gurevych. Readability for foreign language learning: The importance of cognates. *ITL-International Journal of Applied Linguistics*, 165(2):136–162, 2014.

[14] Thomas Hofmann. Probabilistic latent semantic analysis. *arXiv preprint arXiv:1301.6705*, 2013.

[15] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[16] Alexander Soemer and Ulrich Schiefele. Text difficulty, topic interest, and mind wandering during reading. *Learning and Instruction*, 61:12–22, 2019.

[17] Cigdem Hursen and Cizem Bas. Use of gamification applications in science education. *International Journal of Emerging Technologies in Learning (iJET)*, 14(01): 4–23, 2019.

[18] Jorge Francisco Figueroa Flores. Using gamification to enhance second language learning. *Digital Education Review*, (27):32–54, 2015.

Appendix A

User study results

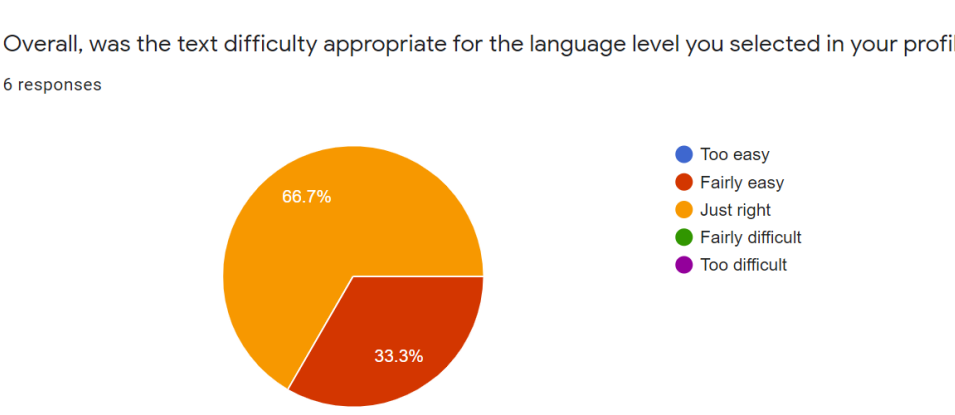

Figure A.1: Responses to "Overall, was the text difficulty appropriate for the language level you selected in your profile?"

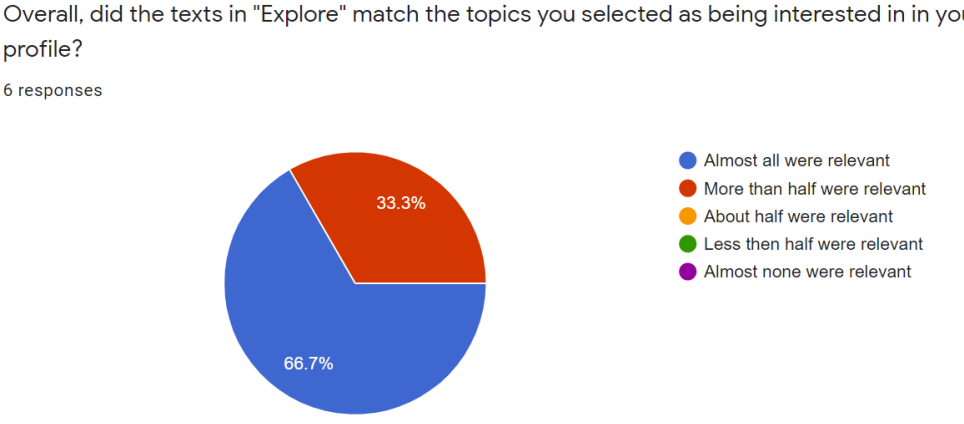

Figure A.2: Responses to "Overall, did the texts in "Explore" match the topics you selected as being interested in in your profile?"

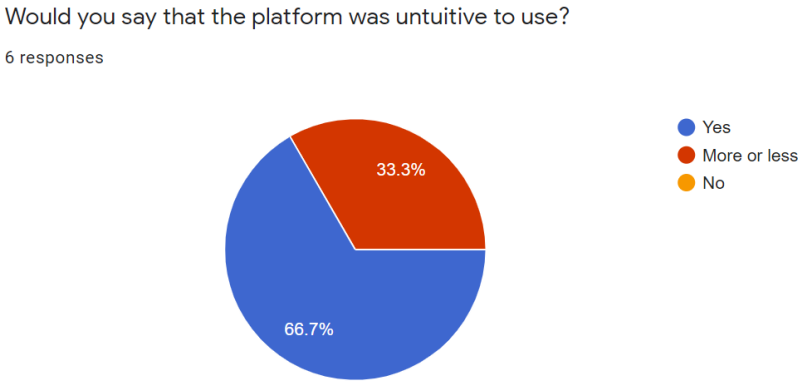

FIGURE A.3: Responses to "Would you say that the platform was intuitive to use?"

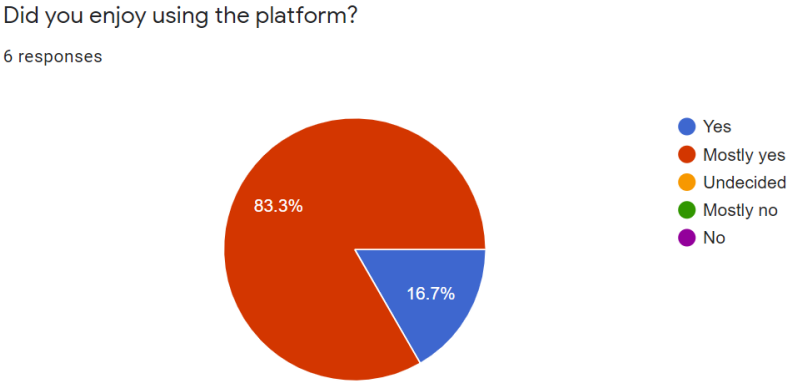

FIGURE A.4: Responses to "Did you enjoy using the platform?"

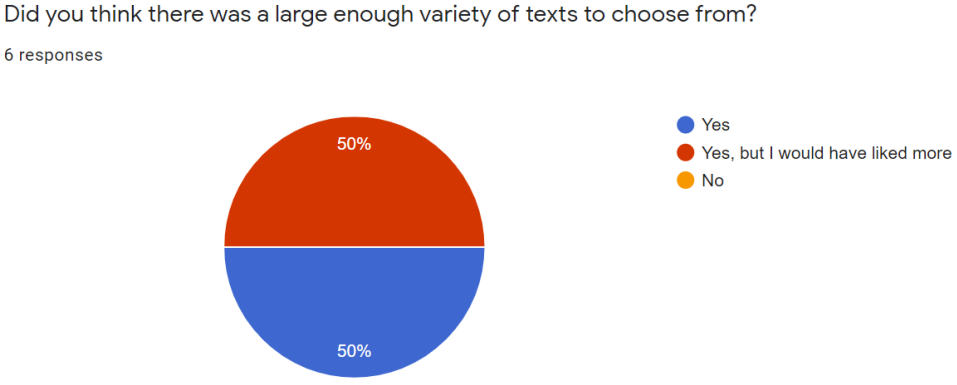

FIGURE A.5: Responses to "Did you think there was a large enough variety of texts to choose from?"

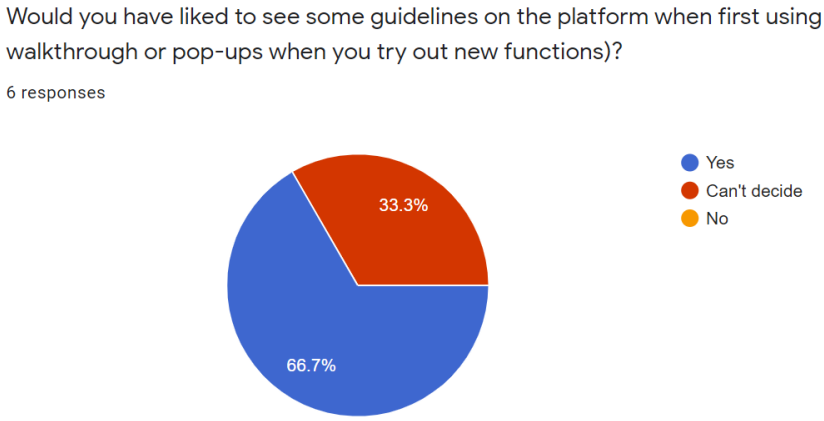

FIGURE A.6: Responses to "Would you have liked to see some guidelines on the platform when first using it (e.g. a walkthrough or pop-ups when you try out new functions)?"