

Gamified Language Learning: Enhancing Learning through Streaming Services and SRS

J. Emanuel Cascone R. S.¹, Guilherme A. Avelino¹

¹Computer Science Department - Federal University of Piauí (UFPI)
Teresina, PI – Brazil

{emanuelcascone, gaa}@ufpi.com.br

Abstract. *The ability to learn a new language is directly related to the user's disposition and comfort for better interaction with learning platforms. This facilitates the gamification of language learning, thereby reducing its level of exhaustion. The process of learning a new language typically commences with activities such as writing, reading, listening, and speaking. This work involved creating an application that facilitates language learning by utilizing a movie streaming platform with subtitles, allowing users to watch movies with the original audio and subtitles and focus on keywords or frequently repeated words for better comprehension. The work is based on the premise that to understand a text, you don't necessarily need to know the whole formation but rather just the keywords or most repeated words, allowing you to apply skimming and scanning techniques. To record these words in the user's memory, gamified forms of Spaced Repetition System (SRS) can be applied to induce the user to better adapt to each scenario in the future based on their current repertoire. By following this approach, users can enhance their experience while watching movies or videos to acquire knowledge of a new language.*

Key words: *language learning, gamification, streaming platforms, subtitles, SRS, skimming, scanning*

1. Introduction

Mastering a new language is undeniably valuable, yet it's often accompanied by challenges such as waning motivation and a sense of discomfort. What if there was a way to transform language acquisition into an engaging and less taxing experience? Imagine users effortlessly learning while enjoying themselves.

In parallel, watching movies and series has become a widespread and enjoyable activity, with streaming platforms emerging as a popular source of entertainment. This trend is evident from the data depicted in Figure-1 and Figure-2, illustrating the growth of Disney+ and Netflix over the years. Currently, there are over 700 million users across all platforms, as shown in Figure-3. Many individuals utilize these platforms to enhance their language skills by utilizing subtitles and audio for learning purposes.

Figure 1. Netflix subscribes from 2011 until 2023. font: company data

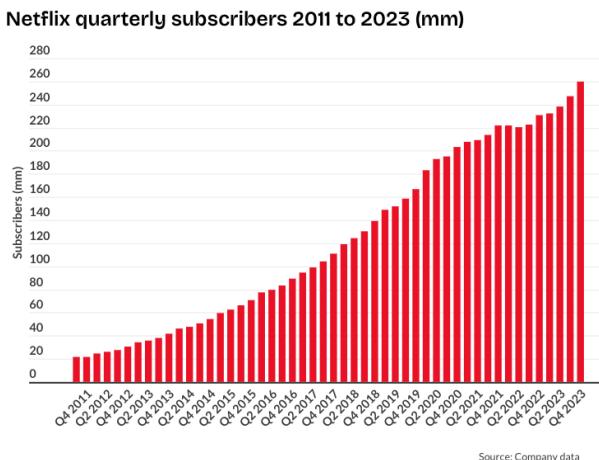
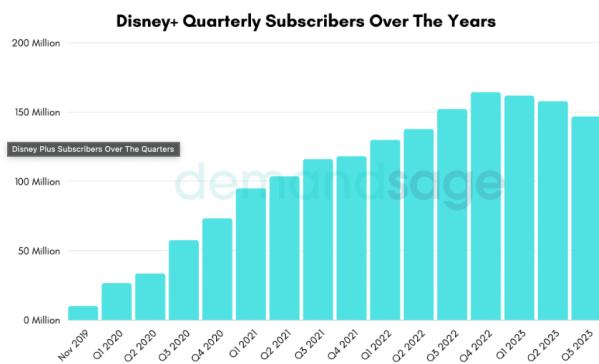


Figure 2. Disney subscribes from 2011 until 2023. font: company data



Studies demonstrate that students acquire a new language at double the speed using a similar learning mechanism called smart subtitles (?). Thus, it is feasible to infer that approaches involving entertainment platforms can also be practical for learning, especially when utilizing subtitles from these platforms.

The fusion of these concepts has the potential to help language learning, making it more

Figure 3. Table with lists of subscribers. Font: Forbes, access: 02.2023

Streaming Services with the Most Subscribers	
Streaming Platform	Users (in millions)
1 Netflix	247.2
2 Amazon Prime Video	200.0
3 Disney+	150.2
4 Max	95.1
5 Paramount+	63.4
6 Hulu	48.5
7 Peacock	28.0
8 ESPN+	26.0
9 AppleTV	25.0
10 Starz	15.8

Source: [Forbes Home](#) • [Embed](#)

Forbes HOME

engaging and less taxing. This work is about the creation of an application that allows users to learn a new language or expand their vocabulary in a gamified and interactive manner using streaming platforms.

1.1. Motivation

The motivation arises from the need to find alternatives to make learning a new language more exciting and efficient. The conventional language learning method can be tiresome and demotivating for many students.

By creating an environment that allows interaction with streaming sites for movies, learning a new language can be done more naturally and playfully, similar to tools that translate web pages to facilitate learning or understanding of other languages already present in the market or research area (?).

By watching movies with audio and subtitles in the original language, the user can learn while applying gamified Spaced Repetition System (SRS) techniques to memorize new words. In succinct terms, the motivation for this work is to provide an efficient and attractive alternative for learning new languages.

1.2. Problem Statement

The traditional method of learning a new language can be exhausting and demotivating for many students. A more playful and interactive approach to learning new languages is necessary to make the process more enjoyable and efficient. To illustrate this, we notice

that the current learning method creates a massive flow of giving up; an empirical example is "only 55, or 19% of 286 Year 11 students reported that they intended to continue studying the language after age 16. The four most frequently cited reasons for not continuing were (a) that they did not enjoy it (35 students), (b) it was difficult (25 students), (c) it was of no use for their planned career (23 students), and (d) they were not good at it (20 students)" (?).

1.3. General goals

This work aims to develop an application that allows users to learn a new language or expand their vocabulary in a gamified and interactive manner. The extension will be designed to interact with movie streaming sites, leveraging the available subtitles on these sites to aid in the teaching process.

The work is based on the premise that learning a new language more efficiently and satisfactorily is possible when gamified and interactive. Additionally, it is based on a similar methodology - smart subtitles - which has been effective in expanding vocabulary (?).

Thus, the goal is to offer a different and efficient approach to using subtitles so that people can learn new languages and expand their vocabulary while watching their favorite movies in a relaxed and effective way.

1.4. Specific Objectives

The work aims to achieve the following specific objectives:

1. Develop browser extensions, a website, and an application accessible to users interested in language learning.
2. Implement functionality in the browser extensions, website, and application to interact seamlessly with streaming platforms for accessing subtitles.
3. Establish and maintain a comprehensive database of keywords for each target language, alongside user progression tracking.
4. Provide translation services for keywords, aiding users in comprehension.
5. Utilize gamified Spaced Repetition System (SRS) techniques to facilitate the memorization of new vocabulary.
6. Enable users to select their preferred language for learning.
7. Offer detailed progress statistics to users, including metrics such as the most challenging words and total words learned.
8. Support multiple languages through AI integration, providing a diverse learning experience.

2. Methodology

To achieve the work's objectives, in this article we followed a methodology based on the following steps: The first step was a literature review and gathering references for similar works or theoretical foundations to turn language learning into a more playful and interactive experience.

The second step was the study of methods for user training to improve the user's progress and how to manage the user information, including their progression in learning to make the learning process more efficient.

The third step was to build a prototype based on the previous steps, to test the idea and the user experience.

2.1. First Step

The literature review identified (?) and (?) regarding subtitles in movies and their role in language learning. However, it is evident from apps like Flex and others that utilize this methodology that users do not receive training tailored to their individual word difficulty levels.

Apps such as Duolingo, Memrise, Busuu, Lingoda, and Drops employ algorithms that adapt to the specific difficulties of each user. While the exact algorithms are not disclosed, users can observe word repetition in different contexts during the training process. This technique, known as Spaced Repetition System (SRS), facilitates user study. Moreover, gamification is integral to these apps, enhancing language learning and potentially improving both user experience and the learning process.

Current methods utilizing subtitles often result in recurrent breaks, as users must pause the movie to view translations by clicking on words or may need to read dual subtitles simultaneously. Another technique involves translating only the difficult parts of a phrase (smart subtitles), but this lacks interactivity and playfulness.

2.2. Second Step

The second step focused on understanding the functionalities of language learning apps like Duolingo and others, and how to enhance both user experience and the learning process.

In 2024, it is notable that these apps leverage gamification to make learning less taxing. Games form the core of the learning process, compelling users to engage actively. For instance, Duolingo incorporates skimming and scanning techniques, presenting phrases and words in various contexts either to teach new words or to facilitate understanding without requiring knowledge of every word.

Another significant app is Anki, which employs SRS to improve both user experience and learning outcomes.

Furthermore, this step involved understanding user data and its utilization for enhancing the learning process. It was identified that tracking when a user encounters a word, whether they learned it, and its significance to the user can significantly improve learning efficiency and user experience.

2.3. Third Step

This step aimed to integrate the four identified cores: SRS, gamification, scanning, and skimming, with the addition of "smart" functionality to subtitles. Users should be trained based on their individual difficulties, without interrupting the enjoyment of watching a video or movie with constant breaks for word translations.

The proposed solution involves creating an application that teaches users before watching the movie, eliminating the need for interruptions. This application should be interactive and playful, incorporating games to enhance both user experience and learning outcomes while employing SRS techniques.

To implement this, interaction with streaming sites is necessary to capture subtitles and provide word translations to users seamlessly. There are two feasible approaches: wrapping the site or using a browser extension. However, due to streaming sites' use of technologies to block requests from different origins, only the browser extension approach is viable.

In addition to the extension, creating other environments such as a website and a mobile app to enhance user experience and learning outcomes is also important to provide an entire ecosystem for the user.

2.3.1. Key Requirements

To meet the work's objectives, a comprehensive list of activities was undertaken. The work's development was segmented into various stages, each following a logical progression towards finalizing the requirement list. Different steps were identified for each environment.

2.3.2. List of Requirements

Based on the work's objectives and the methodology used, we identified the following requirements for creating a prototype:

- Research and study of documentation on browser extensions, sites, and apps.
- Study of possibilities for user training and important data for capture and storage.
- Start of architecture definition and choice of technology to be used.
- Start of creation of the extension's graphical interface.
- Conduct a literature review and gather references for similar works or theoretical foundations.
- Development of logic for capturing and displaying subtitles from streaming sites.
- Data analysis and study of possibilities for user training and important data for capture and storage.
- Testing and improvement of subtitle capture.
- Development of teaching logic based on captured subtitles.
- Testing and improvement of teaching logic.
- Integration of subtitle capture and teaching logic.
- Final testing of the extension.
- Review and correction of any issues identified in final testing.
- Preparation of technical documentation for the script
- Finalized script and teaching logic.
- Creation of the site's graphical interface.
- Creation of the app.
- Graphic adaptation.
- Refactoring of the application and initiation of mini-games creation.
- Technical adjustments.
- Bug fixing and improvements.
- Final testing and delivery of the prototype.
- MVP (Minimum Viable Product) delivery.

2.3.3. Approach to Goal Attainment

The development process for the extension, site, and app involves distinct steps to achieve the overarching goal. While some steps are common across all environments, each has its

unique set of tasks to accomplish the goal.

The development of the extension was based on the following requisites:

1. Identification of streaming sites with available subtitles for movies.
2. Development of a subtitle extraction tool.
3. Integration of the extension with streaming sites.
4. Create a script to show iframe with the application site.
5. Development of teaching logic based on captured subtitles.
6. Implementation of SRS technique to gamify learning new languages.
7. Create ways to communicate between the extension, site, streaming sites, and app.
8. Integrate the extension information with the site and app.

The site:

1. Development of the site's graphical interface.
2. Development of the site's teaching logic based on captured subtitles.
3. Implementation of SRS technique to gamify learning new languages.
4. Create ways to communicate between the site, extension, streaming sites, and app.
5. Integrate the site information with the extension and app.
6. Create a way to update the site without user interaction.
7. Create a way to store user information, progression, keywords, and others.

The app:

1. Development of the app's graphical interface.
2. Show the iframe with the application site.
3. Create a way to communicate between the site and the app.
4. Create a way to store site information.

2.3.4. Difficulties and Challenges

The main difficulties and challenges encountered during the application development were related to integrating the extension with streaming sites and developing the teaching logic based on captured subtitles. The extension's integration with streaming sites required using the browser's internal messaging services, a new and complex feature to work with. Developing the teaching logic was also challenging, as it needed to use Spaced Repetition System (SRS) techniques to gamify learning new languages. This requires a deep understanding of SRS techniques, which data was necessary to store, and how to store it.

The secondary difficulties were related to the development of the site and app, as it required a deep understanding of and applying those technologies to the work, like the iframe, web channels, and others.

Lastly, it is hard to create an excellent graphical interface that is responsive and fast, so that the user can understand and use it without any problems. The work's current focus is to make it easier to operate, but it lacks some visually appealing interfaces.

3. Theoretical Referencial

In recent years, there has been a growing interest in the use of subtitled videos as a way to assist in the learning of foreign languages. However, standard subtitles are not ideal for vocabulary acquisition, as translations are non-literal and are provided only for phrases, not individual words.

To overcome this problem, "Smart Subtitles" have been developed, interactive subtitles focused on vocabulary acquisition. "Smart Subtitles" includes features such as vocabulary definitions when hovering over with the mouse cursor and video navigation based on dialogue (?). Studies have shown that students learn more than double the vocabulary with "Smart Subtitles" compared to Chinese-English dual subtitles, with similar levels of comprehension and satisfaction (?).

Those dual subtitles are a common method for learning a new language, the user see two subtitles, one in the original language and another in the language that the user is learning, and the user needs to read both to understand the movie.

Another approach to assist in the learning of foreign languages is the use of browser extensions. The browser extension "InteractiveSubtitle" (?) is an example of this. The extension features personalized learning, translation of phrases in context on web pages, and support for over 100 languages. The goal is to utilize various hypotheses of foreign language acquisition to create a new language learning tool that is more effective.

Some students of languages doesn't adopt this method, but adopt methods as SRS, that is a technique tested and proved the effectiveness by Landauer and Bjork in 1978, that involves reviewing information at increasing intervals. This technique is used in apps like Anki, Duolingo, and others, and is a technique that is used to improve the user experience and learning outcomes.

Based on these ideas, an application has been proposed that utilizes a real kind of "Smart Subtitles" to train users through gamified practice before watching a subtitled movie in a language they are learning. The application also aims to include features such as interactive exercises to reinforce vocabulary acquisition and games to help users.

Also, to improve the user commodity, an app and a site were proposed to not necessarily use a computer to study, providing a way for the user to study on every device.

3.1. Subtitles

Smart subtitles and dual subtitles have emerged as innovative solutions to enhance the multilingual viewing experience for audiences worldwide. These technologies leverage advancements in natural language processing and multimedia to provide seamless translation and accessibility options for viewers.

Dual subtitles, offer a unique viewing experience by presenting multiple languages simultaneously on-screen. This method is particularly beneficial for language learners and bilingual audiences seeking to improve their language proficiency. The concept of dual subtitles has been explored by scholars such as Dr. François Yvon, who has investigated its effectiveness in language acquisition and comprehension.

Smart subtitle technologies are integrated into streaming platforms and media players, allowing users to enable translation features with a simple toggle. Companies like fleex and Language Reactor have invested in smart subtitle solutions to cater to their diverse global audience. The implementation of dual subtitles often involves specialized software tools or custom video players capable of displaying multiple subtitle tracks simultaneously.

Studies conducted by researchers such as Dr. Spela Vintar and Dr. Koen van Turnhout have demonstrated the effectiveness of smart subtitles and dual subtitles in improving language learning outcomes and comprehension rates. Their research highlights the potential of these technologies to bridge language barriers and promote cross-cultural understanding in multimedia contexts.

Smart subtitles and dual subtitles represent significant advancements in the field of multimedia translation and accessibility. By leveraging cutting-edge technologies and insights from scholars like Alex Waibel, François Yvon, Spela Vintar, and Koen van Turnhout, these methods offer new possibilities for enhancing the multilingual viewing experience and promoting linguistic diversity in digital media.

3.2. Spaced Repetition System (SRS)

Spaced Repetition Systems (SRS) have revolutionized the way we learn and retain information. The concept of spaced repetition dates back to the 19th century, but it gained significant attention and refinement in the digital era. This method leverages the psychological spacing effect, where information is better retained when revisited at increasing intervals over time.

The foundation of SRS lies in the *spacing effect*, a phenomenon observed in cognitive psychology. Hermann Ebbinghaus, a German psychologist, first documented this effect in his studies on memory in the late 19th century. His research laid the groundwork for understanding how repetition and timing impact memory retention.

One of the most influential implementations of SRS is *SuperMemo*, developed by Dr. Piotr Wozniak in the late 1980s. SuperMemo was one of the first computer-based systems designed to optimize the learning process through spaced repetition algorithms. Wozniak's work has inspired numerous SRS applications and research studies worldwide.

Research conducted by cognitive psychologists such as Dr. Sebastian Leitner further validates the effectiveness of spaced repetition. Leitner introduced the *Leitner system*, a practical implementation of spaced repetition using flashcards arranged in different boxes based on recall success. His work demonstrates how SRS can enhance long-term retention compared to traditional learning methods.

Spaced Repetition Systems offer a promising approach to optimize learning and memory retention. From pioneers like Hermann Ebbinghaus to modern innovators like Piotr Wozniak and Sebastian Leitner, the study of SRS continues to evolve, shaping the future of education and cognitive science.

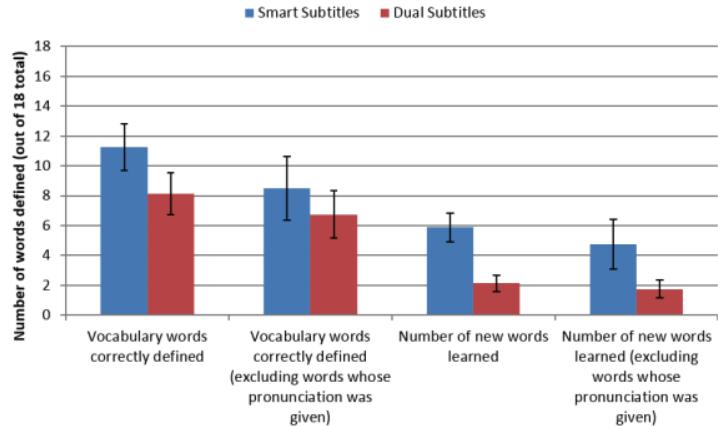
4. Software Design

The application core (the method of teaching) is based on the articles by Kovács et al. (?), which use the premise of subtitles to advocate for a more playful and immersive learning method. This demonstrates the potential of such approaches for language learning and how more playful and intelligent approaches directly influence learning new words, as exemplified in Figure-4.

The work utilizes techniques used by El Batanony et al. in their article "A browser extension to facilitate language acquisition" (?), making the beginning of the development a case study of their article, observing the approach they used to create an extension that interacts with web pages.

Using Spaced Repetition System (SRS) techniques to gamify learning new languages

Figure 4. Learning curves for new words using two types of subtitles and smart subtitles



based on the premise that knowledge of keywords or the most repeated ones allows for text comprehension.

Created using modern technologies, including JavaScript, Typescript, HTML, CSS, Prisma, Nextjs, PostgreSQL, and Web Channels. This work also utilizes the browser's internal messaging services to integrate the extension with streaming sites. The extension works with Prime Video, but the goal is to expand to other platforms like Netflix, Disney+, and others.

The technologies were chosen based on the need to create a fast and efficient application with a good user experience and an excellent database to store user information and progression. Nextjs is a frontend and backend that has a big community and is very fast, having a good integration with Prisma, an ORM (Object-Relational Mapping) that is easy to use and has a good amount of users with PostgreSQL database.

Another important point to consider use PostgreSQL is the existence of sites like Heroku, that provides a free database with a good amount of data and a good speed, and the existence of a good amount of users and a good amount of documentation.

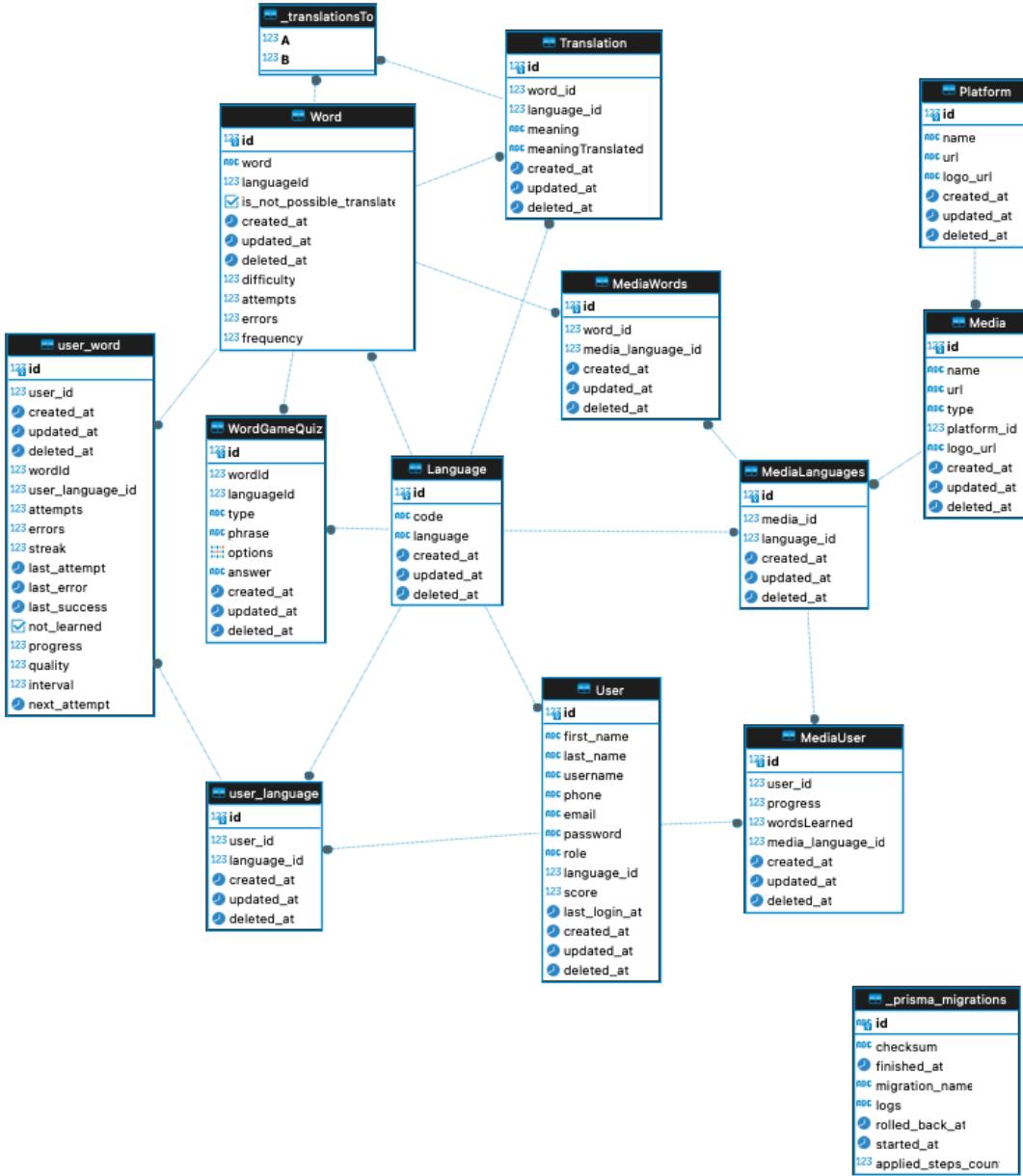
At this point in the prototype creation, an environment to learn a new language was built, including a browser extension, a site, and an app, all integrated and working together to provide a better experience.

The diagram provided in Figure 5 illustrates the intricate structure of the database employed in this work. It meticulously outlines the relationships between various tables, providing a comprehensive overview of how user data and language information are organized and interconnected within the system.

The Container diagram, depicted in Figure-6, offers a detailed visualization of the application's architecture, delineating the interactions between different components such as the extension, site, and app. This diagram showcases the technological framework utilized and elucidates how these components collaborate to facilitate seamless communication and functionality.

The figure-7 is a context diagram that was built to create this environment. Where we could see the core of the system using an AI provided by openAI, to generate translations, meanings, and games. Each one of the environments has its own functions and respon-

Figure 5. Database Diagram for the application

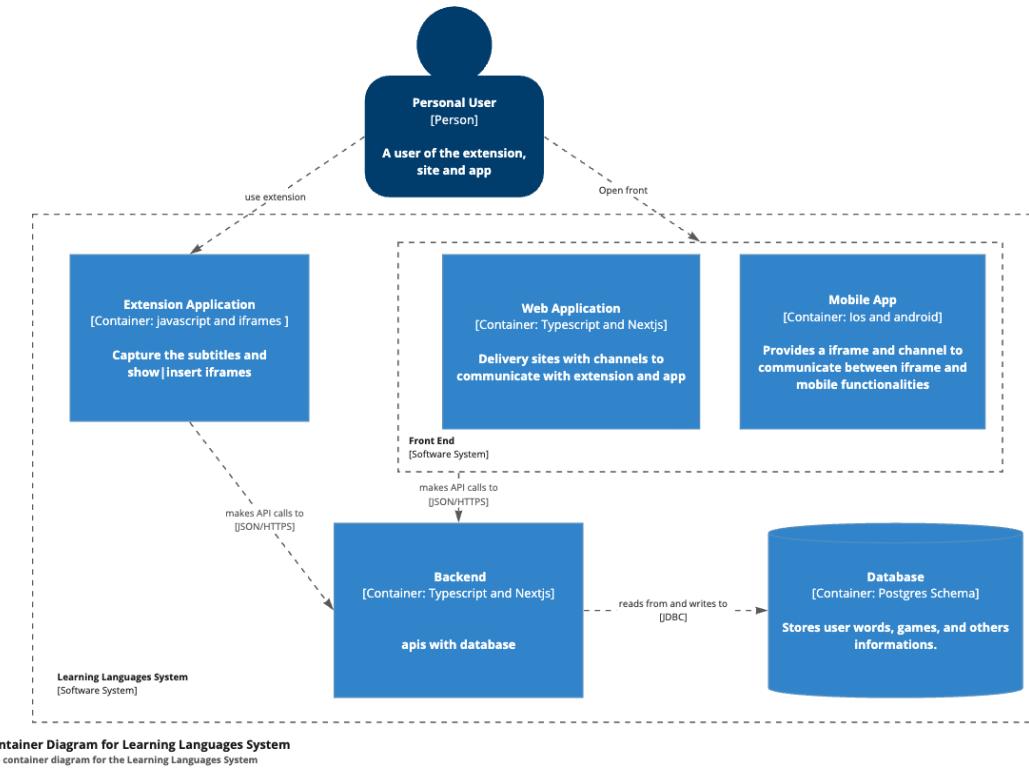


sibilities. The extension is responsible for capturing the subtitles and opening the modal with the iframe, the site is responsible for storing the data and providing the games, and the app is responsible for showing the site inside the iframe and sending the user information to the site.

To be more visible the process of the games and word data acquisition, the figure-8 illustrates it. This flow was selected with a focus on optimizing speed due to the substantial volume of data involved, along with the associated costs of utilizing AI.

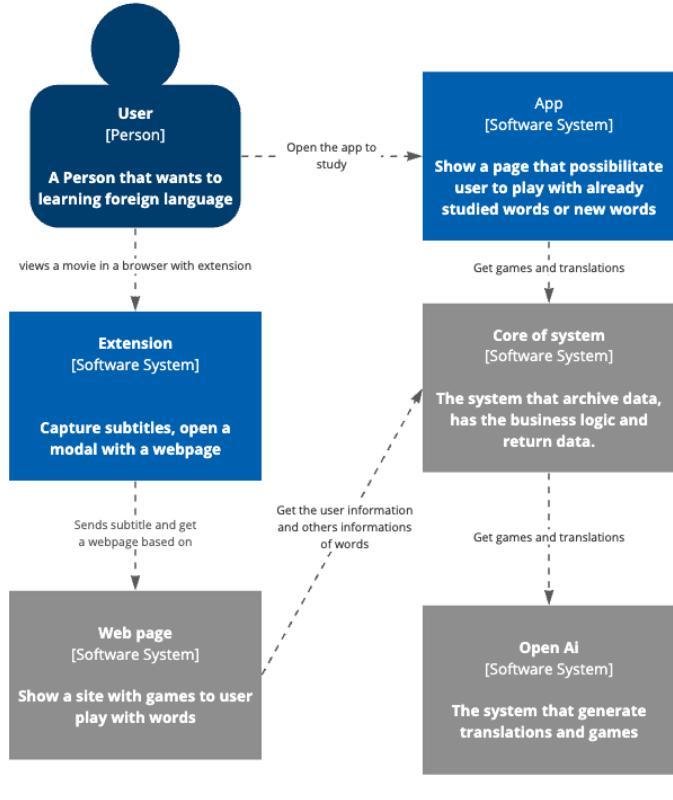
Presently, the AI of choice is the gpt-3.5-turbo-1106 model developed by OpenAI, incurring costs of \$0.0010 per 1K tokens of input and \$0.0020 per 1K tokens of output.

Figure 6. Container Diagram for the application



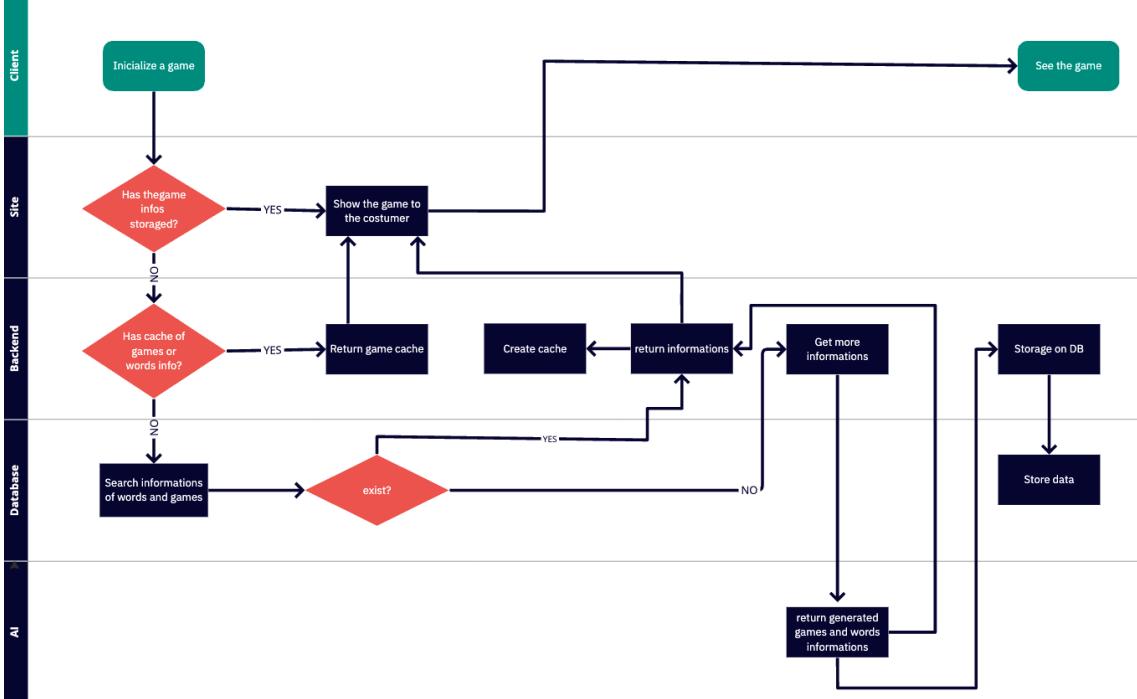
Leveraging a database for storing data offers greater flexibility, particularly since the input remains constant. This approach ensures that when a user learns a word in a specific language and transitions to another, the necessary information about games and the word parameter are retrieved from the AI only once.

Figure 7. Context Diagram for the application



System Context Diagram for Learning Languages System
The system context diagram for Learning Languages System
Last modified: 2024-01-30

Figure 8. The flow of the application until the use of AI



4.1. Extract algorithm

The filter algorithm is a critical component of the application, as it is responsible for identifying and extracting words from subtitles. For each platform (e.g., Prime Video, Netflix, Disney+), the algorithm must be tailored to the specific structure of the subtitles. The algorithm is inserted into the extension and is responsible for capturing the subtitles and sending them to the site.

But the result is the same for all platforms, a json structure with the following content:

```
{
  words: {
    word: string;
    count: number;
    moments: {
      begin: string;
      end: string;
      text: string;
    } [],
    position: number;
    percentage: number;
    total: number;
  },
  jsonFromTTML: {
    lang: string,
    version: string,
    subtitles: {
      moment: {
        begin: string,
        end: string,
      },
      text: string
      word: string
    } []
  }
}
```

The algorithm is based on the following steps:

1. The algorithm receives the subtitle in TTML (or other) format.
2. The algorithm converts the subtitle to JSON format.
3. The algorithm processes the JSON to extract the words and their respective counts.
4. The algorithm processes the JSON to extract the phrases where the word exist.
5. The algorithm processes the JSON to determine the position and percentage (relative to how often the word appears in the subtitle) of each word.
6. The algorithm sends this data to a web channel where the site receives it.

The algorithm is designed to be asynchronous, without damaging the user experience, as it must process a large volume of data. The algorithm is also designed to be flexible, as it must be tailored to the specific structure of the subtitles for each platform.

4.2. AI

The AI is a critical component of the application, as it is responsible for providing translations, meanings, and games. The AI receive a prompt based on the user's request, and the AI returns the requested information.

A example of the prompt is:

```
I am a bot that generate quiz for you to learn new words.  
My quiz is not very good at it, but I am learning.  
Help me to learn by generating a quiz.
```

The quiz are a phrase using a word in the language and you need to translate to the target language.

The options are a list of words to choose the correct translation for the word in the target language.

And the options are made by others translations on the target language or translation of similar words.

Based on the following JSON structure:

```
{  
    words: string[], // the words in that language  
    language: string, // the language  
    targetLanguage: string // the language target to  
    translate  
}  
you need to answer in JSON format, following that structure:  
{  
    [word]:{ // the word in the language  
        quiz: string, // the quiz to translate the  
        word to the target language  
        options: string[], // A list of options  
        to choose the correct translation for the  
        word in the target language  
        correct: string, // the correct translation  
        for the word in the target language  
    }  
}
```

For example, you receive the following JSON:

```
{  
    words: ['leaves'],  
    language: 'english',  
    targetLanguage: 'portuguese'  
}
```

And you need to generate a quiz for the word 'leaves' in english to portuguese:

```
{
```

```

    "leaves": {
        quiz: 'I love the color of the leaves in fall',
        options: ['folhas', 'deixar', 'sair', 'deixar cair'],
        correct: 'folhas'
    }
}

```

Following that example, you need to generate a quiz to those words based on that JSON:

```

{
    words: ${JSON.stringify(words)},
    language: ${JSON.stringify(language)}, // the language original of the words
    targetLanguage: ${JSON.stringify(targetLanguage)} // the language target to translate
}

```

Currently, the prompt is used to generate the quiz, but AI is employed also to generate translations and meanings of the words. Once the AI receives the prompt and returns the requested information, the backend processes and stores it in the database. Storing the quiz is crucial because each request to the AI incurs a cost, which is determined by the number of tokens used in the request.

If the approach were to manually create a database and quizzes, both the cost and time investment would significantly increase, leading to a deterioration in user experience. Given the large number of words required to generate a quiz, users would face a substantial learning curve in acquiring a new language.

4.3. The games

The games are developed to transform the learning process singleton, it means that each word has the own line of learning and when appears on the user screen the user do something that answer if the user knows the word or not.

As choose three games to generate, the first one is a quiz, the second one is a word search, and the third one is a flashcard.

The quiz is show a phrase using a word in the language and the user need to choose a correct translation for the word in the target language. The options are made by the AI.

The word search is show a word in the language and the user need to find the correct translation for the word in the target language. The options are made by the AI.

The flashcard is show a word in the language and the user need to choose with he knows the word or not.

Those games are designed based on apps like anki and duolingo.

4.4. Learning algorithm

Based on SRS and the games generated by the information provided by the AI, as the games are design to transform the learning process singleton, when the API send informations about the word, the game will show based on priority specified on the api.

More priority, early the word will appear on the user screen, and the user will see the word

more often.

The api create a ranking of words and the priority of the word to appear on the user screen. The api ask three questions based on the data storaged:

1. Is today the day for learning?
2. How often does it repeat?
3. Are there many errors?

Following exactly those question the API creates a ranking and filter the words.

4.4.1. Is today the day for learning?

As the api use a SRS technique, the api needs to know if today is the day for learning the word. The approach choose was a exponential function, that is based on the number of times that the user saw the word and when the user do a mistake. The function is based on the following formula:

$$f(x, y) = 2^x \quad (1)$$

Is a simple equation, for example: if the user saw the word today and don't do a mistake, he will see the word again in 2 days, if the user continue without do a mistake, he will see the word again in 4 days, and so on. But if the user do a mistake, the function will be reseted and the user will see the word again in 1 day.

That question filter the words that the user will see today.

4.4.2. How often does it repeat?

The api also know how common is this word (based on the entire database) and how commom this word is on subtitle. The api get the percentage of the word in the subtitle and the percentage of the word in the entire database, and based on that the api create a ranking of the words.

Each of those information is 50% of that ranking, and the api create a ranking based on that information.

4.4.3. Are there many errors?

The api also know how many errors the user do with that word, and how many errors the entire database do with that word. The api get the percentage of the word in the subtitle and the percentage of the word in the entire database, and based on that the api create a ranking of the words.

As the previus step the api create a ranking based on that information, taking 50% of database and 50% of the user.

5. Results

A browser extension was developed that interacts with movie streaming sites and extracts the subtitles available on these sites. The extension uses Spaced Repetition System (SRS) techniques to gamify learning new languages based on the premise that knowledge of keywords or the most repeated ones allows for text comprehension. The goal is to expand to other platforms like Netflix, Disney+, and others.

The site was also developed to provide statistics on user progress, including the most challenging words, the total words learned, and others to help the user understand their learning process. The site also supports multiple languages, which allows users to learn a wide range of languages using AI translations and the creation of games.

The app was developed to provide a way for the user to study on every device, displaying the iframe with the application site.

This work has demonstrated the potential of using technology to facilitate language learning. Integrating our tool with everyday activities such as watching movies can make language learning more engaging and compelling.

5.1. Interface

As a result, each environment has its functions and responsibilities, and based on that it needs a different interface(resources) to have a better responsibility and user experience. And improve the features that each environment possible. The user can interact with the application with the ambient mobile and browser, using the app, site, or extension. Each interface is different and has different layouts and features to improve the user experience.

5.1.1. site

The module of the site has an interface that adjusts to the user's device. The site is the base of the extension and app, and those use an iframe to show the site. The design is based on long and small screens to support the user in every device. The site also provides a way to communicate with the extension and the app, and the user can access the site to continue training the words from the streaming platform. Some examples of ways to interact with the application are:

- Languages Page: where the user started to learn, the possible movies to study, and the movies that the user has started, as demonstrated in the figure-9.
- Language List: A List of languages that the user can learn, as demonstrated in the figure-10 and in the mobile version in the figure-17;
- Profile Page: The user profile with progress for each language, as demonstrated in the figure-11.
- Languages by Movie: A list of languages when the user selects a movie, as demonstrated in the figure-12.
- Games Page: A list of possible games that the user can play, as demonstrated in the figure-13.

Figure 9. Initial site interface for the user with the languages, movies, and started movies.

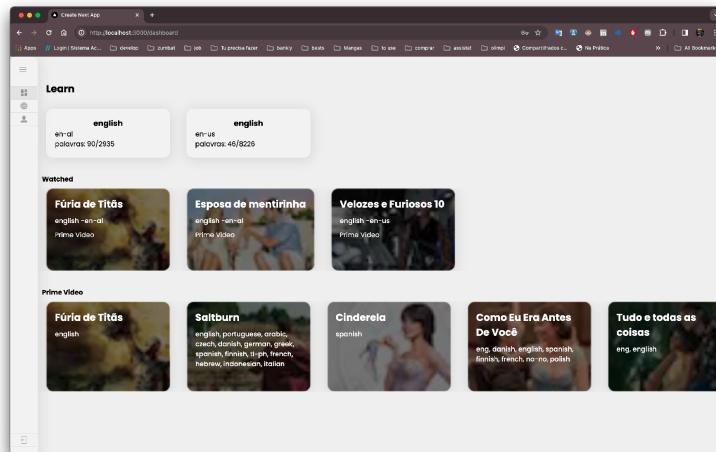


Figure 10. A list of languages that the user can learn.

Languages				
pashto ps-of total: 7878	swedish sv-ox total: 848	english en-es total: 2935	french fr-dz total: 2347	catalan ca-es total: 406
spanish es-es total: 999	dutch nl-nl total: 1076	german de-de total: 1447	portuguese pt-br total: 14699	malay ms-bn total: 231
french fr-ca total: 1747	greek el-cy total: 6	czech cs-cz total: 1088	danish da-dk total: 2524	finnish fi-fi total: 1731
french fr-fr total: 2699	german de-de total: 1869	hungarian hu-hu total: 471	indonesian id-id total: 1544	italian it-it total: 3371
spanish es-es total: 6550	polish pl-pl total: 1494	spanish es-es total: 3135	english en-us total: 8226	vi-vi vi-vi total: 180
eng	ti-ph	no-no		

Figure 11. The user profile with progress for each language.

The screenshot shows the user profile page. At the top, it says 'Welcome, demo demo!'. Below that is a form with fields for 'Email' (demo@demo.it), 'Senha' (*****), 'Nome' (demo), 'Sobrenome' (demo), and 'Telefone' (123456789). To the right, there is a summary for the 'english - en-es' language pair:

Score: 0		
english - en-es	Total de treinos	Palavra mais vista
Em aprendizado 90	182	I'm
Palavra mais difícil I'm	Total de erros 89	
english - en-us		

At the bottom are 'Editar' and 'Sair / logout' buttons.

Figure 12. A list of languages when the user selects a language to learn.

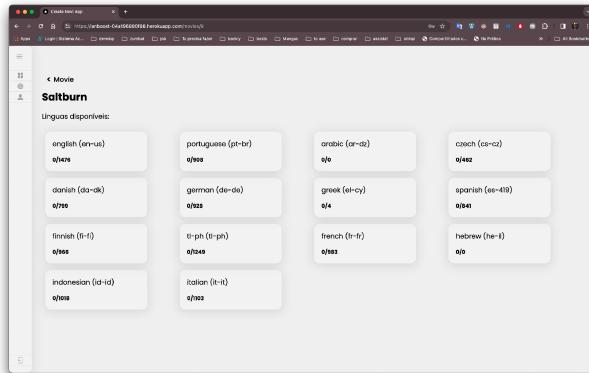
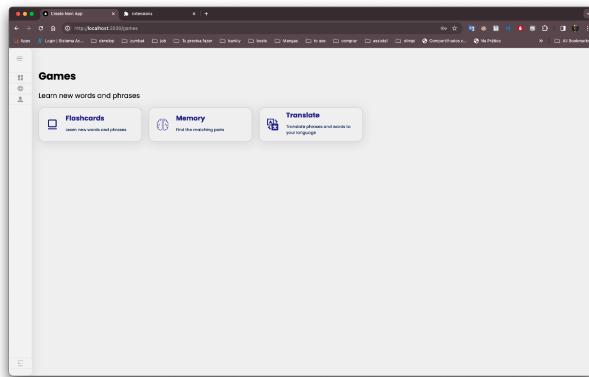


Figure 13. The possible games that the user can play.



5.1.2. iframe to extension

This module is used on modal when the users opens a movie on the streaming site. The user could interact with the application with inicial interface, equal of site in the figure-13, but inside a iframe, where the user could select the gamer witch he wants to play. And those games are the Flashcard, Memory, and Quiz, as demonstrated in the figure-14, figure-15, and figure-16. The entire playing process is inside the iframe, and the user can interact with the application without leaving the streaming site.

Figure 14. Games Flashcard inside iframe on the prime video site.

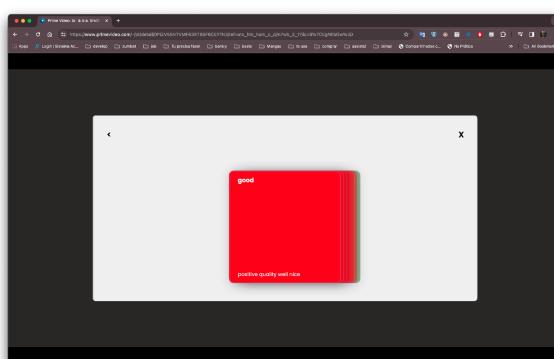


Figure 15. Games Memory inside iframe on the prime video site.

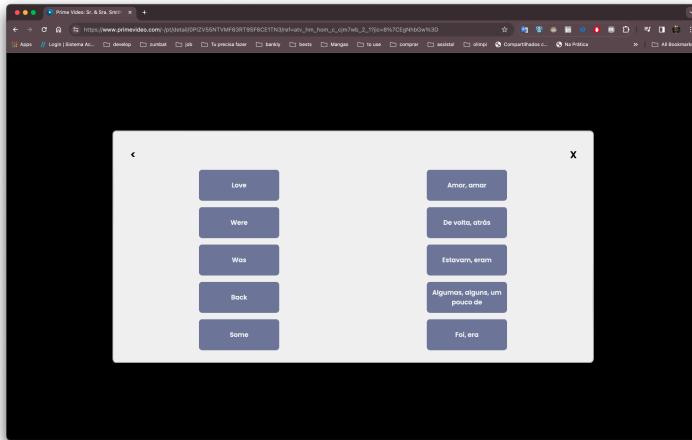
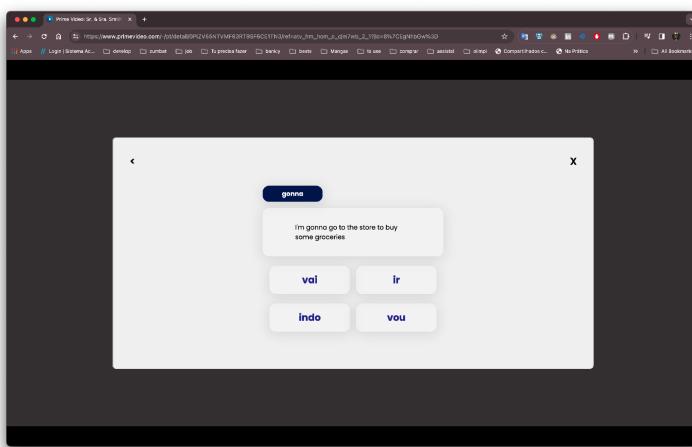


Figure 16. Games Translate inside iframe on the prime video site.



5.1.3. iframe to app

This module is used by the app, but also used on the pop-up of the extension. This provide a small interface and resize components to adapte to that environment. Examples of ways to interact with the application with the following interfaces:

- Login Page: a screen to the user login as demonstrated in the figure-18.
- Home Page: that screen provide a list of languages and movies as demonstrated in the figure-17.
- Languages Page: this interface is where is listed all languages available to learn on the application as demonstrated in the figure-19.
- Language by movie page: list all languages for each movie as demonstrated in the figure-20 and figure-21.
- Game Translate Page: A game where the user needs to inform the correct translation for a word in a specif phrase demonstrated in the figure-22.
- And the others games as the site and iframe extension.

Figure 17. List of languages and movies inside iframe on the app.

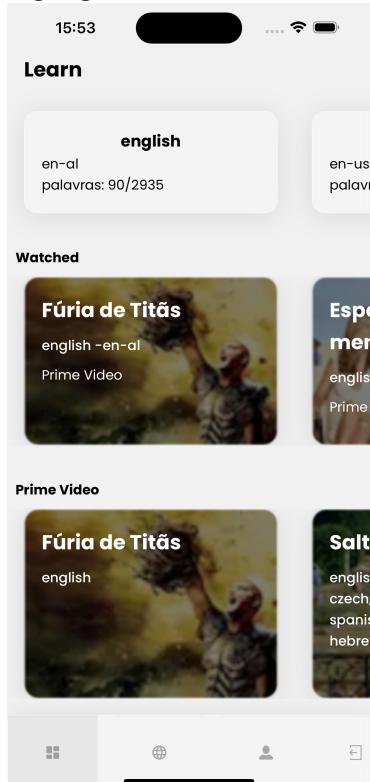


Figure 18. Login inside iframe on the app.

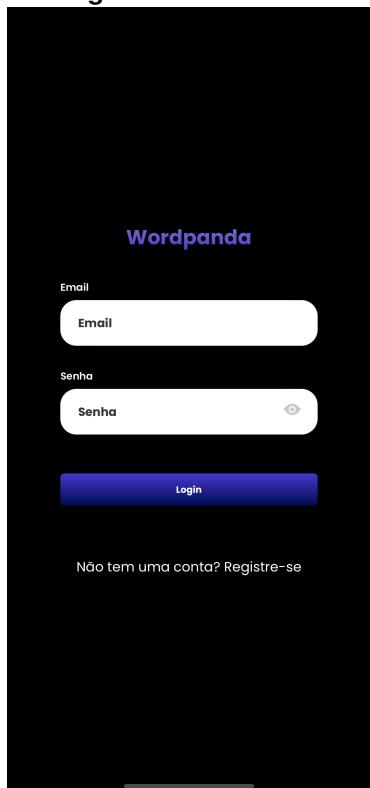


Figure 19. List of languages inside iframe on the app.



Figure 20. List of languages by movie inside iframe on the app.

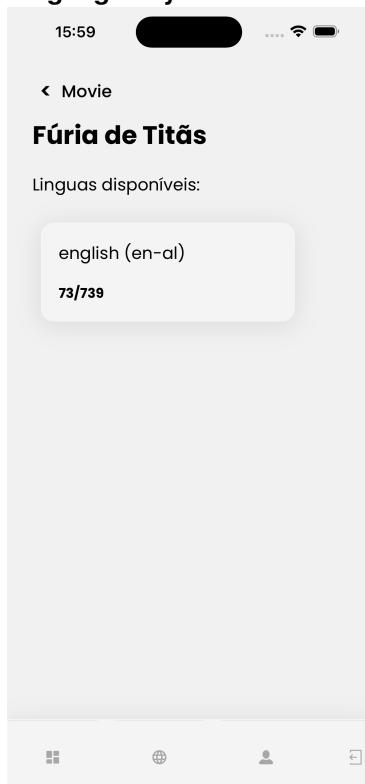


Figure 21. List of languages by movie inside iframe on the app.

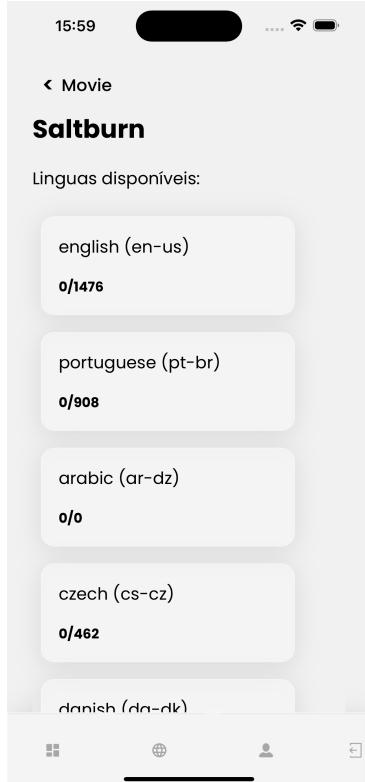


Figure 22. Translate game inside the the iframe on the app.



5.2. Future Improvements

For future improvements, there are subsections for each one and each environment (extension, site, app, backend, and AI) to improve the prototype. Because this work is a prototype, the goal is to improve the user experience and the learning process, make the application more responsive and fast, create a better interface, improve the user experience, and reduce costs to maintain the application.

5.2.1. Extension

The extension needs to be some refactors to turn more responsible and fast, because the background service that takes the subtitles is slow and executes a cronjob to get the subtitles. Is necessary to study ways to do this using dispatch events and workers.

Currently, the repertoire of sites is so small that the extension could communicate, and restrict only to Prime video, but the goal is to expand to other platforms like Netflix, Disney+, and others.

Also is important to create a WebSocket to extract pieces of information from the site, considering that some streaming uses another technology to show the subtitles.

Considering that is possible to create communications between extensions or sites, a discussion about the possibility of integrating with other tools like Language Reactor, is viable.

5.2.2. Site

The site is the core of the application and is necessary to improve the user experience and make it more responsive. The site will be expanded to include new features like interactive exercises and games to reinforce vocabulary acquisition.

About the games, is necessary to create a better interface and improve the user experience. Because the games are the part visible at the core of the learning process, the user needs to have a good experience to continue using the application. The site was created without a professional designer, and games are the part that needs more attention in design.

The entire site needs to be more responsive and fast so that the user can understand and use it without any problems. The UX is missing, and the site needs to be redesigned.

Levels, interests, and other user information are not being used to improve the user experience, and the site needs to be customized to the user's level and interests, providing an interactive and engaging learning environment.

The last part about UX is the current language of the site, which is not possible to change the site language and is a relevant part of improving the user experience.

More about technology is interesting to associate the site with the WebSockets also, and create new channels to communicate with mobile and notify the user to study like a reminder.

5.2.3. App

The app will be improved to capture device information and send it to the iframe to make it more responsible and fast. This possibility sends to the backend that information and

creates a dashboard or extract to discover the user behavior and how to improve it. The channel to communicate with the site to possibly configure notifications is also an important feature, because a huge part of the users use the app to study, and the app needs to remember the user to do this.

5.2.4. Backend

On the backend, the API routers will be improved using workers because the data will be extensive, and the app and site will be faster and more responsive.

Some API routers will be improved using workers because the data will be extensive, and the app and site will be faster and more responsive.

As the data of each word is a complex process a solution is to build routines to run at night that will recalculate the user progress and the word level based on the results of all users. Another important part is the language manager, that will be changed using only the language name, not the language code.

And the last part is the web sockets, which will be created to improve communication between the extension and the site, and the site and the app.

5.3. AI

The AI of choice is the gpt-3.5-turbo-1106 model developed by OpenAI, incurring costs of \$0.0010 per 1K tokens of input and \$0.0020 per 1K tokens of output. The goal is to reduce costs. Enhance prototype performance and stabilize monthly payments by leveraging proprietary AI technology. Considering the adoption of Smaug-72B-v0.1, an advanced open AI model known for surpassing the capabilities of my current paid AI solution. The transition to utilizing in-house AI will occur when it becomes more advantageous than relying on third-party APIs.

5.4. Conclusion

In conclusion, this work has successfully developed a browser extension, a website, and an app that aim to facilitate language learning by extracting subtitles from movie streaming sites. The extension utilizes Spaced Repetition System (SRS) techniques to gamify language learning, while the website provides statistics on user progress and supports multiple languages. The app allows users to study on any device.

Moving forward, several steps are planned for each application component. The extension will be expanded to other streaming sites, languages, and platforms and it will be integrated with other language-learning tools. The website will include interactive exercises, games, and customized learning experiences. The app will capture device information and improve communication with the website.

In addition, the application's backend will be enhanced with improvements to API routers and to the language manager. Web sockets will improve communication between the extension and the website.

Overall, this work demonstrates the potential of using technology to make language learning more engaging and accessible. By integrating language learning with everyday activities like watching movies, users can enhance their language skills in a fun and interactive way.

Guilherme A. Avelino

J. Emanuel Cascone R. S.