Language Retention and Tutoring: LanGauger

Ambadipudi, Chaitra
cambadipudi@tamu.edu

Kulkarni, Akshay
kulk890@tamu.edu

Mankala, Sai Sujith
saisujithreddy@tamu.edu

Potturi, Saketh Ram
sakethrampotturi@tamu.edu

Siravara, Vishwas
vsiravara@tamu.edu

ABSTRACT

We all have a good reason to learn a new language; discovering our roots, passion for travel, academic purposes, pure interest etc. However most of us find it hard to become conversationally fluent in a new language while we use traditional resources for learning like textbooks and tutorials on the internet. In this paper we propose a novel approach to learn a new language. We aim to develop an intelligent browser extension, LanGauger, that will help users learn foreign languages. This application will allow users to look up words while they are browsing, by highlighting the text to be learned. The application will then provide a translation of the word, its pronunciation and its usage context in sentences. In addition, this intelligent tutor will also remember what words have been seen by the user, and quiz them on these words at appropriate times. While testing the recall of the user, this feature will also allow users to frequently think about the language and use it.

Categories and Subject Descriptors

K.3.1 [**Computer Uses in Education**]: Computer-assisted instruction (CAI); H.5 [**Information interfaces and presentation**]: Group and Organization Interfaces

General Terms

Web-based interaction

Keywords

Intelligent tutoring, computer assisted language learning

1. INTRODUCTION

It was established in [2] that one-on-one tutoring performed two standard deviations better than traditional teaching methods. In addition to teaching techniques like these, the crux of education lies in frequent and immediate communication and feedback to students [6]. Additionally, education that involves learning new languages requires frequent and deliberate use and recall of key phrases within those languages [3]. When these three ideologies are tied together, it calls for a highly interactive tutor that provides an individualized tutoring experience for every student. With the rise in computer aided learning, improvement of Artificial Intelligence techniques, and the increasingly heavy usage of the Internet world-wide, we decided to develop an intelligent browser plugin that can perform all these functions and help users learn foreign languages. For such a system that caters to teaching languages, features that test the recall of students and encourage them to use it more naturally are warranted.

2. MOTIVATION

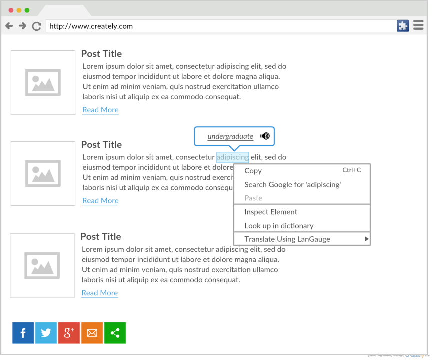
Current language tools extend so far as to assist a user by converting the entire web-page into a user's known/natural language. While these tools help the user understand the content of the page in the target foreign language, they do not play a major role in helping the user absorb the language. Our tool LanGauger provides complete control to the user as to when and what he wants to learn. The tool allows the user to just convert the words the user wishes for into his/her natural language. This ensures that the user learns words that are important to him, and his domain, while building a base of the target foreign language. While language learning tools like Duolingo extensively help a user learn new languages consciously [4], LanGauger helps them acquire it intuitively. For the former, users have to navigate over to a separate website and learn a set course content, which might or might not work for them. Also, strengthening vocabulary can be challenging when the course involves set content. The content that LanGauger uses is much more flexible, and harnesses the power of the Internet. Additionally, we have taken inspiration from existing tools that quiz users occasionally and help in recall. Thus, LanGauger augments current features of language learning tools while adding on new features. This paper provides an insight into LanGauger implementation details and its ability to assist a user in learning a new foreign language.

3. TASK ANALYSIS

3.1 Target users

The target user base is those who are aiming to learn new languages, and who have access to the Internet and a browser, either on their phone, tablet or desktop PCs.

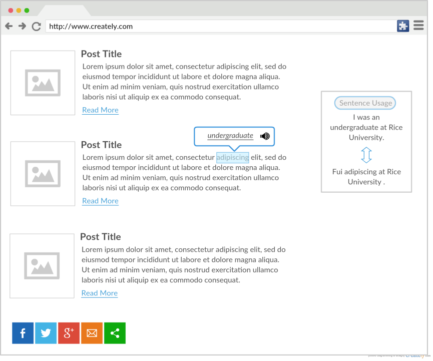
CSCE 634, 2015 Texas A&M University
.



Post Title

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Figure 1: Selecting a word and looking it up



Post Title

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

I was an undergraduate at Rice University.

Fui adipiscing at Rice University .

Figure 2: Contextual usage of a word

3.2 Features planned

3.2.1 *Word lookup and user interaction*

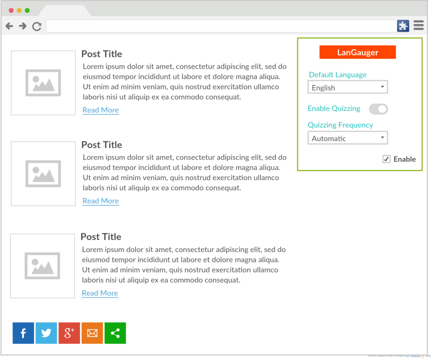
The main aim of LanGauger is to allow users to compare words in the to-be-learned foreign language to the meaning in the learner's known/natural language, by letting them select a word or a phrase (Figure 1).
This choice was made in the consideration that selection of text feels more intuitive than other forms of input, like typing in words. The extension then uses its existing knowledge base to:

- explain the word: provide the meaning of the word in the user's natural language
- illustrate it: provide context-based usage of the words in the user's target foreign language. (Figure 2).
- break it down: display the phonetics of the word in the user's natural language and
- localize it: provide a text-to-speech conversion of the word to allow the user to listen to the pronunciation of the word in the target language

3.2.2 *Registration and sign in*

The tool allows users to register with it initially and later on subsequently sign in, thus keeping track of his or her word look-up history.

3.2.3 *Quizzing engine*



Default Language: English

Quizzing Frequency: Automatic

☑ Enable

Post Title

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Figure 3: Choosing language preferences

Whenever a user logs back in, the browser extension will evaluate his current learning status. When the user browses pages where there are words that he has learned before, or are similar to those that he learned, it will pop up a quiz to help evaluate the user's past learning. The tool will show words that the user is weak on more frequently. Once the user reaches 'expertise' level on a particular word, it is no longer shown to him. This is where the intelligence of the extension comes into play.

3.2.4 *Language preferences*

The tool allows users to change their natural language and their target foreign language from the Preferences section of the application. (Figure 3). With flexible preferences, word look-up can also be implemented in the reverse direction. When the user is reading an article/page on the browser in the unknown language and is stuck at a particular word, he can change the source language to the foreign language and the destination to the natural language using the preferences. Then, selecting a word will immediately pop up a dialog box above the word with the definition of the word in the user's natural language.

3.3 System Architecture

Figure 6 points to a block diagram that shows the main components of the architecture of the tool. The whole architecture of the system can be divided into four modules: Content Module, User Module, Translation Module and Gaming Module. The following subsections describe each of the modules in detail.

3.3.1 *Content Module*

The content module consists of the text content which the user is browsing through. The content can be from any website; for example, it could be some text in a different language which the user finds on Facebook. If the user wants to know more about a particular word, highlighting it will display more insight about the word which our system can provide. The system can use different engines to educate the user more about the word.

3.3.2 *User Module*

The user module has a registration & login for signup/login and also consists of information specific to a particular user. The user can also set preferences like his language, privacy, etc. in this module. A list of words mastered by the user (called Mastered Words) is stored in the user information so that it can be used later in other modules like the Game Engine Module.

3.3.3 *Translation Module*

The translation module is accessed when the user highlights a particular word to know more about it. The translation module consists of two main components: Translation Engine and Contextual Engine. The Translation Engine helps the user translate a word to English and also gives him both the audio pronunciation and phonetics of the word. The Engine consists of a text-to-speech converter to generate the pronunciation of the word. But, to understand the context in which a particular word is used we need to know the usage of the word. The Contextual Engine is used to get the contextual sentences of a particular word. The contextual sentences help the user learn the usage of the word by providing him with different sentences in which the word is used. Therefore, the user is more educated about the word when he uses both the Engines than when he uses only the Translation Module to translate a particular word. After the user has mastered the word by browsing through both the modules and learning it, it is added to the list of the Mastered Words in the User Module.

3.3.4 *Gaming Module*

The gaming module uses the concept of games to test the user's level of mastery of the text present in the Content. The gaming module helps the user learn the language quicker by quizzing him the words present in the content module. The Game Engine quizzes the user by giving him some multiple choice questions to recognize the meaning or the usage of different words. If the user answers them correctly then they are added to the list of the Mastered Words. Thus the user is tested for his mastery of the language and also games act as a learning medium since it keeps the user more engaged and educates him more than traditional reading.

3.4 Implementation details

The architecture of the system as described in the previous section forms a base for the decisions made to implement the plugin. The first task is to allow users to make a profile in the system. The plugin will work even if the user does not register, but to track progress and enable active tutoring specific to the user a profile needs to be created. The process is pretty straightforward to register and login to personalize the plugin to the user(Figures 5 and 6). The profile is where user related features such as preferences and mastered words are associated. The database of the framework contains two tables to represent the user module. One table is used to store user login data, language preferences etc. and the other table is used to keep a list of words the user has learnt and is currently learning along with the mastery score for the word.

The next thing is to implement the language translation engine. The module is build using the Bing API for language translation. The module takes a word in the primary language and the secondary language to which it needs to be converted as inputs and returns the translated text as output. More details on the implementation are described in section 3.5.1.

Along with sending the inputs to the translation engine,

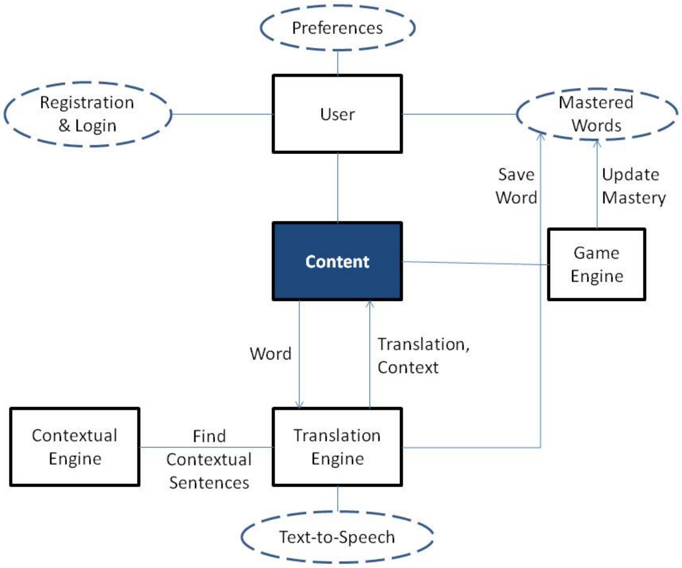
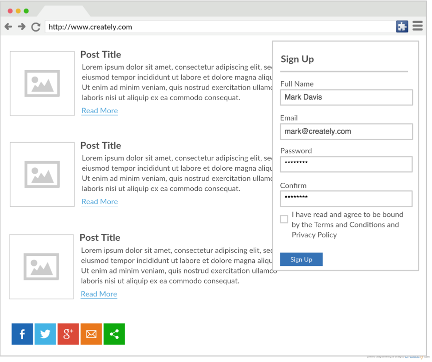


Figure 4: An architecture diagram linking the main components of the tool



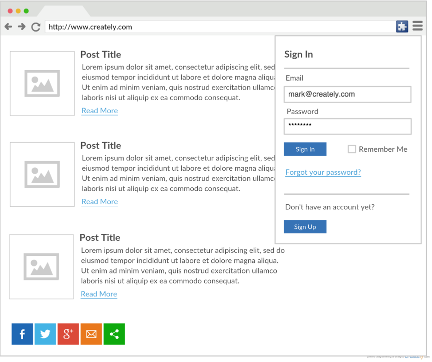
Post Title
Lorem ipsum dolor sit amet, consectetur adipiscing elit, se
eiusmod tempor incididunt ut labore et dolore magna aliqu
Ut enim ad minim veniam, quis nostrud exercitation ullamc
laboris nisi ut aliquip ex ea commodo consequat.
Read More

Post Title
Lorem ipsum dolor sit amet, consectetur adipiscing elit, se
eiusmod tempor incididunt ut labore et dolore magna aliqu
Ut enim ad minim veniam, quis nostrud exercitation ullamc
laboris nisi ut aliquip ex ea commodo consequat.
Read More

Post Title
Lorem ipsum dolor sit amet, consectetur adipiscing elit, se
eiusmod tempor incididunt ut labore et dolore magna aliqu
Ut enim ad minim veniam, quis nostrud exercitation ullamco
laboris nisi ut aliquip ex ea commodo consequat.
Read More

Sign Up
Full Name: Mark Davis
Email: mark@creately.com
Password: ********
Confirm: ********
□ I have read and agree to be bound by the Terms and Conditions and Privacy Policy
Sign Up

Figure 5: Registration for the system



Post Title
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed
eiusmod tempor incididunt ut labore et dolore magna aliqu
Ut enim ad minim veniam, quis nostrud exercitation ullamc
laboris nisi ut aliquip ex ea commodo consequat.
Read More

Post Title
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed
eiusmod tempor incididunt ut labore et dolore magna aliqu
Ut enim ad minim veniam, quis nostrud exercitation ullamc
laboris nisi ut aliquip ex ea commodo consequat.
Read More

Post Title
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua.
Ut enim ad minim veniam, quis nostrud exercitation ullamco
laboris nisi ut aliquip ex ea commodo consequat.
Read More

Sign In
Email: mark@creately.com
Password: ********
Sign In □ Remember Me
Forgot your password?
Don't have an account yet?
Sign Up

Figure 6: Logging into the system

they are also sent to the Contextual Sentences module and Text-to-Speech module. The contextual module queries the database for the sentences involving the word being translated and returns a list of sentences which help understand the context of the word. This module is built as a service and is independent of the others. More information of the implementation is provided in section 3.5.2. The Text-to-Speech module takes in a word and queries the Voice RSS API to get the audio of how the word is pronounced. This is also implemented as a service. More information of the implementation is provided in section 3.5.3.

Once the outputs are returned back to the Google Chrome plugin, they are displayed on screen. The translation and the audio pronunciation is displayed in a bubble above the highlighted word on the webpage and the contextual sentences are displayed in a separate window to the side as displayed in Figure 2.

Every time a word is referenced by the user to learn the secondary language using translation a copy of the word is stored in the database linked to the user's profile. The word is associated with a mastery score, which indicates how confident the user is and the retention level of the word. The score can be from 0 to 100. The first time the user sees a new word the score starts from 0.

All the above details show how the implementation is executed to help the user learn the language, but the important part is retention of the word and its translation in long term memory. Contextual sentences and pronunciation help in improving memory retention of the new language [5], but along with memory retention quizzing is also a vital part of achieving this. The Quizzing module consists of a text scraper and an Artificial Intelligence mechanism to formulate questions. The text scraper implemented by using the Boilerpipe API (`https://boilerpipe-web.appspot.com/`) extracts the HTML text on a webpage and returns it in text form. This serves as an input the quizzing algorithm which cross references it with the mastery list associated with the user and pops out a question to help recollect and recognize already assimilated knowledge. The question can be either one with multiple options presented or one where the user is give the word and pronunciation and needs to identify the word in the primary language. Depending on the evaluation of each pop quiz, the mastery level for the particular word is updated in the database. The mastery score is updated based on adaptive evaluation algorithm. The more questions answered without mistakes, the faster the score increases and vice versa. Once the mastery reaches 100, the word is safely assumed to be in the long term memory and questions for this word occur sparsely later on.

The initial development for each of the above mentioned modules is done independently. After the development of the services the modules are integrated into a single system to build the language translation and tutoring using artificial intelligence. The system is built in the form of a webpage and is then converted into a Google Chrome extension, as it supports the basic blocks of web development: HTML, CSS, and JavaScript (`https://developer.chrome.com/extensions`). Once the plugin is generated from the web page, it will be installed as an extension in the Google Chrome browser and tested out as LanGuager.

3.4.1 User modeling

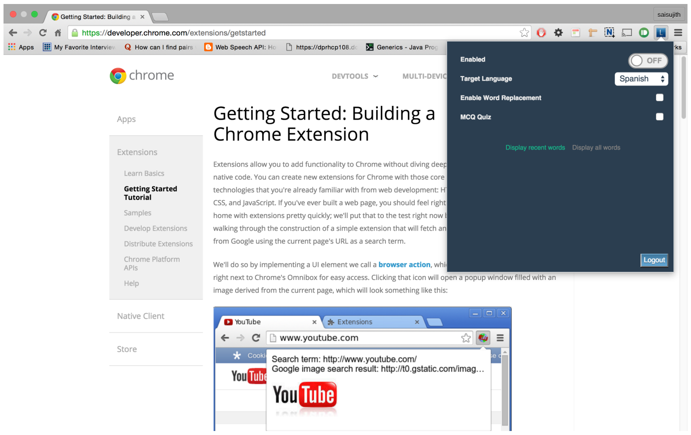
The extension is built with the aim of customizing the system specific to each user. In order for the system to learn about the user, information should be explicitly requested to build a user model and learn from it. Once the model is constructed based on explicit details, the system can start learning about each user by combining the user model with user interaction with the system. The simplest way to model and gather user details is by using a registration and login system. The registration and login schema fulfills the initial task of gathering basic details about the user and get started with creating the user model. The details of each user are stored in a table called *users* in the database. All the entries are written into this schema and used going forward.

The login and registration system only achieves the initial demographic data collection. More information is need about the domain knowledge of the user. This is where the preferences page comes into play, where the user needs to provide explicit information about the domain languages and the goal of using the system. The preferences page also provides with options to enable or disable features in the system along with selecting frequencies of features provided by the extension to help the user. This page provides the power to control the intelligence continuum between the system and the user. All the details about the domain provided in the preferences dialog are stored in the *preferences* table of the database. But the options provided by the extension such as enabling/disabling the system or other features, controlling frequency etc. need to be stored on a local level to reduce reaction time of the system and thus increase performance of the system along with user experience. Chrome provides storage on the browser for variables which need quick seek and access, the extension makes use of this storage to store the other preferences as a map.

3.4.2 Translation and Representation of words

There are a lot of translators available such as Google Translate, Bing Translate, Yandex etc. for developers which provide a programmable interface to have words translated from one language to another. Google Translator API is a cloud-based automatic translation (a.k.a. machine translation) service supporting multiple languages. The aim of the translation process of this extension is to provide near real-time translation of the words selected by the user in the browser. Google's public translator interface provides these features along with Microsoft's Bing Translate. Both these APIs provide a varied range of translation choices and provide suitable response to interpret by the extension. Both Google and Bing translate have been implemented in the Langauger extension and either one can be used. The Translator API also takes care of the representation and visualization of languages, for example, if a translation is made from English to Arabic, then the API provides the response Arabic unicode characters.

As a first step to translate words using the Translator API we created a PHP webpage and used a local web server. The application was later deployed on to Amazon Web Services. The web page is shown in Figure 7. The user initially selects the target language from the dropdown in the preferences.This is shown in Figure 8.As soon as the user changes the target language, the subsequent words would be converted into this changes target language. The user selects a word and depending on the target language selected in the preferences, the translated output is displayed in the same page in its native characters.This is shown in Figure 9.



Enabled OFF

Target Language Spanish

Enable Word Replacement □

MCQ Quiz □

Display recent words Display all words

Logout

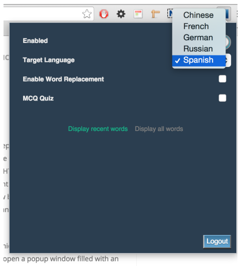
Getting Started: Building a Chrome Extension

Extensions allow you to add functionality to Chrome without diving deep native code. You can create new extensions for Chrome with those core technologies that you're already familiar with from web development: H CSS, and JavaScript. If you've ever built a web page, you should feel right home with extensions pretty quickly; we'll put that to the test right now walking through the construction of a simple extension that will fetch an from Google using the current page's URL as a search term.

We'll do so by implementing a UI element we call a **browser action**, whi right next to Chrome's Omnibox for easy access. Clicking that icon will open a popup window filled with an image derived from the current page, which will look something like this:

Search term: http://www.youtube.com/
Google image search result: http://t0.gstatic.com/imag...

Figure 7: Web page with the extension showing the preferences.



Enabled

Target Language Spanish

Enable Word Replacement □

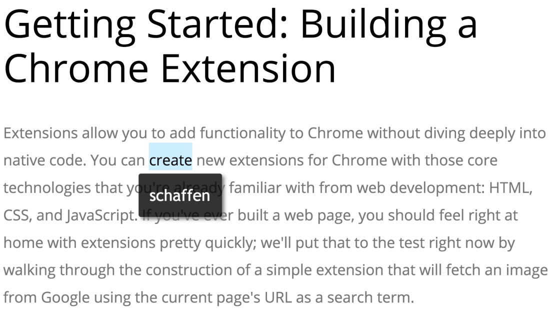
MCQ Quiz □

Display recent words Display all words

Logout

open a popup window filled with an

Figure 8: A drop-down inside the preferences showing the target languages.



Getting Started: Building a Chrome Extension

Extensions allow you to add functionality to Chrome without diving deeply into native code. You can create new extensions for Chrome with those core technologies that yo[schaffen] familiar with from web development: HTML, CSS, and JavaScript. If you've ever built a web page, you should feel right at home with extensions pretty quickly; we'll put that to the test right now by walking through the construction of a simple extension that will fetch an image from Google using the current page's URL as a search term.

Figure 9: Word create(English) converted to schaffen(German)

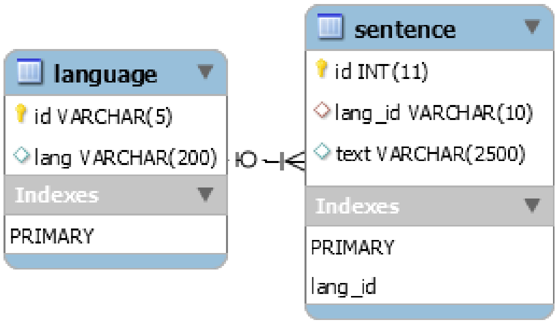
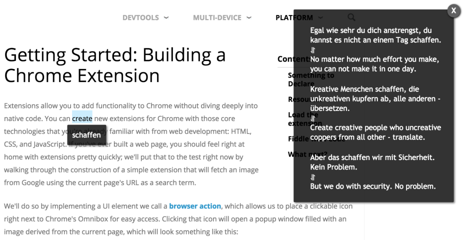


Figure 10: A schema of the database used to store context based sentences



Getting Started: Building a Chrome Extension

Extensions allow you to add functionality to Chrome without diving deeply into native code. You can create new extensions for Chrome with those core technologies that yo[schaffen] familiar with from web development: HTML, CSS, and JavaScript. If you've ever built a web page, you should feel right at home with extensions pretty quickly; we'll put that to the test right now by walking through the construction of a simple extension that will fetch an image from Google using the current page's URL as a search term.

We'll do so by implementing a UI element we call a **browser action**, which allows us to place a clickable icon right next to Chrome's Omnibox for easy access. Clicking that icon will open a popup window filled with an image derived from the current page, which will look something like this:

Egal wie sehr du dich anstrengst, du kannst es nicht an einem Tag schaffen.
⥮
No matter how much effort you make, you can not make it in one day.

Kreative Menschen schaffen, die unkreativen kupfern ab, alle anderen - übersetzen.
⥮
Create creative people who uncreative coppers from all other - translate.

Aber das schaffen wir mit Sicherheit. Kein Problem.
⥮
But we do with security. No problem.

Figure 11: Contextual sentences with the usage of schaffen(German) and the corresponding sentences in English.

3.4.3 *Contextual usage of sentences*

`http://tatoeba.org/eng/` hosts publicly available collections of sentences in various languages. For the purposes of this project, we decided to use this data and index it in such a way that inputting a word will retrieve example sentences that can be shown to the user. The data collection was modified to reflect the schema shown in Figure 10.

1. The '`language`' table contains ISO 639-3 codes from `http://www-01.sil.org/iso639-3/default.asp`, which is a standard list of 3-letter codes for each language.

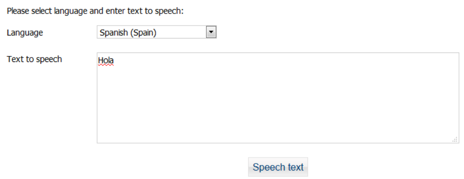
2. The '`sentence`' table contains the context sentences, and a foreign key that points to the 3-letter code from the previous table, indicating which language a particular sentence is in.

3. In addition to defining the schema, the collation of the database was set to UTF-8 Unicode CI to ensure that it can handle all the accented characters from various languages.

The contextual sentences which uses the translated word is shown on the right side of the web page. This is shown in Figure 11. The sentences which uses the word in the target languages are shown first and the corresponding sentences in English are shown below. The idea will be to show the user shorter sentences initially so that it will be easier for him or her to grasp the context. As the retention of the word progresses,more complex sentences can be shown.

3.4.4 *Pronunciation of Sentences*

Although the user may see the phonetics of the word in the translator, not many can interpret the pronunciation



Please select language and enter text to speech:

Language: Spanish (Spain)

Text to speech: Hola

Figure 12: A word and an output Language given as input parameters to the Voice RSS API

just by seeing the phonetics. Therefore, the best way we thought to educate the user with the pronunciation was to use the Voice RSS API. The API is used for converting the text input which has been captured by the user into speech output. This gives the user first hand pronunciation of the user.

The API currently supports 26 different languages ranging from Mandarin to several European and American languages. The API takes two inputs: the language of the speech output and the text that is to be pronounced. We do a GET request from the API using those two parameters and we get back speech output. This API is used when the user clicks on the 'pronounce it' option on the selected word. Now that user has more first hand pronunciation of the word, his level of expertise of the word has increased.

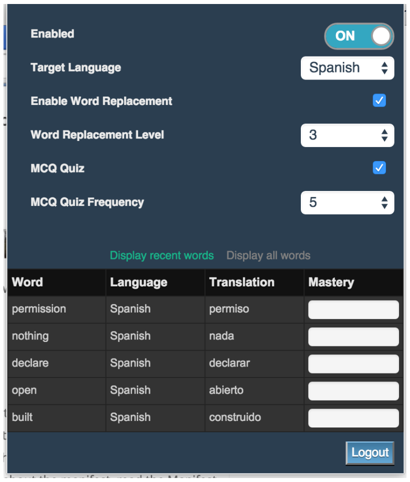
The API could also be used in the quizzing engine. We plan to have some questions in which the user has to type the spelling of different words by listening to them. We think that the API could in very handy in such questions.

An implementation of the Voice RSS API is shown in Figure 13. The input provided to the API is the word Hola (Spanish word for Hello). Output produced by the API is the pronunciation of Hola in Spanish accent.

3.4.5 Word Storage

Over a period of viewing and learning foreign words we may sometimes feel the need to check how we have fared so far in learning new words. Our extension allows the user to check the words they have encountered and their mastery level in those particular words.

The recently translated words are stored in the extension. The stored words are displayed in two styles. One style displays the five most recent words translated. The other style displays all the words translated. Each style has four sub fields. First sub-field is the word in English which is actually translated to the target language. With the help of this field users can easily pronounce the word without having to listen to it again. The second sub-field displays the target language used for this translation. This clears the ambiguity of the language of the word if the same word is used in multiple languages. The third sub-field displays the translated word in the target language. This is for more advanced purposes if the user wants to learn to read and write the words in the foreign language. The last sub-field displays the mastery achieved on the word. This helps us keep track of our learning process and to check if we have become acquainted enough with the particular word. The level of mastery is updated when user takes up the quiz which is demonstrated in the next section. The fields of word storage can be seen in Figures 13 and 14. Overall, the word storage feature helps the user track his progress of learning a new language.



Enabled: ON

Target Language: Spanish

Enable Word Replacement: ☑

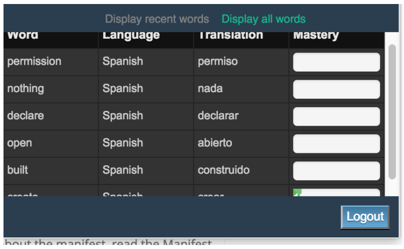
Word Replacement Level: 3

MCQ Quiz: ☑

MCQ Quiz Frequency: 5

Word	Language	Translation	Mastery
permission	Spanish	permiso	
nothing	Spanish	nada	
declare	Spanish	declarar	
open	Spanish	abierto	
built	Spanish	construido	

Figure 13: Most recent five words translated.



Word	Language	Translation	Mastery
permission	Spanish	permiso	
nothing	Spanish	nada	
declare	Spanish	declarar	
open	Spanish	abierto	
built	Spanish	construido	
create	Spanish	crear	

Figure 14: All the words translated so far.

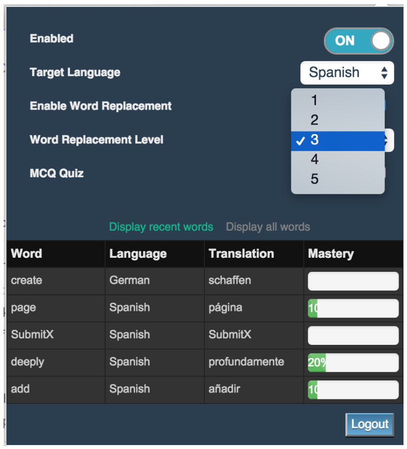
3.4.6 Word Replacement

To learn a new language quickly we could use words from the new language in our native language sentences so that we can get more acquainted the new language. The very same thing could be done while reading as well. Our Extension allows the user to replace the words in the text to the target language so that users get more used to the words and also their contextual usage.

To turn this feature on we have to check on Enable Word Replacement from the drop down menu of the Extension. After Word Replacement has been enabled we get to choose the Word Replacement Level. The levels range from one to five where level one replaces minimal number of words in the text and level five replaces maximum number of words in the text. Suppose we have selected the word replacement level to be one (which is the least possible level), then a word is replaced in every two paragraphs. Suppose we have selected the word replacement level to be 5 then a word is replaced in every two lines. Selection of levels of word replacements is shown in the figure 15. Now that we have replaced some words from the text we could easily grasp the new language since we could interpret the context and usage of the words from the live text. Hence this method improves the cognition of the users.

3.4.7 Quizzing Engine

Education that involves learning new languages requires frequent and deliberate use and recall of key phrases within those languages[6]. This enables the user to constantly think about the language and the contextual use of words which



Enabled ON

Target Language Spanish

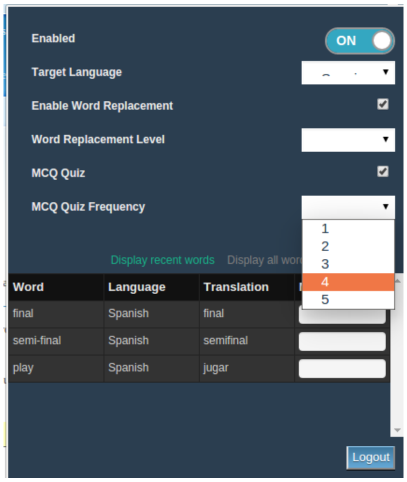
Enable Word Replacement

Word Replacement Level ✓ 3

MCQ Quiz

Word	Language	Translation	Mastery
create	German	schaffen	
page	Spanish	página	10
SubmitX	Spanish	SubmitX	
deeply	Spanish	profundamente	20%
add	Spanish	añadir	10

Figure 15: Selection of Word Replacement Level.



Enabled ON

Target Language

Enable Word Replacement ☑

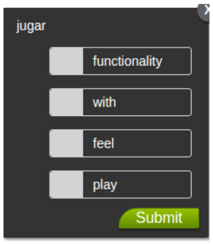
Word Replacement Level

MCQ Quiz ☑

MCQ Quiz Frequency 4

Word	Language	Translation	
final	Spanish	final	
semi-final	Spanish	semifinal	
play	Spanish	jugar	

Figure 16: Frequency of quizzing can be selected from Drop Down. Here frequency is set to 4



jugar

Figure 17: Question appears on the top right side of the screen

helps the user master a new language. LanGauger provides an option for users of the extension to take quizzes periodically in the form of multiple choice questions.
The level of frequency of the quizzes can be adjusted by the user through a drop down menu . The level ranges from one to five where five is the highest frequency of quizzing. The User Interface for selection of frequency is shown in Figure 13. Once the user sets the frequency, questions pop up in the top left corner along with 4 options. Each question has exactly one correct answers in the options listed (Figure 14). The user gets ”mastery” points for each correct answer . The points is given based on speed of answering. The quiz words along with the mastery score are saved in the extension and a user can check the quiz word list from the extension and check his mastery of the language. This feature is shown in Figure 16.
Currently, our quizzing engine works by taking words from the already encountered words by the user. The rate at which quizzes appears depends on the frequency. If the frequency is one (which is the least possible value), a quiz appears only after some time which is currently about a minute after the page has been refreshed. If the frequency is five (which is the maximum possible value), a quiz appears each time the page is loaded. By adjusting the frequency of quizzes we can change the time of occurrences of quizzes. If we don’t want the quizzes to distract us from inspecting the content on the screen we can simply disable the quizzing engine from the drop down menu. Therefore, the quizzing engine provides a platform for ’rehearsing’ the words which the user has already encountered. We plan on improving the quizzing engine by incorporating questions in which user has to recognize the meaning of the words by listening to it rather than seeing a text based question.

4. EVALUATION

Even though the system has not been tested to provide results for the evaluation components identified, the methods are discussed in detail if evaluation needs to performed.

4.1 Cognitive Walkthroughs

An expert needs to look at the system from a usability point of view and walk through all the components and features the system has to provide. The cognitive expert performs a series of typical user tasks supported by the system. At each task the expert asks whether the following questions [7] can be answered when a typical user interacts with the system:

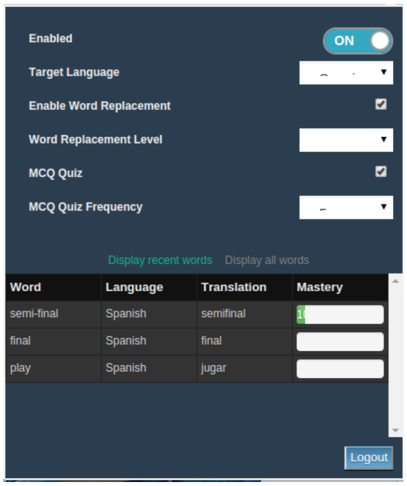
1. Will the user try to achieve the right effect?
2. Will the user notice that the correct action is available?
3. Will the user associate the correct action with the effect to be achieved?
4. If the correct action is performed, will the user see that progress is being made toward solution of the task?

The expert assesses if each point will be a success story or a failure story, and if something is a failure story, then provides insights as to why the user will not be able to complete the task. These insights will help in improving the usability of the system.

4.2 Heuristic Evaluation

Heuristic Evaluation is performed by one or more Human Factors experts. Evaluators measure the usability, efficiency, and effectiveness of the interface based on 10 usability heuristics originally defined by Jakob Nielsen in 1994 [1]. Some of the 10 usability heuristics are

- Provide users timely and appropriate feedback about the system status. If it takes a long time to load a screen, display a progress bar and/or an estimate of



Enabled ON

Target Language

Enable Word Replacement ☑

Word Replacement Level

MCQ Quiz ☑

MCQ Quiz Frequency

Word	Language	Translation	Mastery
semi-final	Spanish	semifinal	1
final	Spanish	final	
play	Spanish	jugar	

Figure 18: Quiz word list along with the level of mastery

the time it may take to load, so users know what to expect.

- Match Between System and the Real World. When designing a system, use terms with which they are familiar and display information in formats they are used to seeing.

- Help and Documentation. Including ”in-line help” in the form of a link that opens a a small help dialogue next to the form field. Alternatively, next to a form field text box or field label, provide users an example of how to input the information using the required formatting.

4.3 Usability Testing

The best way to understand how real users will experience the system is by performing usability testing. Interviews and focus groups allow the user to describe the behavior or performance of the system, but the responses in these methods are informal and not structured to interpret properly. Instead, in case of usability testing well-designed user tests are conducted which measure the actual performance of mission-critical tasks. Only if the users are able to perform these tasks successfully, can the system be regarded to have a good interface and user experience.
In case of the Langauger extension in particular questions relating to the amount of control provided by the system and the performance of different components would be really useful. The quizzing engine might be displayed too frequent or less frequent, the extension might be distracting when the user is busy and reading something important and other such questions can be used in the questionnaires to evaluate the system better.

5. FUTURE WORK

While coming up with features for our tool, we also thought of some ideas, upon which we are yet to perform feasibility analysis and come to a conclusion. These would be great additions to the tool. We are documenting them in this report.

- Speech-to-text conversion: This feature will allow a user to check his pronunciation by letting him speak into a microphone, and then comparing it with the pronunciation of a native speaker’s.

- Social collaboration: This feature will allow users to discuss about words or phrases on a forum-like platform, thus enabling exchange of more context clues.

6. REFERENCES

[1] Usability methods. Accessed December 8, 2015.

[2] B. S. Bloom. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational researcher*, pages 4–16, 1984.

[3] R. Ellis. *The study of second language acquisition.* Oxford University Press, 1994.

[4] S. Krashen. Does duolingo ’trump’ university-level language learning? Technical report, The International Journal of Foreign Language Teaching, 2011.

[5] J. L. M. Mark F. St. John. Learning and applying contextual constraints in sentence comprehension. *Artificial Intelligence*, 1990.

[6] V. J. Shute and J. Psotka. Intelligent tutoring systems: Past, present, and future. Technical report, DTIC Document, 1994.

[7] C. Wharton, J. Rieman, C. Lewis, and P. Polson. The cognitive walkthrough method: A practitioner’s guide. In *Usability inspection methods*, pages 105–140. John Wiley & Sons, Inc., 1994.