WordWhip

Dylan Whitlow and Nick Bissiri

October 1, 2016

1 Concept

WordWhip will be a Chrome browser extension used to assist users studying a foreign language through word replacement in web page text. Words in the user's native language will be replaced with given vocabulary in the studied language, essentially injecting flashcards into the text of things like news articles. The extension will have a few parts. The first will display a translation of a highlighted word or phrase of the native language in the foreign language. Obviously once words have been replaced with the foreign language, users should also be able to hover over them to check the meaning in their native language. The second will be grouped lists of vocabulary words, to which users can add translations of highlighted words or those defined by user input. For example users could have a list for each chapter of vocab in their textbooks that they can toggle on or off independently. The third is a feature common to flashcard apps: users can share the lists that they have made and download lists that others have made. Since no one wants to have to enter all of this data into more than one application, we would like to support uploading common file formats of flashcard decks to generate vocab lists in our extension.

2 Existing Extensions

There are a few extensions similar enough to mention. The first is called Readlang, which has the fundamentally different goal of helping its users read articles in the foreign language they are studying. Used backwards it resembles the first part of our extension. It also tracks words you look up and can make flashcards.

The second extension is Mind the Word. Unlike Readlang it works in the same direction as our application, replacing words in the native language with those of the foreign, but without the ability to click on a native language word and see its translation or optionally add it to the replacement lists. It does still offer the option of directly inputting the words to replace. It does not give users the ability to share their vocab lists. Its main appeal seems to be that by syncing with some translation service (e.g. Yandex, Bing, Google), it can automatically replace words at random in text. That is not as helpful as it initially sounds, because it requires users to blacklist common words such as articles and prepositions and also because it exposes users to many words way beyond their reading level. It also has a fairly obtuse UI.

Language Immersion for Chrome is another similar application, but it does not allow users to control what words are replaced beyond specifying their fluency level. Another application for this sort of thing is Lingualy, but it costs \$5 so it will remain a mystery (it seems most similar to Readlang).

3 Possible challenges

- Dictionary support could potentially be challenging. Several free bilingual dictionary files exist on the web, but these are not guaranteed to have consistent formats. Accessing and formatting dictionary data may require much more work than expected.

- Making the search feature user-friendly will be an interesting challenge. Not all lists should be considered equal: inevitably some users will submit junk lists. Meta-data about each list such as popularity of use or rating should be maintained for each list to allow users to easily find useful lists. Furthermore, users may be interested in searching lists by their content: such as the words contained or the list's difficulty.

- Excellent design of list implementation will be important for adding features. For example, one cycle of development may allow only a single list of words, but we would like to allow users to enable multiple lists or even lists in different foreign languages. With a poor initial design, list collision could warrant a complete overhaul of the list implementation. Many added features are likely to affect lists or list meta-data.

4 Software Architecture

We will use Javascript for the front end of the extension. Several default dictionaries stored on a server may be available for users to download for translating words. These dictionaries and any user-defined dictionaries are stored on the client side. Alternatively, dictionary searches may be performed through the API of one or more bilingual dictionary web apps (such as Google Translate). Each user's lists will be stored on the client side, but may be uploaded to a server side database. That database will be searchable by users through queries processed on a back end server. An appropriate language such as Python or Ruby would be chosen early in development for implementing the back end. Several sources of free bilingual dictionaries are available on the web such as wordreference.com.