Location-Based Auto-Suggested Vocabulary Lists: A Design Exploration

Victoria Kirst
Computer Science Department
University of Washington
vkirst@cs.washington.edu

Andrew Yurovschak
Computer Science Department
University of Washington
yuro@cs.washington.edu

ABSTRACT
Beginners learning a language often have trouble communicating because of lack of vocabulary knowledge. In this paper we introduce a prototypical iPhone application that auto-suggests relevant vocabulary words and definitions to a user based on the user's location. Through a series of explorative design iterations, we study the issues that arise in designing such an application. We also conduct a series of informal evaluations to gauge reactions to and the desirability for the application and use this feedback to fuel future iterations. We found users were generally positive about the premise and found the idea of location-based vocabulary lists intuitive.

Keywords
Language learning, iPhone applications, language barriers, iterative design, web prototyping, auto-suggestion

INTRODUCTION
One of the biggest difficulties in traveling to a foreign country is overcoming the language barrier. Even if someone studies a foreign language for many years, there is still a period of time when he or she is lost in vocabulary.

While there have been countless software applications made to help make language learning easier (e.g. Declan, Rosetta Stone, Tell Me More), there are not many applications meant to assist with daily, on-the-spot communication. People living abroad speaking in a language they are not comfortable with often carry around electronic dictionaries to help with day to day conversation, but dictionary lookups are cumbersome and slow, which can make everyday interactions difficult, embarrassing, and stressful. We have begun to explore an idea to address this problem: instead of having to look up definitions for words manually in a dictionary, have an application that automatically suggests relevant words to the user's conversation.

In this paper we propose a prototype of an iPhone application that suggests useful vocabulary words to a person learning a foreign language. Our heuristic for what constitutes a "useful" word is based on location: the iPhone application would use GPS to locate the user, then would suggest vocabulary words and definitions to the user that one would likely use in that location. For example, if a user at the beach, our application should suggest vocabulary such as "sunscreen lotion."

We describe our iterative design and implementation process in creating a prototype of this application. Our intentions were not to work toward creating perfect, deployable application, but rather, we used each version of our prototype as a vessel to open up discussions about the benefits and the issues involved in creating location-aware vocabulary lists. We have introduced three different versions of this application to student participants from the University of Washington (UW) and we describe their thoughts, feedbacks, and suggestions. We also provide some of our own observations regarding the difficulties in creating this application and our suggestions for future work in this direction.

PROJECT OVERVIEW
Our project was an implementation-driven design exploration, which meant we had to be able to create multiple iterations of our project in the 6-week period. Because we were designing this prototype with an eye on the broader issues rather than the minute details of a location-aware vocabulary system, we made several design decisions to avoid wasting time implementing details that were uninteresting for our study's purposes, but without losing the "illusion" of what the mock-up is supposed to represent.

Web Prototype
It was infeasible to create an actual iPhone application due to the time constraint and instead we simulated the iPhone app via a web page. Using PHP, JavaScript, and AJAX technologies, we were able to create three design iterations of a web prototype of this iPhone application. We stored the locations, suggested words, and user information in a MySQL backend. The source code for the project is available on Google Code [2].

While iPhone applications and web applications have much in common, there were certainly some differences for which we had to compensate. One important functionality that an iPhone application has that a website on a laptop does not is location lookup via GPS. Instead of determining the location of the user for the user, we had a list of locations the user can select from prior to loading the prototype.

We also lacked an iPhone ourselves, so we decided to demo our application on a laptop. There were several advantages to using this approach for our application,

including ease of development and deployment, but there were also downsides. These issues we elaborate in more detail in the Discussion.

Translations

There is an issue that comes up when trying to populate a mapping from locations to vocabulary words: in what language do you store the mappings? For example, if the word "apple" were associated with the grocery store, it would be nice to have apples associated with grocery stores in all languages. But this requires storing the mapping in some arbitrary original language first, since there is no universal base language from which all languages derive. We decided to store all our location → words mappings in English, from which we then translate to whatever language is requested.

One of the most important parts of this application in practice would be the quality of translation of the vocabulary words, but these were details we considered uninteresting as there are several high-quality foreign language dictionaries already in existence and reinventing these dictionaries is a study in linguistics, not HCI. However, we felt it was still necessary to create the "illusion" of word translation to give the user a better feel of the application.

We populated our database with a list of all words and definitions in a public domain English dictionary [1]. We then used the Google AJAX Language API to translate these words and definitions to and from different languages. The precision with which the Google Translator could translate and define various words was hazy at best, but the actual translation did not matter as much as the illusion of translation for our prototype.

It is worth distinguishing some of the things our prototype does versus what we think the real application should do. In the real application, the backend would keep track of the words that are being mapped to locations. When the vocabulary word is suggested to the user, both the original word and its translation are presented. However, because there is not a one-to-one mapping between all words in one language to another, it is important to provide definitions for the proposed foreign vocabulary words to make sure the proposed word is indeed what the user would like to say – homographs in particular can cause translation errors.

For example, say you are an English-speaker trying to learn Korean. The word "pen" is proposed at the bank, and the Korean translation is "우리." To make sure that the correct translation has been proposed, you would want to know the possible definitions of "우리" in English.

Pen - 우리 (def: 1. A cage, a sty)

We now can see that "우리" is the translation for something like a pig pen, not a ballpoint pen like we might have thought.

Managing what words' semantic definitions are associated with locations is certainly a relevant problem for this application, but we decided that getting the *correct* definitions was not necessary for our prototype. Instead of providing the definition of the translation, we provide the English definition of the original word requested:

Pen - 우리 (def: 1. An instrument used for writing with ink)

This is an incorrect translation, but it serves its use for prototypical purposes. This still allowed us to model interesting problems with incorrect definitions in our third iteration of our design.

Evaluation Process

The nature of our project did not lend itself well to quantitative evaluation. So many things were changing between iterations that it would have been very difficult to obtain accurate quantitative results, and ultimately such results would not be of much use to us when designing multiple iterations of the prototype.

Instead, we conducted informal evaluations for each iteration of our prototype with 5-10 participants per evaluation. We used different participants for each evaluation so as not to bias the participants' feedback with a learning effect. Our objective was to evaluate the usability of the system and to gauge what features were useful and what were distracting. We also wanted to explore what features the participants desired in such an application to get a better sense of what a final version of this application might look like.

Our evaluation sessions were conducted face-to-face with random UW students in public locations such as the HUB or the dorm's eating facility. We began the session with a brief explanation of our study's objective and our prototype, and then we encouraged the participant to look up definitions for words as they would if they were using this device in a foreign country. We did not give them much instruction about how to use the device unless we were asked or unless the participant was stuck, as we wanted to see what parts of our design was intuitive and what was clunky.

We did not have a strict rubric for participants to follow when evaluating our product, but we started the discussion with the following questions:

- What were your impressions as you used the app?
- What did you find confusing? What didn't you like?
- Do you have any suggestions to improve the application or features you would like to request?

This feedback we collected and discussed before we planned the design for next iteration of our application.

RELATED WORK

Other academic work has been done to explore the use of location-aware applications to make inferences about the user's needs and context. Natalia Marmasse and Chris

Schmandt of MIT Media Laboratory designed comMotion, a computing environment that suggests to-do lists based on user locations [3]. It tracks the user's travel patterns and then asks the user to label locations once enough evidence has been gathered that a location is significant.

Martin Bauer et al from University of Stuttgart presented two methodologies to model location in context-aware applications [4]. Eija Kaasinen from VTT Information Technology studied the user needs and attitudes toward location-aware devices through empirical studies and scenario evaluations [5]. Kaasinen found that generally users desire topical information when using application on mobile devices, as static information can be researched at home.

The use of computing and the internet to aid foreign language learning has been the topic of several published papers as well. Meena Singhal from the University of Arizona discusses the benefits and challenges of using the internet to teach language [6]. Wang Han from Yantai University studies the history of computer-assisted language learning (CALL) as well as the barriers and benefits of technology in education [7]. Our investigations have not led us to the discovery of other published articles or applications regarding auto-suggested vocabulary lists based on inferred situation context.

FIRST ITERATION: INVESTIGATION OF PRODUCT

In our first iteration of our application, we created a rough, barebones implementation of the design to present to people as a starting point. We wanted the direction of our project to be fueled by user feedback, so we purposely created a prototype with only basic functionality, without making many assumptions about what a user might like as additional features. Figure 1 shows a screenshot of this iteration.

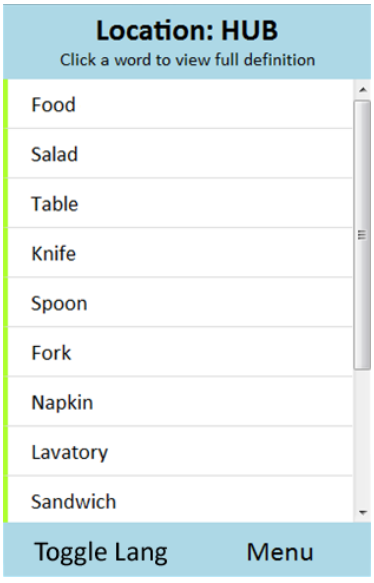

Figure 1: First iteration of design

Static Word List

In this iteration, everything was static: we pre-selected the location to be a restaurant, and we pre-populated our database to have a static list of vocabulary words relating to restaurants, such as "fork" and "menu." The native language was set to English and the foreign language to German. The list of words was given to the user in English, and one could toggle between English and German via a button press. Double clicking on a word led the user to a definition page that showed all definitions for the word in English, whose German translation could also be derived by pressing the toggle button. Users could also search for the definition for a word that was not suggested by our application.

Feedback

Generally, our participants liked the idea of the application and could see the usefulness and desirability of such an application. The concept of location-based auto-suggested vocabulary lists was intuitive to them and did not require much explanation or justification on our part.

Fine-Grained Issues

A fair portion of our feedback we received in the evaluations for each iteration of our design dealt with some of the smaller details of the application, such as color and aesthetics. While this was not the focus of our study, we fixed some of the trivial usability issues so as not to be a distraction to participants in future iterations.

Populating location → words map

One particular question we had in mind for our next iteration was how to populate the location → words map. Pre-populating words for every location was slow and impractical, so we planned to instead have users associate words they look up in the dictionary with locations, possibly by tagging certain words to locations. In the evaluation for this iteration, we also asked for feedback regarding this process.

Generally, participants were supportive of the idea of tagging words to locations. They also wanted to see what words other users have tagged to certain locations as well, and one participant suggested an idea of automatically adding a word to a location when the definition lookup is performed.

A larger issue was also brought to surface: is location necessarily the correct heuristic for vocabulary word suggestions? Our application is based off the assumption that the topic of conversation often correlates pretty well with one's location, but that is not necessarily the case. It may be just as useful to have an "offline" suggestion mode, where a user could browse for vocabulary words associated with various locations without having to be physically at that location.

SECOND ITERATION: SCALABILITY

In the second iteration of our application, we explored designs to make our application more scalable and

deployable. We now had a sign up and login page that allowed users to select a location from a pre-determined list as well as a primary and secondary language. The application then took the user to a list of suggested vocabulary words, and clicking on a word would bring the user to the definition page (Figure 2). The suggested vocabulary words were now user-generated instead of statically pre-populated, and users had the ability to choose between a "roaming" mode, where GPS determined the location of the user and suggested words based on that mapping, and a "learning" mode, where the user could browse through a set of locations and view the vocabulary words associated with each location (Figure 3).

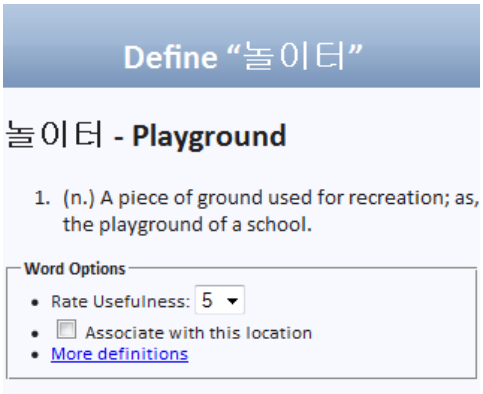

Figure 2: Definition page in second iteration

User-Generated Location→ Word Mappings

When a user looks up a definition for a vocabulary word, the definition page has an option to "Associate word with location." If a user checks this box, the application saves the word – location relationship for this particular user. The suggested vocabulary list is then filled with the words that the user has associated with the current location as well as some number of the most popular words that have been associated with the location by all users. Words can be ranked on a scale of 1 to 5 for usefulness to further sort the list.

There was a design decision to make about how to store locations: do we store specific locations, general locations, or both? "University Teriyaki," for instance, is a specific location, whereas "restaurant" is the general location category. Certainly it would be redundant if users had to populate from scratch the word – location associations for every instance of a restaurant, so it would make sense to have words associated with categories of places. At the same time, limiting word associations to *only* categories of places would greatly limit the variety and usefulness of the vocabulary lists; one wouldn't necessarily want the exact same vocabulary list at McDonald's as he would want at the Four Seasons.

One option would be to allow a user to choose every time whether the word should be tagged to the specific or general location, but this would be clunky and not particularly scalable. Instead, words are tagged by default to specific locations. Every location is given a category behind the scenes, and words that appear in several locations of a particular category are then promoted to being associated with that category.

Figure 3: Options page in second iteration

Roaming versus Learning Mode

In the options menu, users choose whether they would like to be in "roaming" mode, where the vocabulary list is chosen automatically based on a GPS lookup of the user's current location, or a "learning" mode, where users browse through a hierarchy of vocabulary lists and choose the list they would like to use. In this iteration, the vocabulary list types were our general location lists (bank, restaurant, etc.) though future iterations of the application would not need to limit these static vocabulary lists to locations.

Feedback

Our participants generally liked and understood how to use this version of the application, though we noticed a few trends that called for revision. In particular, participants did not tend to search for words, rank words, or associate words with the vocabulary list:

- Users did not deem the search feature as a major feature of the application. This was likely due in part to the test environment, as a user in a foreign country would certainly feel more motivation to look up words not presented in the vocabulary list, but this was still clearly a problem with our interface.
- If the vocabulary word was already associated with a location because of other users, the participant was often confused why he too would need to associate the word with the location and very rarely checked the box to associate the word with the location if it was already in the vocabulary list.
- Furthermore, many participants did not immediately understand the idea of "associating" a word with a location and did not choose to do so very often.

These fixing these usage trends became our top priority for the third iteration of the application because the word – location mappings that are used to auto-suggest vocabulary lists rely on users searching for what words are useful to them and associating them to locations.

Another common problem came from the definitions of words: the first definition given for a word was often not the desired one, and words sometimes had as many as 30 useless additional definitions. Participants requested the ability to associate certain definitions with locations instead of simply words.

Finally, it was inconclusive how learning mode and roaming mode compared in user experience. This was largely due to the limitations of the prototype: because we had to simulate roaming mode by having the participant choose his or her location from a menu, this simulated roaming mode was essentially the same as static learning mode. Any comparisons made between the two modes were therefore speculative. Further studying the differences between learning and roaming mode is something mentioned in the Future Work section of the paper.

THIRD ITERATION: DICTIONARY FOCUS

The third iteration of our application addressed the usability issues of the second iteration. Fundamentally, our application is an intelligent dictionary: users are still searching for vocabulary word definitions, but they are given a suggestion list based on their location. Our previous iterations had the suggested vocabulary lists as the focus of the application when truly the dictionary is still the focus, and the emphasis on the auto-suggested vocabulary lists conflicted with the intended usage patterns for the application.

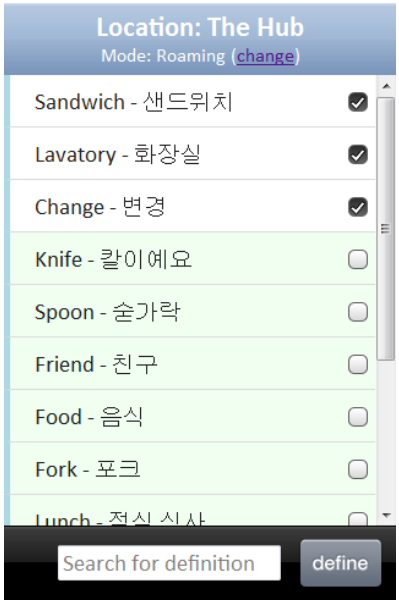

Figure 4: Vocabulary list for third iteration

Searching

The main screen of the application now features a prominent search bar (Figure 4). As the user types, the partially written words are used to filter the user's vocabulary list. Community-suggested vocabulary words are given a different background color as user-suggested vocabulary words, and each word has a button to add or remove the word from this location.

"Tagging" Definitions

We rephrased our idea of "associating" words with locations to "tagging" words to locations. Tagging is a much more familiar paradigm and requires far less explanation than "word – location associations."

We also changed our method of tagging from tagging words themselves to tagging specific definitions. Users could tag words directly from the vocabulary list page, and this would tag the most popular definition to the location.

Feedback

Overall, it seemed this iteration of changes streamlined the learning curve for our application. The participants asked fewer questions than in previous iterations and seemed to discover the various features without needing instruction.

The participants evaluating this iteration of our design also used search far more frequently than before. In the first and second design of our application, participants would evaluate the product by clicking vocabulary words and often did not realize search was an option until told. In this iteration, participants still examined vocabulary words first, but also explored the search bar just about as frequently. The participants were far more likely to play with tagging and untagging words as well, especially since there were now buttons on the suggested vocabulary words to allow for this.

There were still some problems with understanding tagging. Participants understood the idea of tagging *words*, but tagging *definitions* was not as intuitive. One participant mentioned she'd like to tag words and rank definitions for the word. In terms of functionality, our current implementation does exactly this, but it seems we have not abstracted the implementation enough to be usable.

DISCUSSION & FUTURE WORK

Our study has provided an interesting investigation of the many aspects of location-aware applications. It is difficult to make any hard-and-fast conclusions based on the results of our limited and informal user evaluations, but we feel we have brought up many relevant issues that should be considered again in further brainstorming and development of location-based vocabulary lists.

Populating the Location → Words Map

Certainly one of the most interesting and most challenging aspects of location-based vocabulary suggestion is the task of populating the location → words map.

Our approach was to trust users to search for needed words and tag useful words to locations. Because searching and tagging is such a vital part of initially populating the vocabulary lists, dictionary search should be a prominent, easy to access feature. Users must have a motivation to tag words, and tagging words to location must be easy and clear.

We could not test this with a large user base, but even in our smaller evaluations we saw the value of letting users tag words to locations. It would be helpful to deploy a larger-scale prototype in a future study to analyze usage trends.

Location Categorization

The problem of categorization was an ongoing problem throughout the development of the prototype. We decided to let users tag words to specific locations, which in the backend were associated with more general location categories, and tagged words common among location categories would show up as vocabulary suggestions.

However, our categorization system was not very scalable or flexible. For example, both the Seattle Aquarium and the Woodland Park Zoo were categorized as "Zoo" and so words like "tiger" were suggested at the aquarium. These mappings were not useful and generally confused users that encountered them. The problem is symptomatic of the static way that categories are presented in the prototype.

One solution is to place the creation of categories and the labeling of locations in the hands of the users. This solution would also scale a much larger geographical area, since category tagging would not have to be done by a single authoritative source. Some noise in the tagging is to be expected, but overall the application would be much more flexible and would accommodate multiple categories per location.

Knowledge Retention

The application is designed to teach vocabulary, but it currently has very little knowledge of what the user has learned. The same words will be presented to the user regardless of whether the user already knows them. It would be useful to have the ability to judge the vocabulary knowledge of the user and then supply words of suitable difficulty. This could be done easily enough through a quizzing system, but the more difficult problem is assigning a difficulty to each word. A heuristic method might use the number of letters in the word and the letter composition, but this might be less useful for character-based languages. Like the category-tagging problem above, the best solution may be to give the responsibility of difficulty ranking to the users, although this assumes there is a suitably large user base to carry out such a task.

Cultural Bias

Another problem that needs to be addressed is that certain words are biased toward a specific culture and may not be useful in cultures of different languages. For example, depending on the culture, the words "taco" or "burger" may be appropriate for the category "Restaurant" in one language, but not in another. It is possible to restrict word associations to a pool containing only users learning the same language, but this may significantly reduce the number of overall mappings a user sees. Extra mark-up may be needed to distinguish language-centric words, although this runs into the same problem of who's responsible for annotating words.

Limitations of Our Model

In many ways, a laptop is fundamentally different than an iPhone: it is not handheld, it has a large keyboard, and its pointing device is the mouse, not a touch screen.

One thing that the web prototype did not capture accurately was the touch-based nature of the iPhone. The mouse also has the effect of inflating the accuracy of user's pointing ability, which tends to result in buttons and other UI elements that are too small for a touch interface.

It was also difficult for the app to simulate the feel of auto-suggestion. The web app assumes an ability to perfectly identify a user's location, and does not have to take into account a user's movement, lack of signal, or ambiguity in locations (is a user at the book store or the restaurant next to it?). Because of such limitations, our participants in some ways could not help but evaluate the application as if it were a website instead of an iPhone application, and future prototyping using a more portable device would likely bring up issues we could not explore in our study.

CONCLUSION

The development of our prototype showed that location-mapped vocabulary has the potential to be a useful device for learning a foreign language. Although no quantitative assessment was done, feedback from users was positive and turned up no major problems with the concept itself. The usefulness of web prototyping the application instead of using lower fidelity methods is undetermined. There is a significant cost to web prototyping, which needs to be weighed against the benefits of an interactive web prototype. Future work on the concept will focus on further improving the scalability of the application by delegating responsibility to the users or through other methods.

ACKNOWLEDGMENTS

We thank James Fogarty and Kayur Patel for the wonderful course this quarter. We also thank the many kind students who were willing to participate in our evaluation studies.

REFERENCES

1. The Online Plain Text English Dictionary. Available at http://www.mso.anu.edu.au/~ralph/OPTED/
2. Project source code. Available for download at http://code.google.com/p/cse510project/
3. Marmasse, N. and Schmandt, C. Location-Aware Information Delivery with *ComMotion*, in *Lecture*

Notes in Computer Science. Springer Berlin / Heidelberg, 361-370, 2000.

4. Bauer, M., Becker, C., Rothermel, K. Location Models from the Perspective of Context-Aware Applications and Mobile Ad Hoc Networks in *Personal and Ubiquitous Computing*. Springer London, 322-328, 2002.
5. Kaasinen, E. User needs for location-aware mobile services in *Personal and Ubiquitous Computing*. Springer London, 70-79, 2003.
6. Singhal, M. The Internet and Foreign Language Education: Benefits and Challenges in The Internet TESL Journal, 3(6): 1997.
7. Wang, H. Benefits and barriers of computer assisted language learning and teaching in US US-China Foreign Language, 6(9): 2008.