Developing an SRS-Based Intelligent Tutoring System

Myles Lefkovitz
Goergia Tech
Atlanta, USA
mlefkovitz@gatech.edu

ABSTRACT
Through this project a working prototype of a language learning application that implements best practices in language learning along with an intelligent tutoring system (to maximize learning effectiveness) was developed. Prior to the development of this prototype the author was unable to find existing language learning applications that implemented intelligent tutoring systems.

The original goal of this project was to develop a language learning tool that implemented both an intelligent tutoring system and best practices from game-based learning (gamification).

Author Keywords
ITS; Intelligent Tutoring Systems; SRS; Spaced Repetition System; Flash cards; Education; Ed Tech.

ACM Classification Keywords
K.3.m. Computers and Education Miscellaneous

INTRODUCTION
The author was unable to find a gamified ITS implementation for language learning. This paper catalogs the endeavor to build one such system and describes the prototype created.

The created prototype is built using an existing open-source language learning application as a starting point. The open-source app is called "Learn With Text". Basic intelligent tutoring system features were added to this app to enhance learning effectiveness. One such feature is word suggestions. When a student doesn't know a word a suggestion (usually a mnemonic) is given to help them learn the term. Another such feature is word dependency. If a student doesn't know a basic word that acts as a component of another word or a sentence, that sentence cannot be learned.

The author was unable to implement game-based learning techniques within the limited timeframe of this project.

RELATED WORK
The author focuses on three key terms from literature and industry, which are defined below: Intelligent Tutoring Systems (ITS), Game-based learning, Spaced Repetition Systems (SRS).

Intelligent Tutoring Systems:

Intelligent tutoring systems ("any computer program that contains some intelligence and can be used in learning" [1]) apply intelligence to learning by providing prompt and customized feedback to learners [10]. "Research on prototype systems indicate that ITS-taught students generally learn faster and translate the learning into improved performance better than classroom-trained participants." [8]. Another study found statistically significant evidence of the impact that an intelligent tutoring system can play in the classroom setting [6].

Game-based Learning:

"Game-based learning refers to the borrowing of certain gaming principles and applying them to real-life settings to engage users" [9]. 'Gamification' is the application of game-based concepts (like using points, etc) to non-games.

In one study, students who learned with an educational game performed on average 20% better on a post learning test than the control group. Additionally, students respond resoundingly positively to game-based learning [3].

Spaced Repetition Systems:

"The spacing effect is the observation that people tend to remember things more effectively if they use spaced repetition practice (short study periods spread out over time) as opposed to massed practice (i.e., "cramming"))" [11]. "Across 254 studies comparing massed versus spaced practice on later memory for verbal information (e.g., words, sentences, facts, passages), overall, spaced practice dominated massed practice in recall performance" [5]. Spaced repetition systems implement the spaced practice concept.

Related Work Across These Categories:

Applications combining Intelligent Tutoring systems with games can keep learners engaged (gaming aspects) and maximize the learning effectiveness (ITS aspects) of these educational materials [7]. Most popular ITS implementations are focused on math and science education. There are few ITS implementations in the language learning space. Further, there are few game-based intelligent tutoring systems that are focused on language learning. One such system developed in an academic setting describes the necessity for frequent design assessments to be performed [4].

There are many popular SRS-based language learning systems. DuoLingo (the most popular language learning application[11]) and Anki (another popular language

learning application [2]) both implement concepts from spaced repetition practice.

THE PROBLEM

Little work has been done on combining intelligent tutoring and spaced repetition systems, particularly in the field of language learning.

If research consistently shows that spaced repetition and intelligent tutoring systems each improve learning effectiveness, then combining them in a single application should further promote increased learning.

Additionally, research consistently shows that game-based learning increases learning engagement and contributes to increased learning.

Combining SRS, ITS, and game-based learning methods should increase learning effectiveness and engagement.

The author was unable to find a gamified ITS implementation for language learning.

THE PROJECT PLAN

The author set out to solve this problem by developing a new ITS implementation for language learning. With well understood time constraints, the author developed a project plan that entailed identifying existing software tools (a language learning platform and an ITS system) and combining them, leaving the addition of game-based learning for future work.

DEVELOPMENT

The language learning prototype developed by the author began with the open-source language learning application "Learn With Text". "Learn With Text" is written in PHP with some components in JavaScript and a MySQL backend to store the data.

After researching various open source intelligent tutoring systems (ITS), the author was unable to find a suitable one to implement and opted to develop a basic version in-house.

Originally, the language learning application did not have a system for configuring users.

The Data Model:

A functional intelligent tutoring system needs a **student model**, a **domain model**, and a **teaching model**. These pieces needed to be added. The diagram in Figure 1 shows the data model of the upgraded learning tool.

The *student* table stores information about each student.

The *studentwordstatus* table stores the learning-related information for each student and word combination (including 'status' described above).

The *words_new* table stores only word-related information.

The *wordsuggestions* table stores suggestions for each word. These suggestions are presented when mistakes are made within the user interface.

The *worddep* table stores word dependencies. These dependencies are used to prohibit challenging words from being displayed until component words have been learned.

These tables make up the student, domain, and teaching models.

User Interface Changes:

A user selection system was added to the user interface. A simple drop-down box is used to select the current user – this is not secure, but for this prototype it should be sufficient.

Functionality Changes:

The original lesson system was adapted to use the new data model.

Basic ITS features were implemented. When a student doesn't know a word a random suggestion from the *wordsuggestions* table is given, if available. Words in the *worddep* table are locked for learning until all of the independent words have been learned.

TRAINING PLANS

Detailed below are training plans created for 3 students. The language used for these training plans is Japanese.

Student 1:

This student gets the most basic hiragana characters あ、い、う、え、お (a, i, u, e, o) correct, but is incorrect on the next set of hiragana characters: か、き、く、け、こ

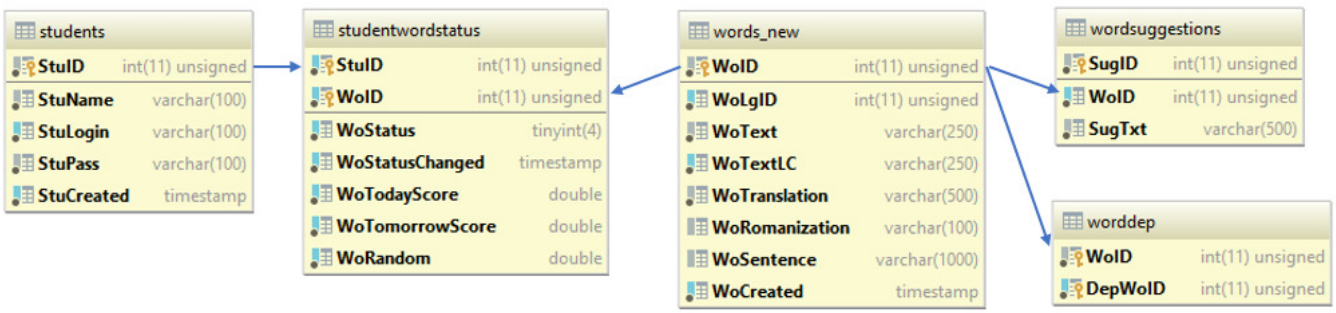


Figure 1 Data Model

(ka, ki, ku, ke, ko).

The ITS suggests mnemonics for each of the incorrectly answered characters.

Student 2:

This student gets all of the katakana characters correct except for one: メ (me)

The student then tries to learn words spelled in katakana that do not contain the character. The system should not interfere.

The student then tries to learn a word spelled in katakana アメリカ (America) that does contain the character. The intelligent teacher prevents the student against learning more complex words that build upon fundamental learnings that the student hasn't mastered.

Student 3:

This student gets all of the hiragana characters correct. This student then tries to learn kanji characters. The student gets some kanji characters incorrect. The ITS should suggests mnemonics for the incorrectly answered kanji characters.

The student then tries to learn words using the kanji characters. The ITS prevents the student from learning these words before the kanji characters have been mastered.

APPLICATION DESIGN

The final version of this tool can be seen in the images below.

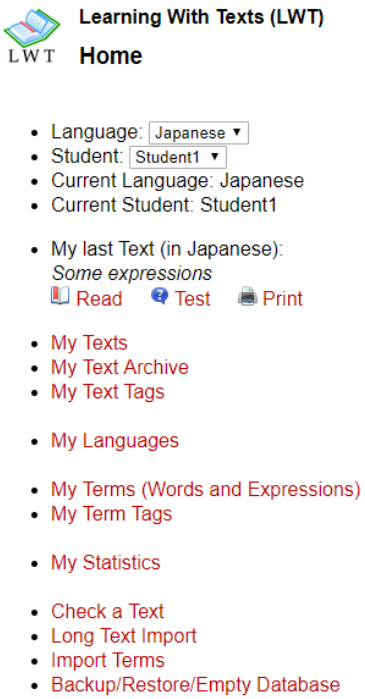


Figure 2 LWT Home/Start Screen

Figure 2 above shows the home/start screen for the application. The home screen allows the user to select the language, 'login' (by selecting themselves) and determine the next course of action:

1. Learn/study new words
2. Test on the existing set of words
3. Change settings

Learning

If the user opts to learn/study new words, Figure 3 below is what the user will see.

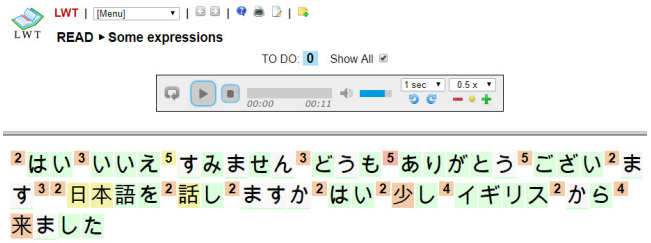


Figure 3 Learn New Words

The user is able to click each individual letter or the number next to each letter (2 in first example) to see details about the word.

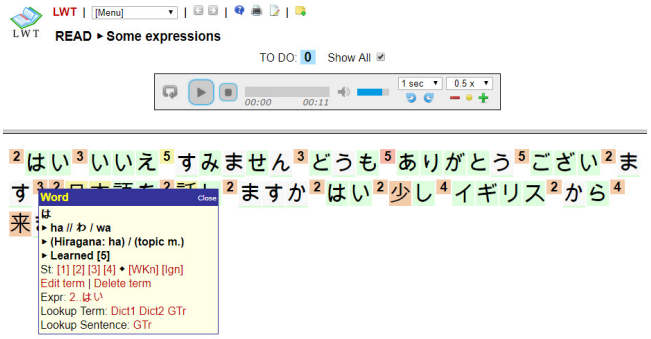


Figure 4 Selecting a Single Character

If the user selects a single character, they see its description in a box and are asked to input the 'status' (1 out of 5, 5 being best) of their knowledge of the word.

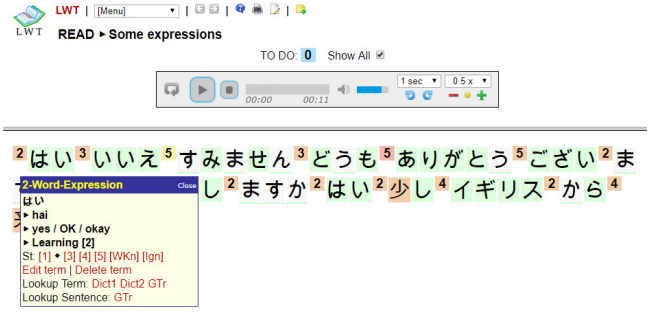


Figure 5 Selecting a Multi-character Word

If the user selects the number next to a word, they see the description of that word and are asked to input the same learning status.

Testing

When selecting the "Test" button the image displayed in Figure 6 below is presented.

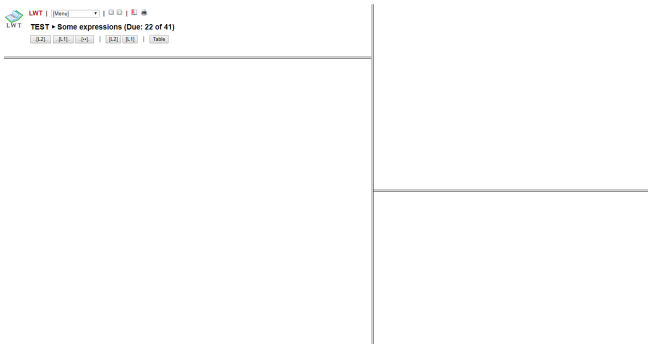


Figure 6 Initial Test Screen

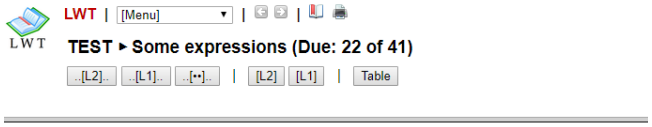
Zooming in on the filled portion of the screen:

Figure 7 Test Screen Header

As Figure 7 shows, it isn't particularly clear what these option buttons mean within the context of testing.

The first 3 buttons "..[L2]..", "..[L1]..", and "..[**].." all show an example sentence with the word being tested in an identical position but presented differently.

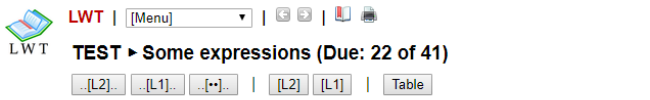
"..[L2].." shows the word in the target language, as in Figure 8 below:



あ り がとうございます。

Figure 8 Test Word in Target Language

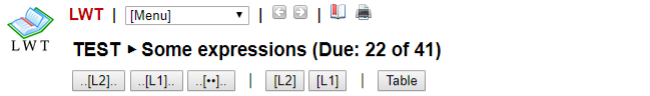
"..[L1].." shows the word's translation, as in Figure 9 below:



いい [(Hiragana: e)] 。

Figure 9 Test Word's Translation

"..[**].." shows the sentence without the word, as in Figure 10 below:



あ [•] がとうございます。

Figure 10 Sentence Without Test Word

The next set of buttons, "[L2]" and "[L1]" show the word in each form (target language for L2 and translation for L1) without an accompanying sentence as shown in figures 11 and 12 below.

り

Figure 11 Test Word in Target Language

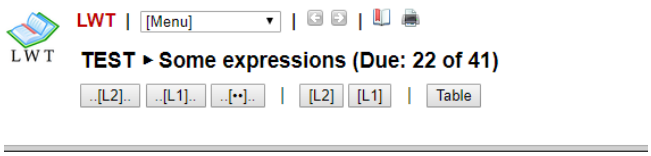


Figure 12 Test Word Translation

Allowing the user to test their memory in both directions (starting from the target language to test memory of the translation and starting from the translation to test memory of the word in the target language) allows the learner to understand more about their current learning level.

The final button, "Table" shows a table of each word in the dictionary, allowing the user to see their current learning status. This can be seen in figure 13 below.

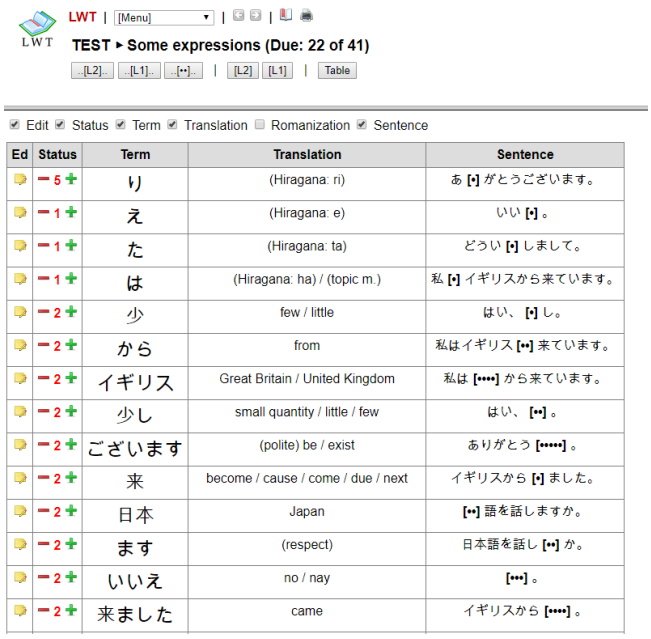
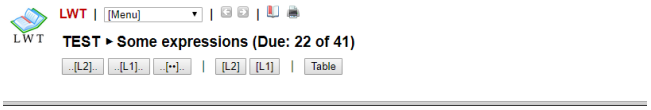


Figure 13 Test Table

No testing can actually take place here, but there is the option to adjust the current learning status for each word.

In any of the above testing screens (not including the table screen), the user simply clicks on the orange box to update the current learning status.

The screenshot in figure 14 below contains an example of the update window that appears on the "..[L2].." page.



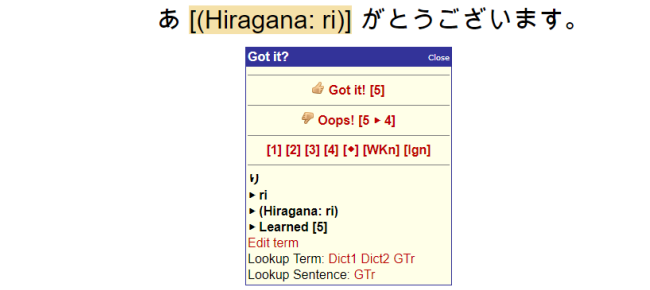


Figure 14 Test Word Answer

The user is shown the correct translation in place of the original target language word and is asked whether they translated it correctly. If so, they should click the "Got it! [5]" button. If not, they should click the "Oops! [5 -> 4]" button. These buttons adjust the learning status described above.

When the user clicks either "Got it!" or "Oops!" they're presented with a new word, but before that, they're presented with a brief confirmation message in the top-right corner of the screen, as shown in Figure 15 below.

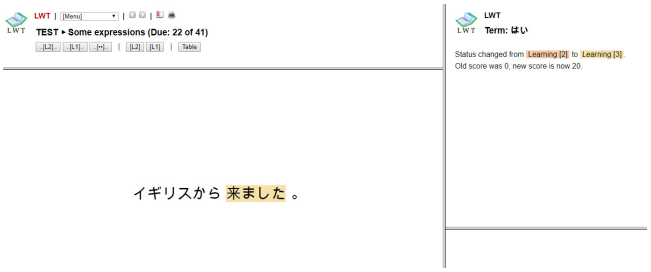


Figure 15 Answering A Question

Zooming in on the text in Figure 16 below we see that the message identifies that the learning status has been changed. If the question was answered incorrectly we might see a learning suggestion given by the ITS system.

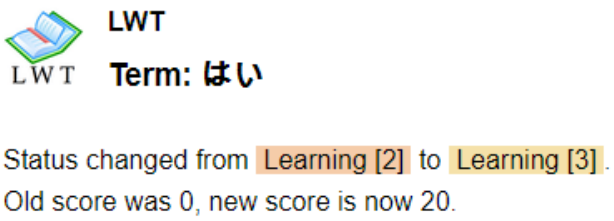


Figure 16 Answering A Question

CONCLUSION

The final version of this product should contain elements of the 3 pieces described above: intelligent tutoring systems, game-based learning, and spaced repetition systems.

The result of the project to-date is a prototype of the full language learning application described above. The prototype has limited features and learning material and is not ready for live users. The purpose of this prototype is to act as a proof of concept that can be used for further testing against the assumption that combining these systems will enhance learning effectiveness and student engagement.

The developed prototype confirms that it is possible to develop language learning tools that incorporate intelligent tutoring system and spaced repetition system features to increase learning effectiveness. Learning effectiveness was not tested with this tool, but prior studies support the hypothesis.

FUTURE WORK

As described above, the development work completed thus far is only an introductory prototype of the full project envisioned by the author. Further would see the development of gamified features, additional ITS features, and making the existing features release-ready.

Gamified Features:

The release version of this tool should have features that increase engagement. The best way to implement these features has yet to be seen, but a popular approach includes giving students points (within the application) for successes allowing them to compete with themselves and others to maximize points. Additional ideas include adding a narrative story, adding more competitive elements, or adding time-delay features.

Additional ITS Features:

In addition to the basic ITS features currently implemented, new, more meaningful, but more complex, ITS features could be added to the benefit of students.

Release Readiness:

The existing application is not ready for deployment or use by real users. It has only basic language information (a student could learn all of the information in the database, but still only speak a few simple words and sentences). It lacks user login features. It lacks any graphic design. All of these issues would need to be resolved before the application is ready for use.

ACKNOWLEDGMENTS

would like to thank my mentors Tammy Preuss and Lingxi Xiao for their support and counsel throughout this semester and on this project.

REFERENCES

1. Freedman, R., Ali, S. S., & McRoy, S. (2000). Links: what is an intelligent tutoring system?. *intelligence, 11*(3), 15-16.
2. Godwin-Jones, R. (2011). Mobile apps for language learning. *Language Learning & Technology, 15*(2), 2-11.
3. Jenkins, H., Klopfer, E., Squire, K., & Tan, P. (2003). Entering the education arcade. *Computers in Entertainment (CIE), 1*(1), 8.
4. Johnson, W. L., & Beal, C. R. (2005, May). Iterative Evaluation of a Large-Scale, Intelligent Game for Language Learning. In *AIED* (pp. 290-297).
5. Kang, S. H. (2016). Spaced repetition promotes efficient and effective learning: policy implications for instruction. *Policy Insights from the Behavioral and Brain Sciences, 3*(1), 12-19.
6. Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education (IJAIED),* 8, 30-43.
7. McNAMARA, D. S., JACKSON, G. T., & GRAESSER, A. (2009). Intelligent Tutoring and Games (ITaG). In AIED 2009: 14 th International Conference on Artificial Intelligence in Education Workshops Proceedings (p. 1).
8. Ong, J., Ramachandran, S., (2000). Intelligent Tutoring Systems: The What and the How. *American Society for Training & Development.*
9. Pho, A., & Dinscore, A. (2015). Game-based learning. *Tips and Trends.*
10. Psotka, J., Massey, L. D., & Mutter, S. A. (Eds.). (1988). *Intelligent tutoring systems: Lessons learned*. Psychology Press.
11. Settles, B., & Meeder, B. (2016). A trainable spaced repetition model for language learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Vol. 1, pp. 1848-1858).