2016

Artificially Intelligent Computer Assisted Language Learning System With AI Student Component

Denee M. McClain
Southern Adventist University, deneemcclain@southern.edu

Artificially Intelligent Computer Assisted Language Learning System With AI Student Component

Denee McClain

***Abstract*—Intelligent Computer Assisted Language Learning (ICALL) systems follow an accepted format, which utilizes an artificially intelligent tutor. The systems allow the user to input a sentence in the target language and the AI tutor analyzes the sentence and provides error correction. This approach can be expensive, impractical, and inflexible. Inflexibility can result in a lower quality of learning for the users of these systems. Here I present an alternative format for ICALL systems that utilizes an artificially intelligent student. This alternative is cost effective and practical because it does not require extra development time to make the artificial intelligence an expert on the language. Flipping the roles of tutor and student allows the user to focus on the basic concepts of the target language and allows the user to learn by teaching, which are critically acclaimed methods of learning.**

I. Introduction

Language learning software development is a multi-million dollar field with plenty of room for growth and innovation. According to IBISWorld, language learning software development brings in $600 million in revenue. This field is currently growing and is projected to continue growing in the future. There are only a few key players within this industry and barrier to entry is declining, thus there is ample opportunity for experimentation. Innovation is an integral part of this industry, as the rate of technology change is high [11]. Fresh ideas for software in this field can lead to increased growth and revenue. It is thus beneficial to explore new opportunities in language learning software development.

There are several types of language learning software. A type of software that is less often used but has high potential is Intelligent Computer Assisted Language Learning (ICALL) software. ICALL systems combine language practice with artificial intelligence. An AI tutor asks the user questions, and the user responds with input. This is not a new idea. The tutor provides most of its assistance through error correction and navigational support, as well as feedback for the user. Unfortunately, the AI necessary to implement this approach must be limited in order to control costs and to make development feasible. This results in a restricted learning experience for the user [6] [8]. This paper presents an improvement upon this approach that will forgo the limitations of Intelligent Tutoring Systems by utilizing an AI student instead of an AI tutor. In this paper, I will refer to the proposed system as the *Language Student*. The user will provide information to the system, from which it will learn more about the language the user is learning. The user will provide error correction to the system, and the system will gain the ability to "speak" in the user's target language. This system employs a concept similar to that of authoring systems; however, the difference is a more thorough incorporation of artificial intelligence [10]. This system places a greater emphasis on how language is naturally learned [14]. The user will enjoy a learning experience free of input and feedback restrictions. The system proposed here may be used in integration with current ICALL systems in order to provide the aspects which these systems lack.

II. Background

The next section describes the foundation on which most Intelligent Computer Assisted Language Learning systems are built and the issues with this foundation. In the following section, the second language learning theories that form an alternative foundation for the ICALL system presented in this paper are discussed.

A. Intelligent Tutoring Systems and Their Limitations

Computer Assisted Language Learning (CALL) is a field that has advanced throughout the years ever since its origin in the 1960s. As researchers learned more about the capabilities of computers and of various teaching methods, improvements and additions were made to the methods used within the field. Originating with a simple stimulus and response format, CALL developers now utilize the Internet, multimedia and Artificial Intelligence to produce error analysis and feedback modules that are more accurate. Examples of such systems include the *German Tutor* developed by Heift and Nicholson [16] and the Web Passive Voice Tutor detailed by Virvou and Tsiriga [12]. The field has explored various approaches to teaching a second language, including the communicative approach, the drill-based approach, and the explorative approach [5]. Branching off from this field, Intelligent Computer Assisted Language Learning (ICALL) often combines artificial intelligence with a teacher-centered approach to second language learning [16], [12]. Just as CALL has benefited from new ideas and new approaches, ICALL can benefit from a new method for integrating artificial intelligence into systems.

The format of most Intelligent Computer Assisted Language Learning systems is based off of that of Intelligent Tutoring Systems. Intelligent Tutoring Systems are defined by Paviotti as computer-based support for educational activities, an application of AI to education [6]. Students most often use these systems to complete practice exercises or work through problems. They submit answers and the system provides error correction and feedback. The system may provide hints to help the student come to the correct answer. ITSs may also aid the student by providing navigational support, prompting the user to go on a specific module based off of how well they have completed the others [10].

Intelligent Tutoring Systems have proved extremely useful. They are most often used in classrooms, and have been shown to improve learning when used in addition with other traditional classroom techniques [9] [6]. However, there are some problems that prevent the widespread use of Intelligent Tutoring Systems. These systems are very expensive, and their cost limits their development and their deployment. In addition to being expensive, ITSs require a large amount of time to develop. According to Paviotti, for every hour of product produced, several hundred hours of work are put in. This time commitment makes the development of new ITSs a difficult task to take on. The time and expense required to develop ITSs has consequences. Oftentimes, these systems are used for research or for a small section of the population because they do not scale up easily to real world use. There are systems that are used extensively, but they are few [6].

The limitations of Intelligent Tutoring Systems extend to their effectiveness. Although these systems do improve learning, there are issues that lower the quality of learning for the users. One common issue is that ITSs have ”limited categories of problems”. Thus, a limited number of users will truly benefit from a system. The low variety in problems contributes to the fact that complex problems are often unable to be practiced using these systems, another downside. Even if the available problems are tailored to the user's needs, another common issue is that the error correction and feedback associated with the given problem may be too broad or too specific [6]. Broad feedback fails to uncover the reason for the incorrect answer given by the user. Feedback that is too specific may be unhelpful to a user who still needs to understand the concept from a broader point of view. Lantolf, an expert in the field of second language learning, explains that ”for corrective feedback to be effective it must be sensitive to the individual learner's ZPD”, or zone of proximal development. A student's ZPD is a combination of his ”past learning and future development” [7]. Some ITSs do indeed take past learning into consideration when providing navigational support, but past learning is not reflected in the feedback. This limits the effectiveness of the system.

The intelligence of Intelligent Tutoring Systems is another limitation. Karlstrom et al. maintain that artificial general intelligence is a field that has not quite reached expectations [13]. Because of this, the ITSs developed can only be as efficient or as intelligent as current efforts within artificial intelligence. The systems are prone to efficiency and accuracy issues. Students make errors for different reasons, yet the system can only guess why a student made a specific error. Oftentimes the system can guess correctly, but when the system guesses incorrectly, then the student misses out on the opportunity for deeper learning. In addition, students may make errors that the system has not been written to handle, sometimes called ”non pertinent” errors. These errors, which can take a number of different forms, can seriously affect the accuracy of a language learning system [18] [13]. German Tutor, an ITS system described in [16], consistently achieves a high accuracy rating, but only because user input is restricted to a select few allowable words and sentences. Systems without this kind of restriction are error prone, and few authors bring up the error rates or the accuracy rates of their systems because of this circumstance.

The final limitation that Intelligent Tutoring Systems run into is described by Koedinger and Aleven as the assistance dilemma [8]. This is the issue of finding a balance between giving information and withholding information when providing a student with feedback. Too much assistance might be given and can lead to such issues as students ”gaming” the system, or requesting enough hints to quickly finish the exercise or problem the system has presented without deep learning taking place [6]. This balance is difficult to find. Koedinger and Aleven maintain that feedback that provides explanations is better for student's learning than feedback that gives simple yes/no answers. However, immediate feedback may also be detrimental to the learning experience. Koedinger and Aleven suggest the use of what they call ”intelligent novices”, AI tutors that do not catch every error the student makes. Allowing the system to miss an error can allow the student to catch it for himself or see and understand the consequences of the mistake in the exercise results. This allows for deeper, natural learning. Over time, error detection will become more accurate and will be able to detect more complex errors [8].

These limitations and proposed solutions are important for ICALL systems because their traditional format is very similar to that of Intelligent Tutoring Systems. The *Language Student* proposed in this paper aims to avoid limitations of cost and development time by incorporating the solution proposed by Koedinger and Aleven and foregoing the attempt to make the AI into an expert on the language. Rather, the system will learn with the user. This approach, detailed below, further avoids the limitations above by widening the scope of practice problems and eliminating the possibility of ”gaming” the system. The learning of the student is no longer dependent on the expertise of the artificial intelligence. Finally, as will be described in the following section, the design of the *Language Student* takes into consideration several current and accepted theories of second language learning. What results is a unique system that can provide a higher quality of learning than that of traditional systems.

B. Second Language Learning

The *Language Student* incorporates current Second Language Learning theories into its design. These theories differ from those reflected in AI-tutor-based ICALL systems. These more traditional ICALL systems and their user input and system feedback model mimic the stimulus, response, and reinforcement process, or S-R-R approach, described by Malone [3]. This approach dates back to the 20^{th} century. Some of the newer and older Second Language Learning research is discussed below.

The S-R-R approach, which lends well to the traditional ICALL system design, stems from a theory called Behaviorism. This theory arose in the middle of the 20^{th} and is most often associated with psychologist and philosopher B.F. Skinner. He believed that any human could learn anything using the three-step process of providing stimulus, receiving a response, and giving positive or negative feedback. Behaviorists since Skinner have advocated that this process can be used to teach second languages [3]. The combination of the S-R-R approach and Second Language Learning resulted in the audio-lingual method, in which a teacher speaks aloud a sentence in the target language, the student repeats, and feedback is given.

This method lost popularity in the 1950s as more research was done into how the brain retains language information [2]. Newer theories take other factors of language learning into account.

Another popular theory pioneered by Stephen Krashen is the Natural Approach. His research culminates in five major hypotheses. The fourth one is most important to the design of the *Language Student*. This theory states that the only thing that is necessary for second language acquisition is ”meaningful input”, or new words and phrases in the second language [3]. Meaningful input is input that is just beyond the current level of understanding in the target language. As long as meaningful input is continually given to the student, learning will take place, according to Krashen. He explains that ”if we provide students with enough comprehensible input, the structures they are ready to acquire will be present in the input”. He insists that ”all that is necessary for language acquisition is input that is interesting and comprehensible” [15]. This theory is reflected in the *Language Student*, albeit in the opposite direction. The user continually inputs vocabulary and grammar concepts into the system. From these simple atomic inputs, the AI student gains an understanding of the second language. This emphasizes the concept of learning by teaching, which is called one of the best ways to learn a language by Adrienne Royo, Ph.D., provides the conclusion that the user will come to a better understanding of the language by providing the simple atomic inputs to the *Language Student*.

Another popular second language learning theory, the sociocultural theory, also comes into play in the design of the *Language Student*. This theory proposes that incorporating social interaction into the second language acquisition process will improve learning. In the traditional classroom, this is accomplished by allowing students to interact with each other and with other speakers of the target language. Malone explains that ”when learners talk in the L2 [target language] they notice a...difference between their knowledge of the L2 and what they want to...say”. In order to remedy this gap, the student must think about the language constructs in order to form a sentence that conveys the desired message. The student is then engaging in a metalinguistic activity, one which results in a more robust understanding of the target language [3]. In an ICALL system, this can be accomplished by allowing the user to interact with the system, similar to the way in which the user interacts with the *Language Student*, again in the opposite direction. When the *Language Student* produces a malformed sentence, the user must think about the target language in order to provide grammar concepts that allow the system to convey the desired message. This sort of ”collaborative revision” is shown by researchers to be beneficial. In fact, Lantolf explains that studies show that ”asymmetrical interactions, in which one of the members of a given dyad is able to control the performance of the other member, may be more conducive to learning” [7]. This is just the sort of environment that the *Language Student* provides. Research suggests that this system may prove very helpful for its users.

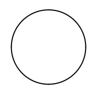
These critically acclaimed theories and the learning by teaching method are combined in the *Language Student*. The following section details the design of this system.

III. Description Of Proposal

The *Language Student* that I am proposing has three main components - a system tracking component, a communication component, and a comprehension component.

System Tracking	User name StudentScore User Vocabulary User Vocabulary Classifications User Grammar Explanations

Table I
Each of the above items are stored within a learner profile, which is stored in the System Tracking component of the *Language Student*



A. System Tracking

The system tracking component of the proposed system will follow the example of many current ICALL systems [16]. Each user will have a learner profile stored in the system's database. The information stored within a learner profile can be seen in table I. The StudentScore is a numerical representation of the *Language Student*'s level, calculated by multiplying the overall rate of correct sentences formed by the number of vocabulary items the *Language Student* contains for that particular user's target language. This number, which combines the skill level of the *Language Student* at grammar with the amount of vocabulary the *Language Student* knows, is a comprehensive representation of how well a particular user is teaching the *Language Student* the language. The user vocabulary will be entered by the user, and will be kept track of separate from the vocabulary of other users. The user vocabulary classifications and grammar explanations, which will be explained in more detail below, allow the user more freedom in teaching the *Language Student*.

B. Communication

The communication component is the interface through which the user interacts with the *Language Student*. The user may interact with the *Language Student* in three different ways. The initial method of interaction is the most simple and only needs to happen once per user. The user simply enters into a form their user name and any other necessary identifying information, including the target language of the user. The other possible methods of interaction with the system are focused on teaching the *Language Student* the target language. The second possible method of interaction feeds the system vocabulary. The student may input any vocabulary word into the system along with its classification. The possible vocabulary classifications are as follows [4] [17]:

1) Nouns
2) Pronouns
3) Infinitive verb
4) Action verb
5) Helper Verb
6) Linking verb
7) Gerund
8) Adjective
9) Adverb
10) Preposition
11) Conjunction

In addition to the possible classifications, a user may define his or her own classification for the word. This classification would persist within the system tracking component for that student, as previously noted. Any vocabulary word placed within one of the above classifications would be then passed to the comprehension component of the system.

The second method of interaction with the *Language Student* allows the user to teach the system grammar. The communication component will prompt the comprehension component to formulate a new sentence, which the communication component will receive and display to the user. The sentence displayed will be formed from the vocabulary words already inputted into the system. If the system does not have enough vocabulary words to create a sentence, then an error message will be displayed to the user. The user will read the sentence the system displays and then determine whether it is a well-formed sentence or a malformed sentence. If the sentence is well-formed, the user will select the appropriate feedback option on screen. If the sentence is malformed, then the user will select the appropriate feedback option and answer a follow up question. The follow up question will always ask the user why the sentence was malformed. The user will be able to enter any description of the error in the malformed sentence. Following this, the user will be allowed to enter a well-formed version of the incorrect sentence originally presented by the system. The feedback will be passed to the comprehension component of the system.

The preceding paragraphs lay out the features for a simple user interface with the *Language Student*. The simplicity and ease of use of the proposed system is a benefit that will widen the appeal for students of a second language.

C. Comprehension

The comprehension component of the *Language Student* takes input from the communication component and may choose from three options of what to do with the information. It either stores the data in its database, intelligently analyzes the data, and/or passes some or all of the information to the system tracking component so that it can be updated. If the input from the user is simple identifying information, then the *Language Student* will update the system tracking component. If the input from the user is new vocabulary words, then the system will add the new words to its vocabulary database and update the system tracking component with any new user vocabulary classifications. If the input from the user is positive feedback in response to a sentence presented by the *Language Student*, then the system will update the StudentScore object in the system tracking component. Finally, if the input from the user is negative feedback in response to a sentence presented by the *Language Student* along with the reasoning for why the sentence was incorrect and a correct version of the sentence, then the system will update the StudentScore object and the user grammar explanations in the system tracking component. The comprehension component will also analyze the corrected sentence from the user, the original sentence displayed to the user, and the grammar explanation. Finally, the comprehension component drafts new sentences to be presented to the user. The details of the sentence analysis and sentence creation follow.

1) Grammar Analysis: In order to correctly link the sentence corrections to the grammar explanation given, the *Language Student*'s comprehension component will match the originally presented sentence to the corrected sentence. The comprehension component will determine the number and type of differences between the original sentence and the new sentence and remember what differences are associated with which grammar explanations. For example, if the difference between one sentence and another sentence is the ordering of the subject and a verb, the *Language Student* will store the grammar explanation given by the user with the instructions necessary to switch the order of the subject and the verb. These explanations and the instructions paired with them will be used again when the comprehension component needs to draft a new sentence, which process is explained next.

2) Sentence Creation: When prompted, assuming there are enough vocabulary words, the *Language Student*'s comprehension component will create a new sentence. If there are no grammar explanations, i.e., if this is the first time the *Language Student* is creating a sentence from the user's vocabulary, then the comprehension component will choose at least one word from the noun or pronoun classification and at least one word from the action verb classification. Next, the system will randomly decide to choose zero or more words from zero or more of the other vocabulary classifications. After choosing the words, the comprehension component will choose a random order in which to place the words. The system will use part of speech tagging [16] in order to keep track of which vocabulary classification appears where in the sentence. The comprehension component will then pass the new sentence to the communication component, which will present it to the user.

If grammar explanations do exist for the specific user requesting the sentence, then the *Language Student*'s comprehension component will go through the process of randomly arranging a sentence just as if there were no grammar explanations. Next, the comprehension component will request the list of grammar explanations from the system tracking component and iterate through each grammar explanation and its associated instruction. The *Language Student* will apply each appropriate instruction to the sentence. An instruction is appropriate if the parts of speech the instruction applies to are present in the sentence. If the instruction is inappropriate, it will be ignored. After iterating through each instruction, the comprehension component will pass the new sentence to the communication component, which will present it to the user.

D. System Architecture

In order to fully implement the *Language Student*, a client-server architecture would be used [16]. The server would store a collection of databases, which in turn will store learner profiles, vocabulary, and grammar explanations. In other words, the server would implement the system tracking component. The client would handle input and output, grammar analysis, and sentence creation, the duties of the communication and comprehension components of the *Language Student*.

E. Future Work

The first step in future work on the *Language Student* that will follow development is testing. Testing the *Language Student* on a variety of second language learners will provide quantitative information about the system's level of effectiveness. Because the system simply stores information about a language, such as the vocabulary or the instructions that are associated with grammar explanations, the system is language independent, so testing on individuals of different target languages will be possible.

In addition to testing over speakers of various native languages and various target languages, testing can be done over speakers with various levels of skill in their target language. Using the *Language Student* may have better results for individuals who are at an intermediate level in their target language. This is because of the system's dependence on the individual being able to teach their language to the system via vocabulary and grammar explanations.

A possible methodology might proceed as follows. Test each subject on their proficiency in their target language. Next, allow each subject to teach their target language to the *Language Student* every day for a set amount of weeks and a period of time each day. At the end of the given number of weeks, test each subject again and then compare the scores from the two tests. An important aspect of this experiment will be the test given to each student. Testing over a variety of language learning areas will allow the experiment to capture which language area use of the *Language Student* enhances.

IV. Conclusion

This paper has presented a solution to the problem of the expense and impracticality of artificially intelligent tutors in the field of Intelligent Computer Assisted Language Learning. In order to avoid these difficulties as well as focus on the best methods for second language learning, the *Language Student* acts as an artificially intelligent student for the user. The user teaches the *Language Student* the target language via the communication component. The *Language Student* uses its comprehension component to learn how to construct sentences in the target language, and its system tracking component retains information about the vocabulary and grammar of the target language. Use of this system alone as well as integration with current ICALL systems may add a unique and useful side to the CALL field of computing.

References

[1] Adrienne Royo, Dean of Modern Languages Department, Southern Adventist University. Interview. Collegedale, TN. February 29, 2016.

[2] Behaviorism. [Online]. Available http://www.tesolclass.com/applying-sla-theories/behaviorism/.

[3] D.Malone. (2012). Theories and Research of Second Language Acquisition. [Online]. Available: http://www.sil.org/sites/default/files/files/theories_and_research_of_second_language_acquisition.pdf.

[4] *Ginger Software*. [Online]. Available: http://www.gingersoftware.com/content/grammar-rules/nouns/gerunds-infinitives/

[5] G. Davies, "CALL (computer assisted language learning)," *Centre for Languages, Linguistics, and Area Studies*, Web, Available: https://www.llas.ac.uk/resources/gpg/61

[6] G. Paviotti et al., "Intelligent Tutoring System: a short History and New Challenges", in Intelligent Tutoring Systems: An Overview, Italy: Pensa Multimedia, 2012, ch. 1, pp. 13-57.

[7] J.P. Lantolf, "Sociocultural Theory and Second Language Learning" in *The Modern Language Journal*, vol.78, pp.418-420.

[8] K.R. Koedinger and V. Aleven, "Exploring the Assistance Dilemma in Experiments with Cognitive Tutors", *Educational Psychology Review*, vol. 19, no. 3, pp. 239-264, 2007.

[9] K. VanLehn, "The behavior of tutoring systems", *International Journal of Artificial Intelligence in Education*, vol. 16, pp. 227-265, 2006

[10] M.A. Emran and K. Shaalan, "A Survey of Intelligent Language Tutoring Systems", in *Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on)*, 2014, pp.393-399.

[11] M. Soshkin. (2015, Dec).*Language Learning Software Developers.* [Online]. Available: http://www.ibisworld.com/industry/language-learning-software-developers.html.

[12] M. Virvou and V. Tsiriga, "Web Passive Voice Tutor: an intelligent computer assisted language learning system over the WWW," in *Advanced Learning Technologies, 2001. Proceedings. IEEE International Conference on*, 2001, pp.131-134.

[13] P. Karlstrom, T. C. Pargman and R. Ramberg, "Designing for collaboration in intelligent computer assisted language learning," Advanced Learning Technologies, 2005. ICALT 2005. Fifth IEEE International Conference on, 2005, pp. 321-322.

[14] R.A. Farmer and B. Hughes, "CASE: A Framework for Evaluation Learner-Computer Interaction in Computer Assisted Language Learning", in *Proceedings of the 6th ACM SIGCHI New Zealand chapter's international conference on Computer-human interaction: making CHI natural (CHINZ '05)*, New York, NY, pp.67-74.

[15] S. Krashen. "Language Acquisition and Application" in *Second Language Acquisition*. Mexico City, Mexico: Cambridge University, 2013, ch.1, sec.1, pp.3-4.

[16] T. Heift and D. Nicholson, "Theoretical and practical considerations for web-based intelligent language tutoring systems", in *Intelligent Tutoring Systems*, 2000, pp. 354-362.

[17] Types of Verbs. (2013, June). *Utah Valley University*. Web. Available: https://www.uvu.edu/writingcenter/docs/handouts/typesofverbs.pdf

[18] Y. Futagi, "The Effects of Learner Errors on the Development of a Collocation Detection Tool", in *Analytics for noisy unstructured text data, Proceedings of the fourth workshop on*, 2010, pp.27-34.