THE IMPLICATIONS OF ITS RESEARCH ON CALL SOFTWARE

Laura Vawter, Nikolaj Troels Graf von Malotky, Alke Martens

University Rostock (GERMANY)

Abstract

Current research into Computer Assisted Language Learning (CALL) software demonstrates an existence of cognitive, constructivist and behaviorist learning styles. Within these systems exist a tendency towards a behaviorist learning style of content creation and user interaction and evaluation. This tendency often leads to a structuring of tasks in a drill and practice format, and the interaction between the user and the system involves the administration of rewards and punishments for the solidification of user knowledge. However, research into Intelligent Tutoring System (ITS) reveals that effective ITSs utilize domain, learner and pedagogical knowledge in their interaction with the user. The implementations of these knowledge forms in ITSs are greater user retention of content. Subsequently, the implementation of learner and pedagogical knowledge provides better individual-focused instruction and a more accurate system model of user knowledge. Furthermore, the evaluation of ITSs based on the existence of these knowledge forms provides consistency for developers and content creators in the implementation of updates and innovations in current ITSs. In contrast, the evaluation and construction of current CALL systems frequently centers more on tasks creation in the system and less on the interaction of the system with the user. A pure ITS system is costly and time consuming to develop, thus this paper argues for a restructuring of CALL systems with the distinction between these knowledge forms within ITSs.

Domain, learner and pedagogical knowledge are often perceived differently from a linguistic perspective. This paper explores the definitions of these knowledge forms in relation to the linguistic concepts of native language (L1) and target language (L2) interdependency and influence on system error analysis. In this paper we investigate some of the current CALL software including Duolingo, Rosetta Stone and Memrise. In our analysis we identify domain, learner and pedagogical knowledge in these systems. We found that domain knowledge was easily identifiable, but the manifestation of learner and pedagogical knowledge was rudimentary at best.

Moreover, this paper examines task construction with learner and pedagogical knowledge and gives recommendations for improvements within these CALL systems. These improvements are related to system feedback, content construction, content sequentialization and user interface— including the implementation of semantic tagging, L1 and L2 error analysis within domain knowledge and learning path creation based on linguistic domain knowledge.

In conclusion, the evaluation of software in relation to the employment of domain, learner and pedagogical knowledge can lead to better consistency in CALL system creation and evaluation. Furthermore, the implementation of the stated recommendations would further increase user retention of the content in these systems.

Keywords: Duolingo, Rosetta Stone, Memrise, Transparent Language, CALL, software, ITS, semantic tagging, domain knowledge, learner knowledge, pedagogical knowledge.

1 INTRODUCTION

Computer Assisted Language Learning (CALL) is becoming more and more popular mode for education in the classroom. Though this form of pedagogy is no panacea for the complexities that come with a language classroom, the individualized feedback and repetition opportunities that CALL provides for language learners is beneficial in a language classroom [1]. Additionally, CALL provides multiple benefits that a teacher in a large classroom cannot often continually and simultaneously provide for each individual in a language classroom. Some of these benefits include "unlimited stored knowledge, focused and personalized interactions, infinite patience and time, full attention, immediate feedback on each response, student-led pacing, and perfect consistency" [2]. In Intelligent Tutoring Systems (ITS) these benefits can be further expanded. ITSs incorporate expert knowledge (often also referred to as domain knowledge) pedagogical knowledge, and student knowledge [3]. Domain knowledge and pedagogical knowledge go hand in hand — pedagogical knowledge bound to the domain knowledge gives the system more information about what to do in certain situations, especially how to give feedback.
of the student's input and what learning path the student should take. In this aspect pedagogical knowledge also works hand in hand with student knowledge — the system evaluates the user actions in regard to the learning path and adapts the learning path to the user as well as preforms error analysis. In error analysis, the system identifies patterns in the student's errors and provides specialized feedback to the student — as to which part of the student's input is incorrect, what type of error it is, how frequently these errors are made, and what content or background knowledge the user needs to be reminded of to prevent or recast these errors.

The actions of an ITS is similar to the actions a teacher takes in task creation in a language classroom. The teacher's role is first to create comprehensible input where the syntactic complexity of the tasks is just beyond the student's comprehension level [4], and the content in the task is based on real-life language use [5]. Then, the teacher will order tasks and content within tasks based on the ideal learning path. The teacher determines the learning path based on the student knowledge — the students current comprehension level, background knowledge and complexity of the language task. This is often referred to as scaffolding. The feedback the teacher gives in the language classroom is, as in ITSs, based on the teacher’s content knowledge but also varies based on the goal of the task. The teacher's feedback changes based on whether the goal of the task is accuracy in language production or fluency [5][6][7][8][9][10][11][12] or if the goal of the task is accuracy in language form or meaning [13]. Contrastingly, the goal of an ITS system is always accuracy. Another way that teacher actions in task creation mimic that of an ITS is in the creation of pre and post tasks. Pre-tasks are often referred to as "consciousness-raising activities" in which there is a focus on the language form or topic that will be used in the coming task. Post-tasks may help the user switch between a focus on form or meaning and a focus on fluency and language production [5].

2 IDEAL COMPONENTS FOR A LEARNING SYSTEM

Ideally, in CALL software domain knowledge would include the systems awareness of the target language structure, form and use. Pedagogical knowledge would include the complexity/difficulty of a certain task/word and to which other tasks it correlates and error analysis. The student knowledge within CALL systems would not only include awareness of the accuracy of the user input as a whole but classify actions of the user by quality and type. Instead of just showing the results to the student, the system can be aware of in what classes of tasks the user is good or bad at and adapt the learning path or the tasks on that knowledge. Knowing the user's native language and its influence on the user’s acquisition of the target language help to not under- or overestimate the skills of the student. This influence between the learner's native language and target language is referred to as "bidirectional transfer". Though there is deliberation as to the degree of transfer that occurs in adult and child language learners as well as what aspects of language are transferable, many researchers attest to the impact of a learner’s native language on the target language and vice versa [4][14][15]. Subsequently, the administration of feedback within CALL software and structuring of a language plan in the pedagogical knowledge would also be in regard to bidirectional transfer.

Current CALL software analysis based on task creation often involves the investigation of user, teacher, and developer preferences and perceptions of the software [16][17][18]. Evaluation from a user's perception is often related to motivation [19][20][21][22]. From a developer's perspective the evaluation of task construction is often in relation to time and cost of development [16]. In contrast, this paper proposes an alternative evaluation of CALL software based on the prevalence of domain, pedagogical and student knowledge. We seek to answer the following question: Which features of the components do current CALL software realize?

3 EVALUATION OF CURRENT SYSTEMS

There are many options for CALL software available on the market today. Duolingo, Rosetta Stone and Memrise remain at the top of the list for the most popular CALL software to date. Duolingo has an user base which is over 200 million users. Rosetta Stone also has a several million user base [1]. Similarly, Memrise has a user base of 20 million [23]. It is ideal to investigate the prevalence of domain, pedagogical and student knowledge in the most popular titles. The following table outlines the tasks that users complete within the system. What follows is an evaluation of each of these components in these CALL software systems.
3.1 Duolingo

Duolingo is a largely focused on grammar and vocabulary instruction [1] that involves the "manipulation of language rules and forms, designed to increase linguistic knowledge, accuracy, and fluency" [24]. The user performs tasks surrounding translation where the user goes back and forth between the user's native language and target language. The tasks the user completes in the app are either between the native and target language or just within the target language. Some of these tasks include arranging words to form a sentence, select the correct translation for a word or sentence, match a word or phrase to its translation, listen to a recording and record a response or repetition of it, read a sentence and record a response, read a sentence and write a response, and listen to a sentence and write a response or repetition of it, look at a picture and describe it or use it in a sentence. The system also includes a discussion forum, leaderboard, daily and weekly goals, badges, and a shop where users can spend what they earn from completing goals or progressing in the system.

In this system, domain knowledge is clearly recognizable as the lessons themselves, they are static and therefore will not change. You are even obligated to repeat the same lesson multiple times to progress. Even in the feedback it shows is only the domain knowledge — the system translates the sentence or word to the correct form if it is incorrect, highlights the response in green or red and plays alternative sound effects depending on the correctness of the input. Very little pedagogical knowledge is evident in the establishment of only one learning path demonstrated in the user interface (see Fig. 1). There is no information given why an answer is incorrect and not adaptation of learning material. The system has a static order of the vocabulary and grammar lessons.

The system's student knowledge is evident in its tracking of user use in relation to goals, badges, and leaderboards, but the knowledge is rudimentary and is not used by the system itself to adapt to the learning speed or previous knowledge of the user. The user is made aware of this student knowledge when they receive prizes in the form of "Lingots" that they can use in the shop to purchase items, unlock content, or help achieve their daily or weekly goal. The student itself has to evaluate their own statistics, even though he is not able to change the learning speed. The design of the domain knowledge includes, from a linguistic perspective, knowledge of the target and native language of the user but the system fails to integrate student knowledge by highlight patterns in student's errors or give recommendations.
to the user based on the bidirectional transfer between the target and native language. It also fails to integrate pedagogical knowledge by not alternating the learning path to fit the user. The system claims student knowledge in its ability to collect information from its large user base which will in turn "inform student learning in terms of student learning objectives, engagement, and outcomes" [25].

3.2 Rosetta stone

Rosetta Stone, like Duolingo is largely surrounding vocabulary acquisition although instead of relying on translation between the target and native language to deliver the content it is delivered in the target language. Some of the tasks that users complete in the system are read and repeat the word, phrase, sentence or passage, match the picture to what you hear, match the word or phrase to the picture, and listen and repeat what you hear, match the word or phrase to complete the sentence. The user can additionally read stories and passages in the target language and go through flashcards of vocabulary and translations. The system as an app additionally lets the user take photos and translates the word for the object into the target language.

In the system domain knowledge is evident that it shows you vocabulary and tasks that are static. Also here the feedback is a simple correct/incorrect, in that it displays a red "X" or green check mark as well as differing sound effects based on if the user's input is accurate in relation to domain knowledge. The tasks in this system are largely in the target language so there is no reference in the pedagogical or student knowledge to bidirectional transfer. Though the system tracks the frequency of the user's correct input (see Fig. 2), the system does not give feedback to the user as to any patterns in their errors nor does it restructure the learning path based on error repetition. Along these lines pedagogical knowledge is elementary at best. As with Duolingo, the learning path is preset and determined by the system in its construction of lessons and units which are unlocked as the user progresses in the system. Are findings echo that of other researchers who have concluded that "while Rosetta Stone provides interesting displays, its binary feedback, poor recognition abilities, and poorly scaffolded, unexplained features leave much to be desired" [1].

![Figure 2. Rosetta Stone user interface](image)

3.3 Memrise

Similar to both Duolingo and Rosetta Stone, Memrise focuses on vocabulary and grammar acquisition. The tasks that are available in this system are match what you read to what you hear, match what you hear or see in a video to a word or phrase, listen and select the translation, arrange the words to form a sentence, arrange the letters to form a word, read the word or sentence and match it to the translation, and read and select a response (in the form of a chat). Like Rosetta Stone, the Memrise app accesses the phone or tablet camera and allows users to take pictures objects and gives the translation in the target language. Additionally, the system offers daily and weekly goals, words per day to learn goals and leaderboard.

As with the other systems, the prevalence of domain knowledge is easily identifiable by static content, which will be the same on repetition. The student may learn the layout of the question, not the answer
to the underlying question. In the feedback it is also only highlighting the answer in red or green and the altering of sound effects based on the correctness of the input. This chat function is an example of stronger pedagogical knowledge than the previous systems. Within this chat function, the system provides hints in the form of grammatical explanations or short bidirectional explanations when the user’s input is incorrect (see Fig. 3). The prevalence of these hints as well as the algorithms that repeat words or phrases from exercises the user answered incorrectly show the existence of student knowledge within the system. Additionally the system keeps track of the words the user has learned and schedules reminders and activities for the user to practice these words at the time when the memory is most likely to fade [17] as well as tracking how many words the user has learned and how many points they have earned from answering correctly, and how frequently the user has logged into the app. The evidence of student knowledge is in the system tracking the user’s frequency of use, daily words to learn goals, and points earned from answering questions correctly.

Figure 3. Chat window in Memrise

4 HOW TO IMPROVE THE CURRENT SOFTWARE

From the investigation of this software it can be seen that static domain knowledge is easiest to attain in CALL software. It also demonstrates that rudimentary student knowledge is also possible in the form of tracking the frequency of the user’s login or activity in the software and in the tracking of the accuracy of user input. Unfortunately, the student knowledge is rarely used by the system itself to adapt to the user. Furthermore, the user can be made aware of his own knowledge estimated by the system easily in the form of rewards, as in Duolingo, in the form of charts and progress bars, as found in all the systems, or in the unlocking of new content within the domain knowledge, as also seen in all of the systems. From this analysis what appears the most elementary or lacking in these systems is pedagogical knowledge. As there are already student knowledge logs pedagogical knowledge would bring the most benefit for lowest effort and cost for the developer. The use of hints is one of the most straightforward advancements that can be implemented in these systems. In Memrise, the user of bidirectional transfer knowledge between the target language and the native language can be used to expand hints to the vocabulary section. Tasks that are multiple choice and serve to be a repetition of vocabulary words in the target language can be decreased and more tasks can be introduced that allow the users to manipulate the words to form sentences and then hints can be implemented based on the users native and target language. These steps can also be done in Rosetta Stone. In Duolingo these tasks already exist so there remains the implementation of hints when the user’s input is incorrect. To further advance pedagogical knowledge or to make pedagogical knowledge more transparent to the
user, the systems can make the learning path manipulative by and adaptive to users. In all three software systems users are able to skip ahead on some level, but it is restricted based on the user's progress in content. If these restrictions were lifted and the users could advance as far as they wanted if they pass specific tests this would make the learning level more transparent to user. The student knowledge could accompany that with the feature, that knowledge is forgettable, so that untested knowledge should be asked now and then. Some additional improvements that can be made to these systems but are more work-intensive are better user evaluations. An example of this is taking the knowledge of the student, (Duolingo and Rosetta Stone do this by recording how frequently the user's input is correct) and then suggesting advance levels, units or even more complicated tasks to the user when their input is consistently correct for a specified pre-set time. In contrast these systems could implement a difficulty scale where the system orders tasks from easy to difficult and implements more difficult tasks when the user's input is consistently correct. It may be argued that Memrise and Duolingo do this already in the implementation of more complex vocabulary and grammar rules, but the implementation is slow and, to the user, has little correlation to the correctness of their input. Where the semantic tagging of the smallest learning elements is also a possible advancement for these systems. Another complex change that can be made is the application or improvement to the chat function. Since the users are given limited multiple-choice responses in the Chat function of Memrise, the users could be allowed open response where they are able to type responses instead. Furthermore, Duolingo have few tasks that allow for free responses where the user is arranging words or letters to form sentences and words. Expanding these free response activities would also allow for hints. This free form discussion would require significant advancement in domain knowledge of possible responses, or an implementation of machine learning. Finally, Duolingo and Memrise have rudimentary peer -peer interactions — Duolingo through leaderboards and Memrise through affinity spaces. However collaborative peer-peer interaction has been observed to increase motivation and use of computer systems [26].

5 CONCLUSIONS

Using the results of research in the field of ITS to improve already existing industry leading teaching software in the domain of language learning it is possible to reduce the requirements. Instead of developing a new ITS, which may be the goal but too far-fetched for a company specialized on learning software today, we propose a solution that is adequate in size to be implemented on already existing software. The software has to be maintainable with a workforce in the size of the existing one to be a viable product which pays off for a company on the language teaching software market. The current CALL software is focused on gamifying the experience and designing a good-looking user interface, but are neglecting the process of learning itself and keep their software as an implementation of only the behavioristic learning theory.

REFERENCES


