ISCA Archive
http://www.isca-speech.org/archive

Speech and Language Technology
in Education (SLaTE2007)
Farmington, PA, USA
October 1-3, 2007

DeSIGN: An Intelligent Tutor to Teach American Sign Language

Ling Xu, Vinithra Varadharajan*, Joyce Maravich†, Rahul Tongia*, Jack Mostow**

(*) Carnegie Mellon University, (†) Western Pennsylvania School for the Deaf
Pittsburgh, PA, U.S.A.
{lingx, vvaradha, mostow}@cs.cmu.edu, tongia@cmu.edu, jmaravich@wpsd.org

10.21437/SLaTE.2007-10

Abstract

This paper presents the development of DeSIGN, an educational software application for those deaf students who are taught to communicate using American Sign Language (ASL). The software reinforces English vocabulary and ASL signs by providing two essential components of a tutor, lessons and tests. The current version was designed for 5^{th} and 6^{th} graders, whose literacy skills lag by a grade or more on average. In addition, a game that allows the students to be creative has been integrated into the tests. Another feature of DeSIGN is its ability to intelligently adapt its tests to the changing knowledge of the student as determined by a knowledge tracing algorithm. A separate interface for the teacher enables additions and modifications to the content of the tutor and provides progress monitoring. These dynamic aspects help motivate the students to use the software repeatedly. This software prototype aims at a feasible and sustainable approach to increase the participation of deaf people in society. DeSIGN has undergone an iteration of testing and is currently in use at a school for the deaf in Pittsburgh.

1. Introduction

Children build and practice their vocabulary through parental action and interactions with adults and other children in addition to explicit instruction at school. This vocabulary is essential to understand texts that they read, and to participate in verbal communication, which is important for their integration into society. Most deaf children educated in the U.S. are either taught via the American Sign Language (ASL) method, where students learn English for written communication and ASL for verbal communication or via the auditory/oral method where they learn speech and lipreading. This work supports the ASL method for effective communication.

The average deaf student graduates from American high schools with a 4^{th} grade reading level [1]. One reason may stem from the fact that 90% of deaf children are born to hearing parents [2]. Hence, many deaf children, whose parents are not familiar with or fluent in ASL, do not have the opportunity to practice ASL at home. A study conducted by Davey, LaSasso, and Macready states that there is a strong correlation between vocabulary knowledge and reading comprehension. This research indicates that a lack of vocabulary knowledge hampers reading skills significantly [3]. Our work endeavors to address this learning gap by reinforcing the meaning and use of vocabulary, and its corresponding ASL signs in a fun and interactive way. In this process, we also illustrate the application of computing technology to the educational needs of the deaf community. The design of the solution went beyond a generic approach to technology, instead observing students in class and learning about their preferences, other software they use, and the general computing environment available to students. This paper presents a 9-month student project conducted by the first two authors. This work was done in collaboration with the Western Pennsylvania School for the Deaf (WPSD) in Pittsburgh and with assistance from Ms. Joyce Maravich, the reading and computer science teacher for 4^{th} to 6^{th} grades.

1.1. Relation to Prior Work

Several projects at various universities currently focus on technology applications for the educational and/or deaf domains. The ASL project at DePaul University focuses on combining technology and linguistics to build an English-to-ASL translator [4]. In contrast to our work, the DePaul project assumes the user is well versed in English and ASL, and does not cater to primary education. Another group at Georgia Institute of Technology addresses the literacy of deaf students through CopyCat, a gesture-based sign language game that uses wrist-mounted sensors, computer vision and machine learning to test students' ability to comprehend and sign sentences [5], [6]. While the primary goal of CopyCat is active learning, techniques used to explicitly train students to sign, the project's user testing and research results helped define our approach of using ASL videos and allowing dynamic content. Finally, Project LISTEN at Carnegie Mellon University is an ongoing effort that aims to improve the literacy level of primary school aged hearing children [7]. Although this project utilizes speech recognition, we incorporated its student modeling techniques to measure students' reading ability in our work.

2. Tutor Components

The software consists of two parts. The first part presents vocabulary words and ASL signs (lessons). The second part evaluates the student's knowledge (tests). An interactive game integrated into the testing component motivates the students to use the tutor repeatedly, an aspect necessary for effective reinforcement. Additionally, a machine learning algorithm lets DeSIGN intelligently adapt its questioning to the student's performance. The software was developed in Java and the content is stored in a MySQL database.

An important goal for the software was to allow easy content updates by the teacher since a pre-selected dictionary of words may not coincide with the words taught in class, and may not match the region-specific ASL signs. This required a feasible and sustainable software solution that would include a simple way to add new signs that would be realistic and engaging for the students. After considering ASL sign generation using animation and motion capture techniques, we decided to use videos of an ASL

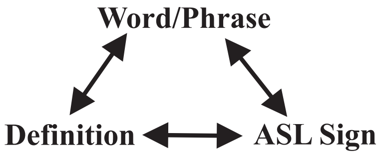

Figure 1: Bi-directional associations between skill components

signer. The process of recording an ASL interpreter is familiar, inexpensive, and relatively easy. Since the videos include either the teacher or an ASL interpreter, they are much more realistic than available motion capture or animation techniques. Videos can also incorporate additional elements, such as the signer's facial expression, that would be difficult to acquire otherwise.

2.1. Lessons

Since we wanted DeSIGN to use established teaching techniques for the deaf, we observed Ms. Maravich's teaching methods and incorporated them into the lessons. Also, we discussed with researchers in human-computer interaction, language technologies, and game design, the different methods of communicating with children, design of graphical user interfaces for children, and the particular needs of deaf children. Based on this background research, the lessons portion of the tutor was designed to help build the vocabulary of the students by reinforcing words taught in class as well as by introducing new words. DeSIGN's lessons focus on the ASL sign, the written word, and the definition in terms of the associations between the three elements as shown in Fig. 1.

Each lesson consists of several words or phrases along with its vocabulary components: definition, ASL video, and optional elements such as synonyms, antonyms, pictures, and uses of the word in example sentences. Each video shows an ASL sign and the finger-spelling of the word or phrase. Initial videos of Ms. Maravich were taken for the tutor prototype. DeSIGN allows the students to click individual buttons to view each component and read through each piece of information at their own speed. For example, the student can choose to read the definition, then look at the video, and finally read through the sentences. This particular layout actively engages the student in the lesson and encourages them to read through all the components of the lesson.

The teacher's interface has functionalities that give the teacher control over the lesson content. Using this interface, the teacher has the ability to add new lessons and words, modify and delete existing words, and also add and modify student information such as the current lesson. Once the lesson content is inserted via the teacher interface, the content is automatically integrated into the student interface.

2.2. Tests

We attended some of Ms. Maravich's classes to observe her interactions with her students, the methods she used to test them, and her feedback to their responses. For example, one mode of testing was to spell a word and ask the student for the sign, while another mode of testing was to ask the student to recognize a word that was signed. DeSIGN tests receptive skills rather than generation, and recognition rather than recall.

The three bi-directional associations shown in Fig. 1 form the basis of the tests. These associations are tested using multiple-choice questions. For example, a question that tests the meaning of a sign displays a video of the sign (without finger-spelling) and four choices of word definitions. The inverse association is tested by displaying the definition of a word and asking the student to choose one video with the same meaning from a choice of four videos. The questions that test the other four vocabulary associations are similar in design. A score keeping track of the student's performance is displayed. A student gets two attempts at a question. A correct first answer is worth two points. If the first attempt is incorrect, hints are displayed and the question becomes worth one point. These hints consist of the other vocabulary components such as examples sentences and synonyms. If the second attempt at answering the question is incorrect as well, the correct answer is highlighted and the student's score does not increase. Messages are displayed that indicate whether the student has answered correctly or not, drawing randomly from a list of messages compiled from our observations of Ms. Maravich.

2.3. Game

A game was included to motivate repeated use of the tutor. Our first step in designing the game was to understand what our end-users found interesting and entertaining. To this end, we discussed different topics such as favorite games, themes, and characters with two 5^{th} grade classes at WPSD. We also found out that the average amount of time spent on a single sitting of a computer-based game is 30 - 50 minutes. Next, we observed students using typing tutor software and game software called *Hollywood*, and perused some of their class work, one of which was the organization of a virtual birthday party. From these observations and Ms. Maravich's comments, we found that both boys and girls liked being creative by decorating virtual scenes, coming up with storylines and character personalities, and sharing their final results with their friends.

This gave us the idea for our game: each student selects an image of a theme or a background scene from a list and designs the scene with items categorized according to the theme. However, in order to select an item, the student must first answer a question correctly. These questions are those described in the previous subsection. The game is expandable through the teacher's interface, which provides functionalities to include new theme and item images. This game also allows practice of topics from other subjects such as science or geography by the inclusion of appropriate themes and items. For example, two of the themes that we provided were 'Solar System' and 'Jungle', with items such as the planets in the solar system and different animals, birds, and insects, respectively. The vocabulary words are not necessarily related to these items, thereby allowing reusability of the game components.

There are three levels within the game. The first level tests the association of word/phrase $\leftrightarrow$ ASL sign, the second tests the association of ASL sign $\leftrightarrow$ definition, and the third tests the association of definition $\leftrightarrow$ word/phrase. The teacher determines the level at which each student starts and the difficulty increases with each level.

2.4. Tutor Intelligence

The next aspect of the tutor that we focused on was to make the tutor adapt its test questions to the students' skills. This required evaluation of the students' knowledge. However, knowledge cannot be directly measured and needs to be inferred from observed performance. An established method to estimate student knowledge is a technique known as knowledge tracing [8]. The knowledge tracing algorithm models skill acquisition by utilizing the relationship between knowledge and performance over time. There are two types of parameters used in knowledge tracing. The learning parameters are prior knowledge, the probability the student knows the skill before using the tutor, and learning rate, the probability that the student learns the skill given a practice opportunity. The performance parameters are 'Guess', the probability the student answers a question correctly without knowing the skill, and 'Slip', the probability the student answers a question incorrectly despite knowing the skill. This tracing technique can be modeled as a dynamic Bayesian network where the learning and performance parameters do not change over time [9]. For our purposes, the skill is the word knowledge represented by all six associations. The Bayes net parameters can be estimated from performance data for a sample of students. The observable performance is whether the student answers a question correctly in the first try.

The data required to estimate the knowledge tracing parameters was collected during the first iteration of user testing. During this iteration thirteen students used DeSIGN and generated 288 data points. Due to the small size of the dataset, the data points generated from different words were treated as if from a single word. Hence, the tutor learned one set of parameters for all the words on which each student was tested. We used the Bayes Net Toolkit for Student Modeling (BNT-SM) to estimate these knowledge tracing parameters [10]. Prior to training, the knowledge tracing algorithm initially assumed zero prior knowledge, a guess probability of 0.25 since there are four answer choices, and zero for the slip and learn probabilities. After training, knowledge tracing estimated a prior knowledge probability of 0.37, a guess probability of 0.84 due to partial knowledge helping to eliminate wrong answers, a slip probability of 0.03, and a learning probability of 0.18. The final student knowledge estimated from the training data was 0.60 meaning that the students' estimated knowledge of the target vocabulary words increased by 0.23 while using DeSIGN. We recognize that this method of student modeling is prone to the identifiability problem, where different sets of model parameter values predict the same performance, and a more accurate estimate of student knowledge would require more data points [11].

We then used the learning and performance parameters determined by the algorithm for online knowledge tracing. As the student answers the test questions, the knowledge tracing estimates the current state of word acquisition by the student using equation 1. K_n and C_n represent the knowledge and performance, respectively, at time n. Equation 1 shows that the knowledge at time n is the sum of the posterior probability that the skill was already learned and the probability that the skill will be learned if it is not already learned. Equation 2 gives an expression for the posterior probability. Both these equations utilize the knowledge tracing parameters estimated using the BNT-SM toolkit.

$$p(K_n|C_n) = p(K_{n-1}|C_n) + (1 - p(K_{n-1}|C_n))p(learn) \quad (1)$$

$$p(K_{n-1}|C_n) = \frac{p(C_n|K_{n-1})p(K_{n-1})}{p(C_n)} \quad (2)$$

The learning parameter computed after each question is then compared to a threshold value. If the learning parameter is less than the threshold, the skill is tested again later in the testing session. The threshold value used in the second iteration of user testing was preset to 0.5 to ensure the student receives at least one more practice attempt if he or she answers questions incorrectly.

3. User Testing and Feedback

User testing was conducted at WPSD with 5^{th} and 6^{th} graders, ages roughly 10 to 11, with a 2^{nd} to 3^{rd} grade reading level. Most of the students possessed a better signing vocabulary than written vocabulary, just as hearing children's oral vocabulary exceeds their reading vocabulary. Additionally, we encouraged and were receptive to feedback from the students and the teacher throughout the project. For example, after we had designed the game, we used a prototype to test the concept behind the game and to gauge the student's interest in the idea. We found that the teachers and the majority of the students were enthusiastic about it and it seemed like the game would be well received. Hence, we decided to proceed with the implementation of the game. After completion of the game, three fellow graduate students tested the software extensively. This round of testing, though not representative of the end-users, was very useful in finding small glitches and software bugs.

Next, we conducted two iterations of user testing at WPSD and made improvements to enhance the usability and effectiveness of the tutor between the two iterations. The first iteration lasted two weeks and tested a version of DeSIGN with three of the four project goals implemented: lessons, tests, and the game. The data gathered from the first iteration was used for learning the conditional probability tables for the knowledge tracing algorithm. The tutor with knowledge tracing was tested in the second iteration which lasted three days. Thirteen students (10 girls, 3 boys) participated in the first iteration of user testing. The teacher first explained how to use the software and was available to answer any questions that the students had while using the software. During the first iteration, Ms. Maravich conducted paper based pre and post tests for 8 of the 13 students due to time constraints. These tests assessed the students' knowledge of the seven ASL signs that DeSIGN taught. The tests were rigorous in terms of correctness of tense and preciseness of usage. However, the students were assured that the tests did not count for a grade. The students used the software for 20-25 minutes between the pre and post tests.

4. Results

The results from the pre and post tests conducted during the first user testing iteration showed an improvement of 17.9% from 53.5% to 71.4% in student performance in the target vocabulary words after using DeSIGN once. This improvement is consistent with the increase in student knowledge estimated by knowledge tracing. The average time spent on each test question was $20.03s$. Students who performed well on the pre-test did not improve greatly on the post-test after using DeSIGN, but students who performed less well on the pre-test seemed to improve significantly on the post-test. It is important to note that these results are not statistically significant due to the small dataset (roughly 300 data points that were non-independent due to within-student correlation). However, these preliminary results indicate that DeSIGN has the potential to help deaf students learn and retain vocabulary.

Beyond direct improvement in vocabulary, DeSIGN successfully captured the students' attention and they did not seem distracted while using the software. Also, the tutor piqued the curiosity of some students and they were eager to find out how the software was created. An interesting finding was that the students initially showed a lot of interest in playing the game and populating the scene. However, after some time they no longer played the game, and instead focused and spent more time on answering the questions. One way of interpreting this behavior is that the game provided initial motivation, but answering the questions correctly was challenging enough later. Some other possibilities are that completely filling their scene with items or having used all the items for that theme at least once caused students to lose interest in the game.

Overall, we received positive feedback from the students and teacher regarding DeSIGN. Also, the teacher noticed that the game helped in other aspects besides vocabulary such as learning to finger-spell the names of the different items placed in a scene by asking the teacher. Finally, our discussions with the school indicate that they would like their other teachers to use DeSIGN in their classes.

5. Conclusion

We have developed a prototype of an intelligent tutor that has the potential to accelerate the learning process of deaf students. To this end, the tutor has the basic and essential components of lessons and tests that provide opportunities for effective reinforcement of English vocabulary, ASL signs, and definitions. Additionally, we have included a game to motivate the students to use the software repeatedly by tapping into their creativity. Finally, a knowledge tracing algorithm has been included to make the tutor intelligently adapt to the students' knowledge.

This project devoted significant time to understanding the needs of the deaf community, compiling specifications of users' requirements, gathering and incorporating feedback, and conducting background research and investigation. This information, in combination with a rapid implementation phase, enabled the development of a usable and promising piece of software. We believe that projects geared towards addressing the needs of an underserved community need to be sensitive to the culture of the community. To this end, committed partners within the community, such as WPSD, are invaluable. They provide a wealth of information and intuition regarding effectiveness, feasibility, and sustainability of proposed solutions.

In conclusion, our work is an example of the development of intelligent educational software for a differently-abled community, and is an application of computing technology to the education of under-served communities.

6. Future Work

In order to thoroughly assess the effects of the software application, DeSIGN should be used for longer periods of time, after which further testing can be performed and the effectiveness of DeSIGN can be evaluated. An impartial assessment of the game requires user testing with other students and teachers who have not participated in the project. The input of multiple teachers would help refine the features of the tutor. Larger sets of data and tests will allow for statistically significant results. The current implementation of DeSIGN uses the knowledge tracing algorithm to repeat questions on which the student requires more practice. However, the algorithm is not used to eliminate words the student already knows. This results in a slight loss of interest due to repeated testing of familiar vocabulary. Aspects with respect to deaf education that require specific attention include chunking and spelling. In ASL, a group of words "chunked" together can have a different meaning or a different sign from the individual words in the group. Feedback provided by the teachers shows that students have difficulty grasping the nuances of these word chunks.

In addition to these short-term improvements, there are a few long-term directions that DeSIGN can now take. We envision that these steps can be made in the form of other student projects and assignments at Carnegie Mellon as part of the TechBridgeWorld program. For example, the inclusion of realistic and accurate animations might greatly increase the appeal of the software. While we have worked on recognition of ASL signs, it would be useful to incorporate sign recall by using computer vision techniques to recognize the gestures of the students captured via a camera. Additional games or a multi-user mode allowing collaboration or competition, perhaps even over the Internet, might make the software more motivating.

7. Acknowledgements

This project was funded in part by Carnegie Mellon University's TechBridgeWorld under the V-unit program, and in part by the National Science Foundation under ITR/IERI Grant No. REC-0326153. The authors would like to thank the faculty and students at WPSD for collaborating with us on this project. Thanks to the V-unit coordinators, Dr. M. Bernardine Dias and Prof. Manuela Veloso, for helping us refine and extend our goals. Also, the authors thank the various faculty, students, and staff at Carnegie Mellon University who helped with this project. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation or the official policies, either expressed or implied, of the sponsors or of the United States Government.

8. References

[1] Bowe, F. (1991). Approaching equality: Education of the deaf. Silver Spring, MD: T.J. Publishers.

[2] Gallaudet University. Regional and National Summary Report of Data from the 1999-2000 Annual Survey of Deaf and Hard of Hearing Children and Youth. Washington, D. C., 2001.

[3] Davey B, LaSasso C, Macready G. Comparison of reading comprehension task performance for deaf and hearing readers. Journal of Speech and Hearing Research Vol.26 622-628 December 1983.

[4] American Sign Language Project, DePaul University, http://asl.cs.depaul.edu/.

[5] Valerie Henderson, Seungyon Lee, Harley Hamilton, Thad Starner, Helene Brashear, Steven Hamilton. Development of American Sign Language (ASL) Game for Deaf Children, Proceedings of IDC(Interaction Design & Children). Boulder, CO. June 2005.

[6] Seungyon Lee, Valerie Henderson, Harley Hamilton, Thad Starner, Helene Brashear, Steven Hamilton. A Gesture-based American Sign Language (ASL) Tutor for Deaf Children, Proceedings of CHI(Computer-Human Interaction). Portland, OR. April 2005.

[7] A. G. Hauptmann, J. Mostow, S. F. Roth, M. Kane, and A. Swift, "A Prototype Reading Coach that Listens: Summary of Project LISTEN." In C. Weinstein (ed.), Proceedings ARPA Workshop on Human Language Technology,March 1994, Plainsboro, NJ, page 237. Morgan Kaufmann Publishers, Inc.

[8] Corbett, A. and J. Anderson, Knowledge tracing: Modeling the acquisition of procedural knowledge. User modeling and user-adapted interaction, 1995. 4: p. 253-278.

[9] Reye, J.: Student modeling based on belief networks. International Journal of Artificial Intelligence in Education 14 (2004) 1-33.

[10] Chang, K., Beck, J., Mostow, J., & Corbett, A. (2006, June 26-30). A Bayes Net Toolkit for Student Modeling in Intelligent Tutoring Systems. Proceedings of the 8th International Conference on Intelligent Tutoring Systems, Jhongli, Taiwan. 104-113.

[11] Beck, J. E., and Chang, K.-m. (2007, to appear, June 25-29). Identifiability: A Fundamental Problem of Student Modeling. Proceedings of the 11th International Conference on User Modeling (UM 2007), Corfu, Greece.