Automatic Generation of Technical-Style Notes from Live Lecture

Adrian Rusu, Gary Dainton, Kevin Dahm, and James Metting
rusu@rowan.edu, dainton@LCRinfo.com, dahm@rowan.edu, metting61@students.rowan.edu

Abstract - With student persistence and retention being a cascading issue within higher education (particularly in Computer Science and Engineering), developing processes and tools to help students overcome the challenges of systemized learning becomes an academic necessity. The authors contend that if an information translation apparatus was in existence that could generate individual learning data formats, students will have another tool to understand and apply information that was delivered in a fashion un-natural to their personal learning processes. Survey data has revealed that in many situations students spend mental effort trying to record, via note-taking, exactly the words being used by the instructor instead of the synergy and application of the content. Voice recognition software (voice-to-text) is the vehicle by which data is digitally recorded for format manipulation between a marriage of human learning and technology. The research-tested Interactive Learning Model© is the foundation of the learning systems component of this project and an innovative algorithmic development was utilized to manifest specific data formats consistent with technical learning processes.

Index Terms - accessible multimedia, active learning, synchronized speech and text, visual summaries

INTRODUCTION

Voice recognition software (VRS) has the potential for many educational applications beyond the prevalent function, to record lectures/discussions for the hearing impaired. Using VRS by mainstream students can appear redundant and inconsequential when thought of as an active learning tool. After all, typical output of VRS interface is a simple linear written recording of each spoken word. The output recording is a great tool for archiving lectures, meetings, discussions and to provide data for the hearing impaired, but not typically considered a valuable learning tool for mainstream students. This project was designed with the intention of researching non-traditional applications of VRS as an active learning tool in and out of the lecture hall.

Across the curriculum, institutions of higher education promote and design experiences that require deductive/inductive reasoning, enhance problem solving skills, and enable self discovery. However, single-loop learning continues to be the norm on most campuses, particularly in general education courses. Single-loop learning is characterized by information dissemination, time, then information collection. In this case assessments are not based on reasoning, problem solving, and self discovery but are replaced with note-taking skills and memory. Internal conflict can arise within students as some feel the need to scribe accurate and organized notes at the exact time they are expected to connect theories and concepts. If single-loop learning (data regurgitation) continues as a norm in higher education it is self evident as to which task students will focus their mental force [1].

Double-loop learning is the concept that most resembles the goals of higher education. Double-loop learning is characterized by an opportunity for students to actively engage information, apply the information, and reflect upon the outcomes of that intentional engagement and application of said information [2]. A central component of double-loop learning, active learning, continues as a trend in higher education in that it reinforces theories and works to connect concepts through classroom exercises [3].

When considering VRS as a tool to enhance double-loop learning, we must address the following ubiquitous question: How does the output format of the software affect student learning? The authors contend that VRS output format is central when considering it as a tool to enhance learning experiences. Not only is formation of data important, but VRS has the potential for placing student focus on the global aspect of information by ameliorating the internal drive for proper note-taking as text records of lectures can be provided simultaneously or post-lecture in a format selected by the student [4].

The remainder of this paper is organized as follows. In section 2 we discuss learning processes that form the basis of our research. In section 3 we reveal benefits and limitations of translation software (VRS). Section 4 contains details and outcomes of the research in this paper. Finally, we conclude and list future directions in section 5.

LEARNING PATTERNS

Learning is the manifestation of several components of the individual physiology and human essence. Data collected via the five senses (sight, sound, taste, touch, and smell) is electrochemically processed, sorted, and stored in the brain. In any given moment, those five senses are providing the brain with millions of individual bits of information that can
create both conscious and unconscious reactions. The key to conscious learning is memory. Memory is a physical occurrence that is affected by environmental and physiological factors such as stress and functioning chemical receptors. In simple terms we can consider the brain as the hard-drive of individual human existence [1].

Along with the five ever-present physical senses, multidisciplinary research suggests that individuals possess a sixth sense. If we define "sense" as a data collections source that affects the function of the brain then our sixth sense would be the mind's operation. Just as the five senses provide data that alter the synaptic responses within varying sections of the interactive brain, the mind (thoughts, feelings, action) creates synaptic responses throughout the brain creating a constant bio-chemical loop or interface between the brain and mind [5]. As the brain is the hard-drive of our existence, it is our mind that is the software that decodes the stored bits of data into symbolic representations we can manipulate to communicate, problem solve, and survive in general. In many experiences and situations inside and outside of the classroom, the student's internal mind-brain connection (software package) is not naturally compatible with the instructor's mind-brain connection, thus creating a potential data interface error.

The Interactive Learning Model© (ILM) is the theoretical base of the integrated learning processes [6] for this initiative. The ILM purports that during the brain-mind interface data is processed using different degrees of four interactive patterns of operation and learning. These four observable patterns manifest themselves not only in our cognition (thinking), but our affectation (feelings) and conation (actions) as well. In order to accurately measure the four learning patterns the Learning Connections Inventory was developed [7]. The Learning Connections Inventory (LCI) is a 28-item Likert-style instrument that contains three interactive short answer questions used to validate responses. The results of the LCI yield ranges into which a person "avoids", "use as needed", or "use first" all four of the interactive learning patterns.

For example, the scale score for Sequence somewhere at a Use First level indicates the following:
- They want clear directions.
- They need step-by-step directions.
- They want time to do their work neatly.
- They like to do their work from beginning to end.
- They want to know if they are meeting the instructors or their teammates' expectations.

The scale score for Sequence somewhere at an Avoid level indicates the following:
- They tend not to read directions.
- They don't plan or live by a schedule.
- They rarely double-check their work.
- They find following directions confusing - and maybe even frustrating.

The scale score for Precise at a Use First level indicates the following:
- They want complete and thorough explanations.
- They ask a lot of questions.
- They like to answer questions.
- They need to be accurate and correct.
- They like test results.
- They seek written documentation.

The scale score for Precise somewhere at an Avoid level indicates the following:
- They rarely read for pleasure.
- They don't attend to details; details are bothersome and boring.
- They find memorizing tedious and a waste of time.
- Much of the wordy conversation going on around them simply sounds like, "blah, blah, blah".

The scale score for Technical at a Use First level indicates the following:
- They don't like to write things down.
- They need to see the purpose of what they are doing.
- They like to work by themselves.
- They like to figure how things work.
- They don't like to use a lot of words.
- They look for relevance and practicality.

The scale score for Technical at the Avoid level indicates the following:
- They don't get involved with taking things apart to understand how they work.
- They work with others to do building and repair work.
- They don't venture into the tool aisle.
- They problem solve with others not alone.

The scale score for Confluence at a Use First level indicates the following:
- They don't like doing the same thing over and over.
- They see situations very differently than others do.
- They like to do things their own way.
- They don't like following the rules.
- They enjoy taking risks.

The scale score for Confluence at the Avoid level indicates the following:
- They think taking risks is foolish and wasteful.
- They would rather not make mistakes than having to learn from their mistakes.
- They are more careful and cautious in how they go about making life decisions.
- They seek conventional approaches.

When learning patterns fall into the As Needed range then they use these patterns when they need to. They just don't feel a great urgency to use them, especially if they fall into the mid-range. These patterns tend to lay dormant until
they need to be tapped into depending on the task. Use as needed patterns don't drive learning like "Use First" and "Avoid" patterns.

Research conducted 2003-2006 at Rowan University, a four-year University, revealed that the freshman and sophomore classes of engineering students (N=304) began their learning at the "Use-First" level in the Sequential learning pattern and Technical Reasoning learning pattern and used the remaining two learning patterns (Precision and Confluence) at the "As Needed" level as determined by the mean and median scores of the LCI. Similar correlation research revealed that the faculty at the same university (N=53) used their Precise and Confluent learning patterns at the "Use First" level while Sequence and Technical Reasoning were used at the "As Needed" level as determined by mean and median [5].

An analysis of the instructor versus student learning patterns suggests a potential disconnect within the sequence and precision learning patterns. The students are seeking more organization, modeling, and expectations than the instructors naturally offer while the instructors may be seeking more detail, information, and use of language than the students naturally offer. This can cause a potential disconnect between student and instructor unless both parties can articulate their differences and develop strategies to negotiate, connect, and meet the needs as both instructor and learner.

Multiple-option VRS output format can be a valuable tool for students to translate speech into text formats that more naturally fit personal learning processes.

**TRANSLATION SOFTWARE**

During the 1950s, research on machine language-to-language translation took form in the sense of literal translation, more commonly known as word-for-word translations, without the use of any linguistic rules. Since then, the accuracy of language-to-language translation has significantly improved, allowing for a plethora of software applications, including all types of automatic translation software: text-to-text [8], [9], speech-to-speech [10], text-to-speech [11], and speech-to-text [12], [4]. In addition to language-to-language translation, text-to-speech and speech-to-text software have been developed for same language translation. Limitations for such software packages have included accurate use of punctuation and difficulty in comprehending a stream by the end user [4].

Research and development of speech-to-text software is a simplistic dynamic. Yet, the evolution of software packages remains a complicated issue. The confounding variable in all speech-to-text software remains the human personage. Populist trends in speech recognition involve the handicapped and learning disabled but there are no restrictions on the benefits of such software when considering heuristics in human capacity and learning.

**Session S3B**

**SPEECH-TO-TEXT-TO-TECHNICAL SOLUTION**

The speech-to-text-to-summary scenario follows the following major steps:

- Learning Connections Inventory is administered to faculty and students once to identify their personal learning patterns. In this case the LCI is administered to all incoming freshman during orientation and databases are maintained. Faculty are invited to complete the LCI during several on-campus professional development seminars.
- The instructor develops a personalized voice profile by teaching speech recognition software to understand her speech.
- During lectures the instructor uses a wireless microphone connected to a computer system running speech recognition software.
- The speech recognition software receives a digitized transmission of the spoken lecture and converts it to electronic text.
- The students receive personalized lecture summaries based on their individual learning patterns.

In this paper we have targeted processes that accommodate those who possess the Technical learning pattern at the "Use First" level. The authors contend that the technical learners are those who, to a greater degree, are seeking to apply information in other formats beyond linear spoken or written words. The Technical learning pattern is that part of us that problem-solves and seeks to understand how things practically operate and function in the physical world.

Our software takes as input the text produced by speech-to-text off-the-shelf software (for example IBM ViaVoice [12]) and models the output in the form of a hierarchy.

In the current implementation, our focus was on the visual component rather than on making the system instructor-independent. The instructor has full control over what the final lecture notes will contain, through the use of twelve keywords. The selected keywords are simple, can be easily spoken while giving a lecture, and do not appear often as part of a lecture:

- **Start**: begin the lecture.
- **Stop**: end the lecture.
- **New Topic**: begin a new topic.
- **New Subtopic**: begin a new topic nested under a topic.
- **New Microtopic**: begin a new topic nested under a subtopic.
- **More Detail**: elaborate more on a topic.
- **Pause**: temporarily suspend the lecture.
- **Resume**: continue with the lecture.
- **Important**: select the most important concepts from the main lecture.
- **Back To**: start the process of rewriting a previous topic.
- **Write**: overwrite an existing topic with a new one.
- **Append**: add more information to an existing topic.
The topics are presented in the order spoken, unless the user uses the "Back To" command to alter existing topics. The lecturer is allowed to use up to three levels to distinguish the layout of the lecture. The names of these levels are topic, subtopic, and microtopic. A sample input file for a lecture on the role of information hiding in software engineering design is presented in Figure 1. Its corresponding output, which accommodates sequential and technical learners, created by our software is presented in Figure 2. The user could further explore the information by clicking on any of the nodes in the hierarchy. For example, if the user needs to obtain more information on topic "Algorithms", the user clicks on the node labeled Algorithms and a Google "I'm feeling lucky" search is performed, which brings the Wikipedia web page on Algorithms (see Figure 3).

We have implemented our sequence-to-technical software in C++, and the current implementation consists of about 1500 lines of code.

To accommodate technical learners' attraction to problem solving and information formats other than words, we used two other separate concepts: Foundation and Puzzle. We selected the Foundation format because it allows the technical learner to build or construct the information into a representation that demonstrates how concepts are weighted and rely upon other concepts to operate in combination. We selected the Puzzle format because it allows the technical learner to problem-solve and gives the perception of connecting the concepts to be more of a game than traditional memorization of words. The puzzle requires the user to not only recognize the terms of concepts but also how shapes can demonstrate the interconnectedness of those concepts. In both cases, Foundation and Puzzle provide learners formats for recognizing concepts and the relationship of concepts other than through traditional linear word systems.

In the Foundation display, the learner is presented with the information such that the topics are the base information hierarchy, followed by subtopics, which are placed on top of the topics, and microtopics, which are placed on top of subtopics. Topics, subtopics, and microtopics have assigned distinguished backgrounds (see Figure 4 for the Foundation output corresponding to the input presented in Figure 1).
FIGURE 4

Foundation output tailored to technical learners.

We used the web development technique Ajax (Asynchronous JavaScript and XML) to implement the Foundation display in order for the users to be able to interact with the notes online in real time.

In the Puzzle display, technical learners are presented with topics, subtopics, and microtopics spread over the screen in the form of puzzle pieces (see Figure 5 for the initial puzzle setup). In the current version, the topics, subtopics, and microtopics are distinguished by the size of the text inside of the puzzle pieces. They can also be distinguished by the size of the puzzle pieces assigned to them.

The puzzle pieces are designed in such a way that corresponding concepts can "snap" together (See Figure 6 for an intermediate Puzzle step). In order to allow the technical learner to rationale about the information, puzzle pieces do not have predetermined shapes; rather their shapes are randomly generated. We use the following technique to generate puzzle pieces: all puzzle pieces start by having assigned rectangular shapes, and then various-sized shapes are "carved" (or appended) from each initial pieces to randomly generate the final pieces. For related concepts A and B, the same size shape that is carved from piece A is appended to piece B, so the two related concepts can be connected by the learner. In the example in Figure 5, the puzzle piece assigned to concept "Algorithms" has appended a rectangular shape and has carved a triangular shape, which are appended and carved from puzzle pieces assigned to concepts "Data representation" and "Input and output formats", respectively, so that the three puzzle pieces could be connected together (see Figure 6 for the new puzzle piece which is formed by connecting the puzzle pieces "Data representations" and "Algorithms").

The learner can rotate puzzle pieces by mouse clicking and can use a simple drag and drop technique to move a topic, subtopic, or microtopic at a time and connect it with their corresponding concept. If the topic, subtopic, or microtopic is connected with the correct concept, the puzzle pieces remain connected, thus indicating that they are indeed related concepts. Otherwise, the topic, subtopic, or microtopic retreats to its initial location, thus indicating that it has been attempted to be connected with an unrelated concept. See Figure 6 for an example of an intermediate puzzle step and Figure 7 for the finalized puzzle corresponding to the initial setup presented in Figure 5. Notice that the puzzle piece corresponding to concept "Input and output formats" is rotated counterclockwise before it is connected to concept "Algorithms". Similarly, the puzzle piece corresponding to concept "Module interface insensitive to change" is rotated clockwise before it is connected to concept "Anticipate change identify items likely to change".

In the example in Figures 5, 6, and 7, we appended/carved circular, rectangular, and triangular shapes to the initial rectangular puzzle pieces, but other shapes can be appended/carved as well, thus allowing the Puzzle display to function even for highly complex lecture notes.

FIGURE 5

Initial Puzzle display: topics, subtopics, and microtopics are assigned puzzle pieces spread over the screen.

In the example in Figures 5, 6, and 7, we appended/carved circular, rectangular, and triangular shapes to the initial rectangular puzzle pieces, but other shapes can be appended/carved as well, thus allowing the Puzzle display to function even for highly complex lecture notes.
Intermediate Puzzle step: related microtopics labeled "Data Representations" and "Algorithms" have been connected by the learner.

We used the graphics engine JOGL (OpenGL for Java Bindings) to implement the Puzzle display, because of its capability of generating non-trivial graphical displays.

CONCLUSION AND FUTURE DEVELOPMENTS

Automatic notes can be a tool to bridge instructor teaching processes with student learning processes for enhanced double-loop learning and academic success. Traditional lectures can be transformed into active learning classrooms by reducing student self-imposed pressure to accurately scribe and organize spoken words toward elasticity to simultaneously connect core concepts. Our speech-to-text-to-summary initiative also provides assistance in distance learning, as it complements the existing techniques and technology [13] by providing automated summaries available online.

The initial focus of our initiative was to investigate discreet displays for sequential [14] and technical learning patterns and to ensure compliance with potential trademark and copyright infringement of web data. The reason we do not perform "I'm feeling lucky" searches for images on the web is that it would violate such copyright rights.

As a first next step, we plan to integrate all our implementations into a single software package.

Next, we intend on evaluating our displays in a classroom environment and then refine them to be in accordance with students’ learning patterns.

We also plan to investigate display formats that are characteristic to other learning patterns and to work on finding methods to ease our current reliance on keywords and instructor, by developing techniques to automatically parse the lecture material and "learn" relationships among important concepts.

ACKNOWLEDGMENT

We thank students Eric Vermullen and Giovanni DeCristofaro for their contribution to the implementation.

REFERENCES


AUTHOR INFORMATION

Adrian Rusu, Department of Computer Science, Rowan University, rusu@rowan.edu

Gary Dainton, Learning Connections Resources LLC, Dainton@LCRinfo.com

Kevin Dahm, Department of Civil and Environmental Engineering, Rowan University, dahm@rowan.edu

James Metting, Department of Computer Science, Rowan University, mettin61@students.rowan.edu