Automatic Generation of Sequence-Style Notes from Live Lecture

Adrian Rusu
Department of Computer Science
Rowan University
Glassboro, NJ 08028, USA
rusu@rowan.edu

Gary Dainton
Learning Connections Resources, LLC
Turnersville, NJ 08012, USA
Dainton@LCRinfo.com

***Abstract*— Speech recognition software (speech-to-text) accuracy is improving at a fast pace, thus making possible a diverse range of new applications. The Liberated Learning Initiative has demonstrated that automatic speech recognition software provides the potential of making instruction more applicable to students, particularly for deaf students who can follow the lecture without the distraction of an interpreter. In this initiative, speech recognition software is used as a tool to help students with varying learning processes.**

The research-tested Interactive Learning Model asserts that individuals process information using a combination of four learning patterns. Ongoing research has shown that the majority of computer science and engineering students use Sequence as a primary learning pattern. In this paper we present a novel speech-to-text-to-sequence method for supporting learners who seek Sequential instruction. Our method encourages active learning as students can focus on the content and context of the instruction with the understanding that class-notes are automatically being generated for their use.

***Index Terms*— real time visual summaries, synchronized speech and text, automatic speech recognition, accessible multimedia, active learning**

I. Introduction

As an emerging educational technology, speech recognition software (SRS) is typically used in education settings to record lectures for the deaf and the hearing impaired. SRS has the potential for many applications in educational settings including a secondary informational resource for all students. Traditional data interface in educational settings between instructor and student is dominated by lecture or "sit-n-get" and assessed via single-loop learning modalities. Most higher educational institutions promote a learning experience that enables self discovery, develops problem solving skills, and endorses deductive reasoning. However, if lecture continues to be the standard in higher education, self discovery, problem solving, and deductive reasoning take a back seat to note-taking skills and data retention.

Active learning and student participation continues as a trend in higher education, in computer science and engineering in our case, that includes reinforcing theories and connecting concepts through classroom exercises [1], [2], [3], [4]. A potential 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 data regurgitation continues as a norm in higher education, it is self evident as to which task students will focus their mental force [5]. SRS has 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 [6]. Format of SRS output has not had much attention as limitations of SRS packages continue to be refined for greater accuracy.

There are two central concerns when considering SRS to enhance double-loop learning in higher education. The first is the accurate recording of spoken words including punctuation and inflection [6]. While some SRS packages extol download-and-play software, the majority of quality SRS packages necessitate several hours to train the software to recognize user speech patterns. The second issue with SRS is the format in which speech is recorded. Does the text format of the recording have a consequence on the effectiveness as a learning tool? We believe this is the case, and in this paper we provide a novel solution to improve the effectiveness of SRS as a learning tool by formatting the text to target individual student learning needs (see Figure 1).

The rest of the paper is organized as follows. We discuss learning patterns in Section II. In Section III we present the history and discuss the state-of-the-art of translation software. In Section IV we discuss trends and obstacles in computer science and engineering teacher-student interaction and the role of our solution in overcoming them. We present the overall speech-to-text-to-summary concept and our speech-to-text-to-sequence solution in detail in section V. In Section VI we present students' reaction to a lecture given using our method. Finally, we conclude with future developments in Section VII and a summary of major contributions in Section VIII.

II. Learning Patterns

Under normal physiological conditions each individual uses their five senses to collect data that the brain electro-chemically sorts, stores, and processes. Once organized, it is the individual mind that creates symbolic representations. Along with the five ubiquitous physical senses, multi-disciplinary research suggests that individuals possess a sixth sense. If we define "sense" as a data collection source that affects the function of the brain then our sixth sense would

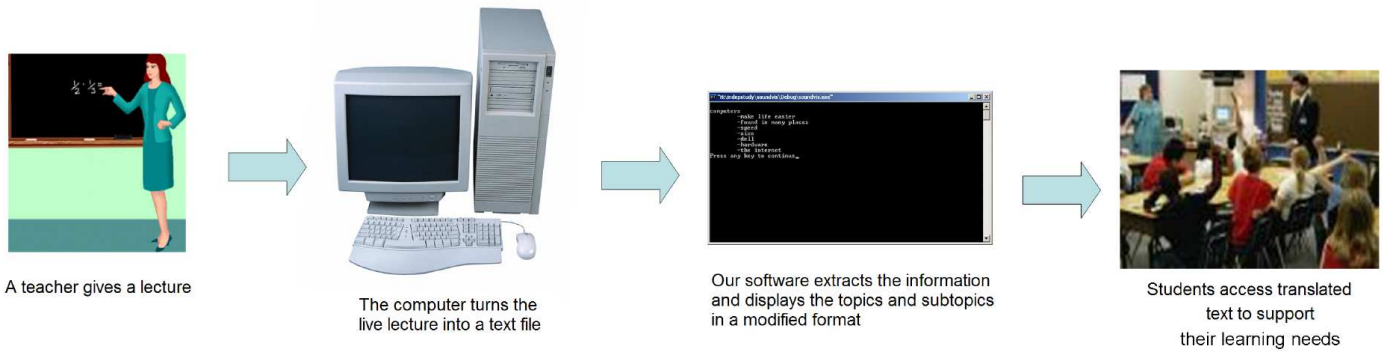

Fig. 1. Speech-to-text-to-summary overall concept.

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 [7].

The Interactive Learning Model© is the cornerstone of understanding integrated learning processes for this initiative [8]. It is the authors' contention 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 (LCI) was developed [9]. 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, if the scale score for Sequence is somewhere at a Use First level. That indicates the students want clear and step-by-step directions, seek time to do work neatly, complete their work from beginning to end without change or interruption, and most importantly want to know if they are meeting the instructors' expectations.

If the scale score for Precise is at a Use First level. That indicates the students' need to ask and answer many questions, want complete and thorough explanations, have a drive to be accurate and correct, enjoy test taking, and seek conclusive documentation.

If the scale score for Technical is at a Use First level. That indicates the students' need to see the relevance and purpose of the task they are completing, seek autonomy, have the drive to understand how things operate/work, and to communicate through action and not words.

If the scale score for Confluence is at a Use First level. That indicates the students' need to think outside of the box, take risks, avoid rules, and have the drive to see situations differently then others do.

Scale scores in the "use as needed range" suggest a moderate need for the aforementioned characteristics and scale scores in the "avoid" range suggest no need or application of those same characteristics.

Research conducted at Rowan University (Glassboro, NJ, USA) revealed that the freshman and sophomore classes of engineering students (N=204) began their learning with the use of the Sequential learning pattern and technical reasoning pattern and used the remaining two learning patterns at the "As Needed" level as determined by the mean and median scores of the LCI. Similar correlation research revealed that the faculty at Rowan (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 [7]. In general the students were searching for clear planning, organization, and consistent expectations from their instructors while the instructors were looking for detailed information and a willingness to take chances from the students. This scenario becomes self evident when you consider the roles of each of the two parties and what is required of those individuals within that system. In order to obtain placement in a regional institution of higher education one needs to have a solid record of grades, discipline, and test scores. In order to obtain the position of University Professor the individual must be successful in an information based occupation and be recognized for performing unique research or instructional practices. There's the rub, two populations with differing sets of mental processes and expectations.

With student persistence and retention being a cascading issue within higher education (particularly in Computer Science and Engineering), providing tools to help students overcome the challenges of systemized learning becomes paramount. The authors contend that if an information translation apparatus was in existence that could coordinate between two or more learning patterns, students will have another tool to understand and apply information that was delivered in a fashion unnatural to their personal learning patterns. 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. The goal of this research was to translate information delivered using a "Precise" approach into a "Sequential" format.

III. 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 [10], [11], speech-to-speech [12], text-to-speech [13], and speech-to-text [14], [6]. 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 [6].

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.

IV. Trends and Obstacles in Computer Science and Engineering Teacher-Student Interaction

Because of their variety, teaching computer science and engineering concepts has proven a difficult task.

Computer programming is related to several fields of technology, and many university students are studying the basics of it. There are many documented instances in the literature on the difficulties novice students experience when learning basic programming concepts and, especially, when trying to apply these concepts to problem solving in areas such as Engineering because the teaching patterns are incongruous with learning patterns [15]. For instance, although beginning students easily master the syntactic details of a programming language, many fail when asked to use that language to solve a concrete problem [16], [17], [18]. Several studies also have evaluated the difficulties in learning programming [19].

Researchers developed web-based visualizations of programming concepts for use in classroom and for supporting independent learning [20]. Researchers have also tried to use animation to help students overcome difficulties experienced by students studying various algorithms. Students who develop good visual representations of the changing data structures gain deeper understanding of the algorithms [21], [22]. Our speech-to-text-to-summary initiative provides potential for using an added visualization tool, in this case a formatted synthesis of verbal instruction.

Recursion is a central concept in computer science and is considered a powerful and useful problem-solving tool. Research studies have concluded that the concept of recursion is difficult to learn and comprehend due to the many layers simultaneous synthesis of data. Moreover, students have difficulties in applying recursion to their problem-solving activities. Comprehending the concept of recursion, and its use to solve problems, is expressed in the ability to evaluate and formulate recursive algorithms [23]. Again, our speech-to-text-to-summary initiative can be embedded in the process of recursion as students can possess the confidence that lectures can be formatted to their natural learning patterns, thus allowing mental effort to focus on synchronization of processes.

Our speech-to-text-to-summary initiative also provides assistance in distance learning, as it complements the existing techniques and technology [24] by providing automated summaries available online.

Learning being the common thread in these trends, education must evolve from an instruction-based model to a learning based model that includes respect for naturally occurring individual learning patterns and systemic approaches to support the needs of the learner via technology.

V. Speech-to-Text-to-Sequence 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.
- 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.

To accommodate sequence learners, our software takes as input the text produced by the speech-to-text off-the-shelf software [14] and formats the output in a bulleted-style list. We use HTML tags for indentation and to emphasize sections of the text.

In the current implementation, 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.

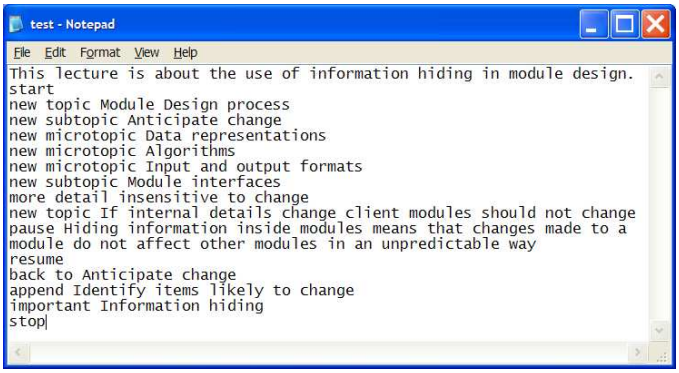

Fig. 2. Sample input file generated by the speech recognition software, directly from the lecture.

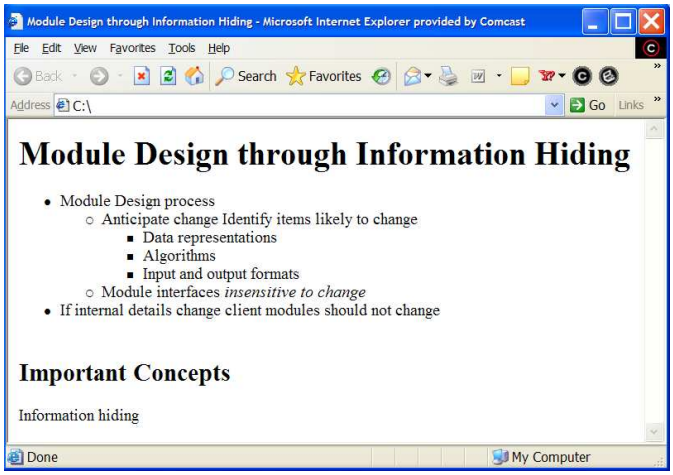

Fig. 3. Sequence-style notes generated by our software, given the input file in Figure 2.

- **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. The lecturer is also able to label specific concepts as important, and these concepts are set apart so the students could easily focus on them. A sample input file for a lecture on the role of information hiding in software engineering design is presented in Figure 2. Its corresponding output, created by our software is presented in Figure 3.

The keywords are stored in a linked list (see Figure 4). Each segment across is a different keyword, and spanning down are the words that make up a compound keyword. Users are given the option to alias different commands. Those commands that

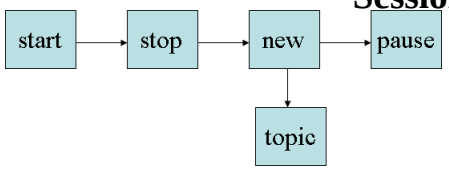

Fig. 4. Commands layout. Commands are in a linked list spanning across for each command and down for the words in the command.

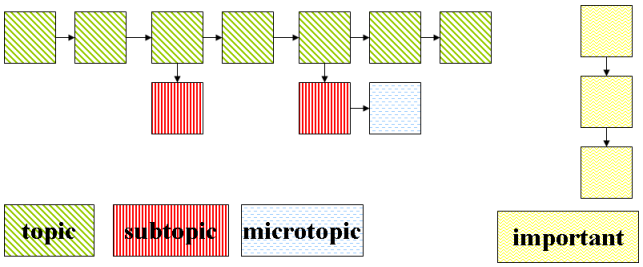

Fig. 5. Topics layout while the program is running. The wide downward diagonal pattern represents topics, dark vertical represents subtopics, dashed horizontal represents microtopics, and zig zag represents important words.

the user makes an alias for are appended to the end of the list. There are four different levels of storage: topics, subtopics, microtopics, and important concept objects (see Figure 5). Each topic is stored next to the previous topic. Subtopics are stored below their topics. Microtopics are stored next to their subtopics. Finally, important concepts are stored in one row, each concept next to the previous one. The difference between topic objects and important concept objects is that important concepts have no need for a second level of depth since they are supposed to be set apart from the rest of the lecture, and they do not have a string allocated for more detail. We have implemented our software in C++, and the current implementation consists of about 1300 lines of code.

VI. Case Study

Our speech-to-text-to-sequence initiative was field tested in an undergraduate computer science lecture-format class, to evaluate the operation of the software package, followed by a survey administered to the participating students. Format keywords were embedded into the twenty minute sample lecture on operating systems concepts to produce a desired speech-to-sequence output. Results of the field test include:

- The third party software (IBM Via Voice [14]) requires significant (more than a couple of hours) user speech training to master accuracy of dialect and inflection.
- Amelioration of extraneous pronouns and conjunctions poses challenges to robust sequential formatting.
- Embedded keywords produced 100% accuracy in signaling modification to text formats.

A brief survey was administered to the 22 students in attendance immediately following the field test using a likert-style forced response where one is ”Less True” and five is ”More True.” Results of the student survey include:

- Mean response of 2.83 to the question ”I found it difficult to understand the lecture when the keywords are being

said" which suggests the embedding of the keywords had neither a positive nor negative effect on comprehension of the content.

- Mean response of 3.67 to the questions "Since the notes are provided at the end I feel more at ease listening to the lecture" suggesting the potential for greater immediate comprehension of the content.
- Mean response of 3.79 to the question "I find the notes generated to be useful" suggesting a potential need for the sequential text format.
- Mean response of 1.94 to the question "I would like it more if it was a direct printout of what was said (no formatting or bullets)" suggesting a potential need for the sequential text format.
- Mean response of 3.47 to the question "I find difficulty in learning to be attributed more to the teacher; rather than the subject" suggesting the bridge between instructor teaching patterns and student learning patterns is as important as content itself.

VII. Future Developments

The initial focus of our initiative was to investigate the technical aspects of modifying text formats to applicable learning patterns and to discover the need for such modifications. The survey data suggests that students are seeking additional resources to aide them in classroom during learning experiences. Future developments include:

- Refining targeted keywords to simplify the speech embedding process.
- Develop technical protocols to eliminate required software speech recognition training and intentional use of predetermined keywords.
- Investigate potential speech-to-text formats that are characteristic of learning patterns other than sequential.
- Consider text translation formats that include other representations via real time web searches including pictures, graphs, designs, and historical documents.

Our current implementation depends on the speech recognition software to translate our keywords properly. If a keyword is not properly translated, our software will not recognize it and it will not process it accordingly. We plan to improve our parser to automatically recognize and correct mistakes in the input file.

The majority of these future developments require careful investigation to ensure compliance with potential trademark and copyright infringement of proprietary software packages.

VIII. Conclusion

There are rich implications in the data and processes developed to date. It has become self evident that learning is a personal process and requires a multi-disciplinary instructional approach without overtaxation of professorial focus. Auto-formatting SRS packages 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 to active learning classrooms by reducing student self-imposed pressure to accurately scribe and organize spoken words toward elasticity to simultaneously connect core concepts.

Acknowledgment

We thank students Vincent Futia and Stephen Ferzetti for their contribution to the current implementation.

References

[1] R. M. Felder, R. Brent. Learning by doing. *Chemical Engineering Education*, 37(4), pages 282-283, 2003.
[2] J. J. McConnell. Active Learning and its use in Computer Science. *Proceedings 1st Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)*, 1996.
[3] B. Timmerman, R. Lingard, G. M. Barnes. Active Learning with Upper Division Computer Science Students. *Proceedings 31st ASEE/IEEE Frontiers in Education Conference*, 2001.
[4] Bhagyavati, S. Kurkovsky, C. C. Whitehead. Using Asynchronous Discussions to Enhance Student Participation in CS Courses. *Proceedings 36th ACM Technical Symposium on Computer Science Education*, 2005.
[5] J. M.Schwartz, S. Begley. The mind and the brain: Neuroplasticity and the power of mental force. *HarperCollins*, New York, NY, 2003.
[6] M. Wald. Using Automatic Speech Recognition to Enhance Education for All Students: Turning a Vision into Reality. *Proceedings 34th ASEE/IEEE Frontiers in Education Conference*, 2004.
[7] G. Dainton, C. Johnston. Learning to use my potential. *Learning Connections Resources, LLC*, Turnersville, NJ, 2006.
[8] C. Johnston. Unlocking the will to learn. *Corwin Press*, Thousand Oaks, CA, 1996.
[9] C. Johnston, G. Dainton. Learning Connections Inventory. *Learning Connections Resources, LLC*, Turnersville, NJ, 1996.
[10] http://www.systransoft.com/index.html
[11] http://www.babylon.com
[12] http://www.lingvosoft.com
[13] http://www.readplease.com
[14] IBM ViaVoice. http://www.nuance.com/viavoice/pro
[15] C. Demetry. Use of educational technology to transform the 50-minute lecture: Is student response dependent on learning style? *Proceedings 2005 ASEE Annual Conference and Exposition*, 2005.
[16] A. Robins, J. Rountree, N. Rountree. Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2):137-172, 2003.
[17] E. Soloway, J. Spohrer. Studying the Novice Programmer. *Lawrence Erlbaum Associates*, Hillsdale, New Jersey, 1989.
[18] R. Lister, E. S. Adams, S. Fitzgerald, W. Fone, J. Homer, M. Lindholm, R. McCartney, J. E. Mostrom, K. Sanders, O. Seppala, B. Simon, L. Thomas. A Multi-National Study of Reading and Tracing Skills in Novice Programmers. *Proceedings 9th Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)*, 2004.
[19] E. Lahtinen, K. Ala-Mutka, H. M. Jarvinen. A Study of the Difficulties of Novice Programmers. *Proceedings 10th Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)*, 2005.
[20] http://www.codewitz.com
[21] V. Ciesielski, P. McDonald. Using Animation of State Space Algorithms to Overcome Student Learning Difficulties. *Proceedings 6th Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)*, 2001.
[22] L. Stern, H. Sondergaard, L. Naish. A strategy for managing content complexity in algorithm animation. *Proceedings 4th Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)*, 1999.
[23] B. Haberman, H. Averbuch. The Case of Base Cases: Why are They so Difficult to Recognize? Student Difficulties with Recursion. *Proceedings 7th Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)*, 2002.
[24] P. Thomas, K. Logan. Observational studies of student errors in a distance learning environment using a remote recording and replay tool. *Proceedings 6th Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)*, 2001.