Improving the Editing Process of Automatically Produced Lecture Transcripts Based on Natural Language Analysis

Miltiades Papadopoulos
Accessibility Research Centre
Teesside University
Middlesbrough
M.Papadopoulos@tees.ac.uk

Elaine Pearson
Accessibility Research Centre
Teesside University
Middlesbrough
E.Pearson@tees.ac.uk

Abstract — **Automatically produced lecture transcripts can act as an alternative to traditional note taking, benefiting those students whose needs and preferences are not met in the traditional learning environment. Nonetheless, despite the substantial progress that has been made in the area of Automatic Speech Recognition (ASR), the performance of ASR systems is still below the levels required for accurate transcription of lectures. This paper describes the development of a tool, which facilitates the evaluation of automatically produced transcription files, based on Natural Language Analysis. This tool is a step forward in the production of meaningful materials for disabled students, with minimal investment in time and effort by academic staff, thereby improving the accessibility of traditional teaching methodologies.**

Accessibility; Automatic Speech Recognition (ASR); Natural Language Processing (NLP)

I. INTRODUCTION

Advancements in learning technology have placed an emphasis on new educational approaches, with a focus on the adoption of flexible means of delivery. Nonetheless, the traditional lecture still remains the most dominant method of teaching, despite growing criticism of its efficiency, flexibility and accessibility [13]. Today's lecture environment isolates students with hearing disabilities, who find it hard to follow speech and therefore are dependent on intermediaries. In addition, students studying in a foreign language and those whose note taking skills are limited find lectures hard to follow and understand. Current legislation requires institutions to offer services that are accessible to students and implies that all staff, academic and support, have a responsibility for providing a learning environment in which disabled students are not disadvantaged [12]. A disability can be defined as a mismatch between the needs of the learner and the learning environment or education delivery [7]. Therefore, there is a growing awareness of the need to improve the accessibility and flexibility of the traditional lecture, in an attempt to fulfil the access requirements of students with disabilities, giving them equal access to higher education, enhancing their learning experience and increasing the quality of the learning process.

Research has shown that Speech Recognition technology can be employed in the University classroom to make lectures more flexible through the use of text transcriptions [15]. Automatic Speech Recognition (ASR) technology can provide transcribed lecture notes as an alternative to traditional note taking, enabling students and staff to concentrate on learning and teaching issues, and in addition, benefit those learners, whose needs and preferences are not met in the current learning environment. Despite the substantial progress that has been made in the area of speech recognition technology, the performance of ASR systems in real lecture situations is still below the required levels. Challenging environments, such as the University classroom, affect the efficiency of current ASR systems and decrease the quality of the produced transcripts [10]. Therefore, lecture transcriptions should be asynchronous, allowing essential editing to the produced files, in order to be usable and meaningful to learners.

This paper discusses the development of a method that significantly reduces the time and effort required to produce accurate lecture transcripts, derived from Automatic Speech Recognition software. Through automatic syntactic and semantic analysis, users are able to quickly and efficiently identify transcription inaccuracies, amend them and accurately target the retraining process of the ASR software. The use of the Semantic and Syntactic Transcription Analysing Tool (SSTAT) provides an easy way for academics to create usable and accessible lecture transcriptions. It represents a step forward in producing meaningful materials for students and addressing the needs of students with disabilities and those studying in a foreign language and thereby enhances their learning experience.

II. AUTOMATIC SPEECH RECOGNITION IN THE LEARNING ENVIRONMENT

A number of innovative approaches have been adopted to supplement lectures through the use of real time captioning and asynchronous transcripts, in an attempt to improve their accessibility. An early study conducted by Leitch and MacMillan [9], involving eight institutions and seventeen lecturers, revealed that the mean accuracy rate of the produced transcripts in real lecture situations was approximately 77%. Wald [14] suggests that the poor results reported from these experiments are due to the fact that

ASR systems are based on models created from written documentation rather than spontaneous speech and that reasonable accuracy rates can only be achieved by committed lecturers after extensive training.

The Villanova University Speech Transcriber (VUST) system was designed to improve the accessibility of computer science lectures with real-time transcriptions. This study evaluated the impact of the VUST system on the effectiveness of a portable, centralised, laptop-based ASR system, designed to augment note-taking by deaf and hard of hearing students in the college classroom [16]. The system was evaluated as a standalone speech transcription system for recognition accuracy and perceived accessibility, and was tested in a controlled environment with pre-prepared lecture materials, as well as in a genuine 90-minute classroom setting. The results suggested that in order for the system's accuracy rates and overall accessibility of the lecture to be satisfactory, sufficient training was necessary. The overall transcription accuracy of the trained system in the classroom setting was 85% [8].

Related work in this area includes an experiment conducted by Papadopoulos & Pearson [10], which measured the performance of trained and untrained ASR software in real lecture situations, and evaluated the quality of the produced transcripts. The results of this study confirmed that current systems are not yet suitable for large-scale employment in the university classroom and while their overall performance does increase after the training process, the quality of the transcription files does not improve significantly. The poor performance of the system was due to the environment's excessive background noise, especially at the start and the end of the lecture, the lecturers' spontaneous and creative speech, the varying proximity to the microphone and finally the quality of the recordings.

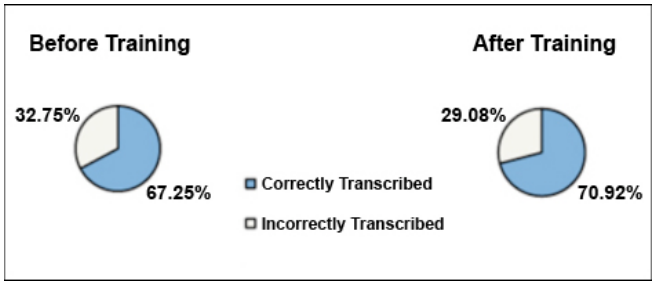

Figure 1: Accuracy rates before and after training

In a lecture situation, speaking rates vary between one hundred and two hundred words per minute [1]; therefore, a mean speaking rate of 150 words per minute translates into 9000 words spoken per lecture. An accuracy rate of 90% indicates 900 transcription errors per lecture, which would require a considerable workload for the editing process of the produced transcript, in order to be truly meaningful to learners. However, not all mistranscriptions need to be corrected. Numerous inaccuracies do not alter the overall meaning of sentences; therefore, editing is not necessary [10]. It becomes clear that it is quite difficult to specify a standard accuracy figure, in order for a transcript to be considered successful and in addition, a method of classifying errors in a way that is meaningful to the academic would significantly improve the efficiency of the editing process.

III. THE SSTAT TOOL

A. Towards an Analyser

Current systems' accuracy rate is arguably the most significant factor, dictating their success or failure in the teaching environment. Higher accuracy rates mean less editing time for academic staff. Systems that produce transcripts that require minimal investment in time and effort for editing purposes are likely to become accepted by academics, while time consuming ones are likely to be rejected. Accepting the fact that neither untrained nor trained systems will produce transcriptions with sufficient accuracy to support disabled students, a tool that will simplify and improve the efficiency of the editing process is a step forward.

The analysis tool needs to be able to identify possible lexical errors, including transcription inaccuracies, false starts and hesitations and also report on syntactical and semantic errors. The tool should be capable of detecting incorrect sentences and in addition, report on what is wrong with them. Taking this as a starting point, it should provide three basic levels of functionality:

- Analyse text and identify erroneous syntactic and semantic transcriptions.
- Classify transcription errors so they can be easily observed and interpreted by academics.
- Attempt to remove lexical inconsistencies, such as false starts, hesitations and repeated words.

B. Modelling the System

Nuance NaturallySpeaking is utilised by the system, as it is one of the general-purpose speech-to-text applications that currently dominate the field of machine recognition. NaturallySpeaking can achieve impressive accuracy rates by trained speakers in controlled environments [2]. Figure 2 demonstrates the basic architecture of the system. Academics utilising the Semantic & Syntactic Transcription Analysing Tool (SSTAT) will be required to carry out a brief initial training procedure, in order to allow the software to get used to their voice, speech pace and accent. The process involves dictation of social and subject-specific pre-prepared scripts.

Once the initial training has been carried out and the lectures have been recorded, the audio files will be run through the ASR software. NaturallySpeaking requires Waveform audio files in order to complete a transcription, and therefore, the recordings must be exported as WAV files. Subsequently, the produced transcripts will be processed by the tool, which will identify the

mistranscriptions and present them to academics in a user-friendly format. The overall editing time for the analysed transcripts created by SSTAT will be significantly lower, than the editing time that is needed for the original transcripts.

In addition, SSTAT produces a document, called 'retargeting text', which includes all the semantic errors that have been identified in the original transcript, and the number of their occurrence. This document is used as the basis for targeting the retraining process of the speech recognition software. Academics need to record the most frequent mistranscriptions and train the software by dictating that list, utilising the 'Add a single word' feature in Nuance NaturallySpeaking. It allows users to add an individual word or phrase to the software's vocabulary. The efficiency and speed of the re-training process is thereby improved.

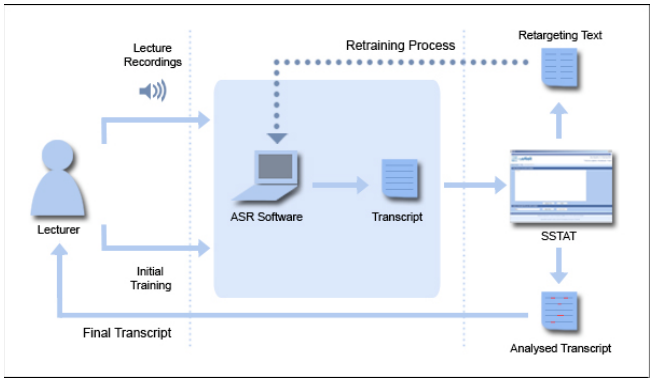

Figure 2: Basic architecture of the system

C. *Transcription Analysis*

In order to provide an evaluation of the produced transcripts, a set of rules that defines the correct, as well as the incorrect sentences needs to be specified. A generative grammar of a language indicates a set of rules that can predict the morphology of a sentence and, in addition, which combinations of words will form grammatical sentences. Based partially on mathematical equations, generative grammar is a set of rules that provide a framework for all the grammatically correct sentences in a language [4; 5]. In addition, the analysis will need to identify any possible lexical inconsistencies and report on transcription errors. Thus, the tool needs to handle both syntactic and semantic knowledge. Prolog was chosen as the programming language, since it is closely tied to the search for computational formalisms for expressing syntactic and semantic analyses for natural language sentences. It is a programming language in which phrase-structure and semantic interpretation rules can be easily expressed. Prolog, in addition, has the advantages of being simple and precise. As it is an Artificial Intelligence language based on formal logic, it has the ability to express both facts and rules. Facts can be thought as explicit knowledge and rules as mechanisms to infer new facts [6].

In our case, a set of Prolog facts includes a large vocabulary and the assigned lexical category of each word in the vocabulary database, as well as phrase-structure and semantic interpretation rules. Consequently, utilising Prolog's grammar notation, a set of Prolog facts that lists the verbs that have been spoken in a lecture could look like:

```
verb(singular) --> [finishes].
verb(plural) --> [finish].
verb(singular) --> [stops].
verb(plural) --> [stop].
```

This would simply tell us that 'finishes' and 'stops' are singular verbs, while 'finish', 'stop' are plural verbs. In the same way, we can create the phrase-structure and semantic interpretation rules.

```
/* Basic Sentence Structure*/
sentence --> np(X), vp(X), fullstop.

/* NP and VP rules */
np(X) --> determiner, noun(X).
vp --> verb(X).

/* Some Vocabulary */
determiner --> [the].
noun(singular) --> [lecture].
noun(plural) --> [lectures].
verb(singular) --> [finishes].
verb(plural) --> [finish].
fullstop --> [.].
```

'np' is a noun phrase and 'vp' is a verb phrase. By giving term arguments we are tying the plurality of the verbs to that of the nouns. Following the same pattern a more elaborate grammar, as well as semantic rules can be created.

Prolog provides the analysis of the lecture transcripts, while the reporting function is presented as a graphical interface developed in Java. In addition, Java is responsible for identifying and removing repeated words, false starts and hesitations in the produced transcripts. Common simple disfluencies include filler words, such as 'um', 'ah' and 'erm', as well as discourse markers like 'you know'.

D. *Reporting Interface*

The primary consideration was to categorise errors, according to their type. The following categories have been formed:

- Semantic errors – mistranscriptions that corrupt or alter the intended meaning of the sentences.
- Syntactical errors – ungrammatical constructs that do not affect the meaning of the sentences.
- Notifications – hesitations, false starts and repeated words.

Consequently, errors need to be demonstrated as a graphical interface, in such a way that they could be easily identified and interpreted. Visualisation is an effective means for exploring and analysing complex data. Color-coding is a fundamental technique for mapping data to visual representations [3; 11]. Therefore, inaccuracies in the analysed text are displayed in a colour-coded manner. This

means that a particular colour will be assigned to each error category. Mistranscribed words in the text are highlighted according to their error type, so that they can be easily interpreted.

SSTAT is targeted for academics. Academics' IT skills vary and therefore, the design should also fit the needs of those whose computer skills are not particularly advanced. Therefore, a comprehensible user-friendly interface is required. SSTAT's main screen can be viewed in Figure 3.

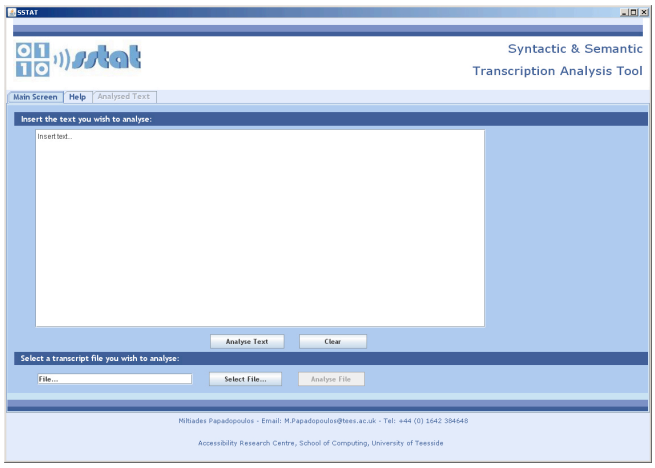

Figure 3: SSTAT's interface

Users can insert the text they want to evaluate, or upload a full transcription file. Taking the simple sentence 'The students is revising' as an example, it can be easily identified that the sentence is syntactically incorrect. If this particular sentence is entered in SSTAT, Prolog analyses it and produces a text file, containing the error that has been identified and the incorrect words.

```
Syntactic Error. students is
```

Java reads the text, processes it and highlights the words according to their error type (Figure 4).

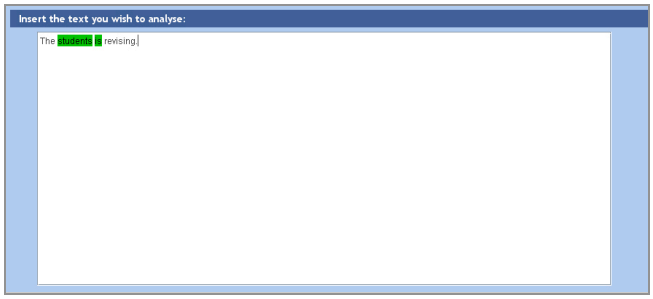

Figure 4: Analysed text

Once the analysis has been performed, users will be able to either print the analysed transcript or save it as an HTML file. The analysis panel (Figure 5) contains detailed information about the nature of mistranscriptions. Users can choose to view the exact number of errors for any category and opt to remove notifications. The information about the number of inaccuracies in the panel is also colour-coded. Visual support provides memory links so that information is better retained. It will, therefore, enable users to memorise the error category that each colour is assigned to.

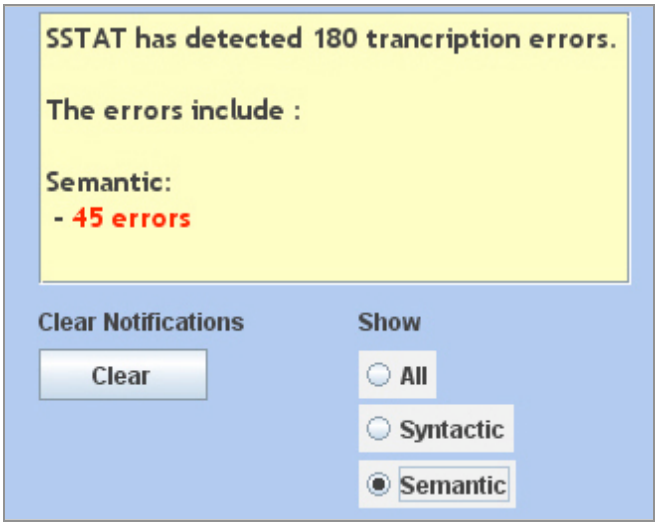

Figure 5: The analysis panel

IV. Preliminary Test Results

A number of experiments has been conducted in an attempt to evaluate the efficiency of SSTAT. The intention of the first experiment intended to:

- Examine whether and how the utilisation of SSTAT reduces the overall effort and time needed for the editing process of the lecture transcripts and,
- Investigate whether by addressing the most common transcription errors, the retraining process of the ASR software can be efficiently targeted.

The experiments were conducted by the researchers and the developers that were involved in the implementation of the tool. The transcriptions used were based on two lecture recordings of a postgraduate-level module in the area of Information Technology. Lecture A was a 45-minute presentation, while the duration of Lecture B was approximately 40 minutes. Both lectures were given by native English speakers. The transcriptions that were used had been produced by the trained ASR system.

In order to determine whether the workload was reduced through the use of tool, the editing time for the original transcripts was calculated and compared to that of the transcripts produced by SSTAT. The analysis of the results demonstrated that the editing time for the analysed transcripts was significantly lower than that of the original ones (Figure 6). Editing for each inaccuracy was mostly a straightforward process. However, in cases where the meaning of a sentence was altered, editing was more challenging, since users had to either guess the correct word or listen to the lecture recording. The overall editing time for Transcript A was one hundred and ten minutes, while the utilisation of the tool reduced it to sixty one minutes. The editing time for Transcript B was reduced from ninety to

fifty two minutes. In both cases, the amount of time needed to correct the errors was reduced by approximately 43%.

Transcript	Editing Time (without analysis)	Editing Time (with analysis)	Decrease (%)
Transcript A	110 minutes	61 minutes	44.5%
Transcript B	90 minutes	52 minutes	42.2%

Figure 6: Evaluation results - Editing time needed for each transcription file

Most transcription errors, especially those affecting the meaning of sentences, were subject specific words. The retraining process was based on the 'retargeting text' and the most common inaccuracies were added to the vocabulary of the ASR system. An interesting observation was that once the retraining was carried out, most of the incorrectly transcribed words were transcribed correctly when the recording was run through the ASR software for the second time.

V. Summary & Future Work

Research shows that current ASR systems cannot yet produce efficient transcriptions in the learning environment, in order to address the accessibility problem of the traditional lecture. Reducing the time required to produce an acceptable quality of transcripts is the way forward. The unique aspect of this study is the implementation of a tool, which will provide an easy way for academic staff to create usable lecture transcriptions. The results of the experiments are promising, while tests to-date confirm that the editing time is reduced considerably using the analysed transcripts.

The next stage of the research requires an extensive evaluation of SSTAT to gain more concrete results about its efficiency in real lecture situations and, in addition, aims to investigate the views of academics towards its validity. An updated version of SSTAT is currently being produced. The new version will be based on matching algorithms, in order to suggest possible corrections for mistranscribed words.

Finally, there are accessibility problems associated with colour-coding techniques and consequently with SSTAT. A software program that requires users to distinguish between identical shapes of different colours, could pose problems to people with vision impairments. Alternative approaches of determining transcription errors and conveying information will be examined. Potentially effective color scales, which can provide a range of colours varying in hue, saturation and brightness or additional tools enabling users to adapt colour scales according to their needs, will also be examined.

References

[1] K. Bain, S. Basson, and M. Wald, "Speech Recognition in University Classrooms: Liberated Learning Project," Proc. 5th International ACM Conference on Assistive Technologies (Assets 02), ACM Press, Jul. 2002, pp. 192-196.

[2] S. Bennett, J. Hewitt, D. Kraithman, and C. Britton, "Making Chalk and Talk Accessible," Proc. 2003 Conference on Universal Usability (CUU 03), ACM Press, Nov. 2003, pp. 119-125.

[3] J. Bertin, Simiology of Graphics. The University of Wisconsin Press, 1983.

[4] K. Brown (Editor), Encyclopaedia of Language and Linguistics, 2nd ed., Oxford: Elsevier, 2005.

[5] N. Chomsky, "Three Models for the Description of Language," I.R.E.Transactions on Information Theory, vol. 2, no. 3, pp. 113-123 Sep. 1956, doi:10.1109/TIT.1956.1056813

[6] S. Green, E. Pearson, and S. Gkatzidou, "Formal Specification of an Adaptable Personal Learning Environment Using Prolog," Proc. 1st ACM SIGMM International Workshop on Media Studies and Implementations that Help Improving Access to Disabled Users (MSIADU 09), ACM, Oct. 2009, pp. 29-38.

[7] IMS, IMS Global Learning/ Dublin Core AccessForAll Project. 2004. Available: http://www.imsglobal.org/accessibility

[8] R. Kheir and T. Way, "Inclusion of Deaf Students in Computer Science Classes Using Real-Time Speech Transcription," Proc. 12th Annual Conference on Innovation & Technology in Computer Science Education (SIGCSE 2007), ACM Press, Sep. 2007, pp. 261-265.

[9] D. Leitch and T. MacMillan, Liberated Learning Initiative Innovative Technology and Inclusion: Current Issues and Future Directions for Liberated Learning Research, Year III Report, Nova Scotia: Saint Mary's University, 2003.

[10] M. Papadopoulos and E. Pearson, "An Analysing Tool to Facilitate the Evaluation Process of Automatic Lecture Transcriptions," Proc. World Conference on E-Learning in Corporate, Government, Healthcare and Higher Education (E-Learn 2009), AACE, Oct. 2009, pp. 2189-2198

[11] P. Schulze-Wollgast, C. Tominski, and H. Schumann, "Enhancing Visual Exploration by Appropriate Color Coding," Proc. 13th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG 2005), UNION Agency – Scince Press, Feb. 2005, pp. 203-210.

[12] SENDA, Special Educational Needs And Disability Act. 2001. Available: http://www.opsi.gov.uk/acts/acts2001/20010010.htm

[13] Smith, A., Ling, P. and Hill, D. 2006. The Adoption of Multiple Modes of Delivery in Australian Universities. Journal of University Teaching and Learning Practice. 3, 2, 67-81.

[14] M. Wald, "Using Automatic Speech Recognition to Enhance Education for All Students: Turning a Vision into Reality," Proc. 34th Frontiers in Education Conference (FIE 2005), IEEE Press, Oct. 2005, pp. 22-25, doi:10.1109/FIE.2005.1612286

[15] M. Wald and K. Bain, "Universal Access to Communication and Learning: The Role of Automatic Speech Recognition," International Journal Universal Access in the Information Society, vol. 6, no. 4, pp. 435-447, Feb. 2008.

[16] T. Way, R. Kheir, and L. Bevilacqua, "Achieving Acceptable Accuracy in a Low-Cost, Assistive Note-Taking, Speech Transcription System," Proc. IASTED International Conference on Telehealth/Assistive Technologies (Telehealth/AT 2008), ACTA Press, pp. 72-77.