Interface Design Strategies for Computer-Assisted Speech Transcription

Saturnino Luz  
Dept. of Computer Science  
Trinity College Dublin  
Ireland  
luzs@cs.tcd.ie

Masood Masoodian  
Dept. of Computer Science  
The University of Waikato  
New Zealand  
masood@cs.waikato.ac.nz

Bill Rogers  
Dept. of Computer Science  
The University of Waikato  
New Zealand  
coms0108@cs.waikato.ac.nz

Chris Deering  
Dept. of Computer Science  
Trinity College Dublin  
Ireland  
deeringc@cs.tcd.ie

ABSTRACT

A set of user interface design techniques for computer-assisted speech transcription are presented and evaluated with respect to task performance and usability. These techniques include error-correction mechanisms which originated in dictation systems and audio editors as well as new techniques developed by us which exploit specific characteristics of existing speech recognition technologies in order to facilitate transcription in settings that typically yield considerable recognition inaccuracy, such as when the speech to be transcribed was produced by different speakers. In particular, we describe a mechanism for dynamic propagation of user feedback which progressively adapts the system to different speakers and lexical contexts. Results of usability and performance evaluation trials indicate that feedback propagation, menu-based correction coupled with keyboard interaction and text-driven audio playback are positively perceived by users and result in improved transcript accuracy.

Categories and Subject Descriptors

H.5.2 [User Interfaces]: Natural Language

General Terms

Human Factors; H.5.2 [User Interfaces]: Voice I/O; H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems

Keywords

Automatic speech recognition; error correction; computer-assisted speech transcription

1. INTRODUCTION

Automatic speech recognition (ASR) is a mature technology which has found application in diverse areas. However, speech transcriptions generated by modern ASR systems are often far from perfect. Although for certain applications, such as interactive voice response, potentially adverse effects of high recognition error rates can be effectively mitigated through appropriate interaction design, transcription applications usually require more substantial user involvement in order to produce satisfactory results. This paper focuses on systems that combine ASR and user interface design techniques to support effective creation of long transcripts by novice transcribers, also known as computer-assisted (or computer-aided) speech transcription systems.

The importance of computer-aided speech transcription, as opposed to fully-automated (ASR) transcription, and the need for effective user interfaces to support it have been increasingly recognised. This is specially true of application domains, such as medical and legal setting, where high accuracy demands make human post-editing of transcripts a practical necessity. In the area of health informatics, for instance, this trend towards computer-aided transcription is apparent even in the titles of articles published in this area over the years. They have ranged from an optimistic “Computer-based speech recognition as a replacement for medical transcription” in 1998, to a more realistic “Computer-based speech recognition as an alternative to medical transcription” in 2001, to a representative of the emerging consensus in this area: “Speech recognition as a transcription aid” published in 2003 (emphasis added).

In addition to health care and law applications, the considerable growth in availability of recorded speech and multimedia resources on the Internet, ranging from videos of various sorts to dedicated news channels and repositories of

In court record production, the term “computer-assisted transcription” is also used to denote a method of stenographic reporting in which a computer functions as an enhanced stenographic machine. Such methods do not usually involve ASR and therefore bear no relation to the techniques described in this paper.
recorded broadcasts, has created a demand for fast and effective transcription also in these domains. Automated speech recognition systems can play an important role in meeting this demand. Requirements for these kinds of transcription tasks vary from application to application. In the case of news broadcasts, for instance, approximately accurate but quickly produced “rush transcripts” are often sufficient [12]. However, except perhaps for applications in the area of information retrieval, where the transcript is not actually meant to be read in its entirety (e.g. [6, 18]; see also [7] for a survey) a minimum of understandability is a basic requirement. While state-of-the-art speech recognition systems can deliver intelligible transcripts, human intervention is often required.

The approach to computer-assisted transcription we propose and investigate in this paper essentially consists in the following. An ASR system is initially employed to generate a (typically imperfect but time-aligned) transcription of a long speech recording. From that point on, fine tuning is performed through iterations of human-driven error correction steps supported by an interactive system. This approach therefore shifts the system’s focus from automatic recognition to user interface support for error correction. We explore a number of user interface design strategies to support the creation of time-aligned transcriptions of speech recordings. These techniques have been implemented through a prototype “transcript editor” system and evaluated for usability and effectiveness.

2. RELATED WORK
User interface support for error correction in speech applications varies greatly. Error correction methods employed in dictation systems [21], for instance, is very different from repair in dialogue systems [10]. Such techniques usually involve error recognition via the visual modality and repair through speech interaction (“re-speaking”) [1] or, more recently, through combinations of modalities [24, 3]. Interface design for computer-assisted transcription seems to have received less attention than dialogue or dictation systems. However, techniques developed for those types of systems, specially dictation systems, might be profitably employed in speech transcription. It is somewhat surprising that, although techniques like alternative lists [1, 16] and multimodal interaction [21] can apparently benefit transcription tasks, little usability and human factors research has been done for such applications. One could speculate that this is due to the fact that the distinction between ASR-enhanced transcription of pre-recorded speech and dictation is blurred by the focus on error-correction strategies. Focusing entirely on error-correction obscures certain aspects of transcription tasks that differ considerably from dictation and could be exploited through user interface design in order to improve task performance. One such aspect is that, in computer-assisted transcription, one can take advantage of the fact that the entire speech to be transcribed is often available (in recorded form) when the transcription task starts, allowing the system to propagate changes dynamically and take maximum advantage of user feedback.

The usual form of representation of the speech signal on the user interface is as a waveform or a spectrogram stretching along a timeline. Some speech transcription systems, notably those used by speech scientists for transcribing and segmenting speech records for further processing or corpora production, employ this interface style, allowing access to the audio to be transcribed via horizontal scrolling. The Transcriber [1] is a system that uses that strategy. Since its primary goals is annotation of the speech signal, the local nature of the transcription (and alignment) suits the sliding-window view afforded by the timeline representation. Other systems that specifically target annotation tasks, but not necessarily at the speech signal level, such as the Elan tool [15], also employ a similar interface style. In the latter case, however, since the typical use of such systems is dialogue annotation, it is not clear whether displaying the annotation on a single horizontal timeline would not prove a hindrance.

A departure from the timeline metaphor is illustrated by the system described in [17], which employs a table-like visualisation to display the alternatives considered by the ASR component as a confusion matrix. This system incorporates correction propagation within sentence boundaries through re-scoring of the recognition lattice. The authors report good results in term of accuracy and usability when the system is tested against a baseline (keyboard input only) system, even though the tabular presentation restricts the amount of text that can be displayed on the screen. More recently, a system [17] has been proposed which allows users to correct an ASR-generated transcript through handwriting input. This system is designed for non-expert transcribers and does not necessarily rely on a timeline metaphor. Although the system aligns the audio signal to the text strings it generates internally, no feedback of this alignment operation is presented to the user.

Finally, a two-stage strategy has been proposed [9] for transcription which combines automatic speech recognition (ASR) and a human transcriber. It consists in first manually generating an approximate transcript and then employing ASR to align this manually generated transcript to the speech signal, often correcting transcription errors in the process. This strategy is therefore the opposite of the approach we described in this paper. Furthermore, it targets a different user group in that it is aimed at experienced human transcribers, who are able to manually produce reasonably accurate transcripts in real time. Our system, on the other hand, caters for the novice or occasional transcriber.

3. DYNAMIC TRANSCRIPTION AND ERROR CORRECTION
Unlike most audio editing and transcription tools [11], ours has a user interface that is based on a text editor metaphor augmented with components for visualisation and interactive manipulation of the input speech signal. The user “loads” the transcript onto the editor screen by selecting an audio file and running it through the recogniser. The system then allows the user to correct transcription errors manually. User feedback in the form of error correction is then used to dynamically update the recogniser’s vocabulary and speaker model, thus reducing the likelihood of later occurrences of the same transcription errors during the recognition process. As the transcription task progresses user feedback repeatedly causes such dynamic updates, thus gradually reducing the number of transcription errors at each iteration and therefore the amount of manual intervention required.
The speech transcript editor is shown in Figure 1. The editor shows the waveform representation of the input signal above the transcription (Figure 1 b). Depending on the nature of the application (e.g. creation of rush transcripts versus creation of an aligned speech corpus), the user may reduce or increase the zooming level of the waveform component (i.e. time scale and waveform amplitude), and add speech and text boundary markers. These markers can be altered, words sequences can be merged into a single word, and words can be split into sequences. Words that have been assigned low confidence scores by the recogniser are highlighted on the screen. Selecting such words activates a menu of alternative transcriptions. The user can also select specific segments and play the portions of the speech recording associated with the selected segments. This has been carefully integrated with the keyboard commands to allow the user to smoothly move between the listen/check and transcript correction processes.

According to the classification of errors that occur in ASR-enabled and human transcription presented in [23], transcription errors can be due to: speaker mispronunciation (annunciation errors), out-of-vocabulary terms (dictionary errors), misrecognition of the appropriate form of a word (suffix errors), substitution of homonyms (homonym errors), omission of spoken words (deletion errors), spelling errors, failure to capture the meaning of sentences in context (non-sense errors), and ambiguous expressions which could lead to misunderstanding on the part of the reader (critical errors). Our system incorporates distinct prevention and correction mechanisms to deal with these potential sources of errors. Prevention of critical and nonsense and deletion errors is facilitated by the running-text layout of the transcription, which helps the human transcriber visualise the broader context of the passages being transcribed. Correction of annunciation and dictionary errors is dealt with directly through the correction propagation functionality. Menu-based correction and auto-completion mechanisms attempt to minimise spelling, suffix and homonym errors. Annunciation, spelling and homonym error correction is also addressed through re-scoring of the recognition lattice or, in certain cases, re-recognition. If the file is large enough, user preferences for certain spelling variants can be recorded so that their scores can be boosted in the existing word lattices, potentially altering the transcript, or incorporated into the language model for future runs of the recogniser.

Commercial human transcription services often find it helpful to be able to “clean up” the audio as well as the transcript by removing noise, speech dysfluencies, and sometimes even whole sentences [9]. Our speech transcript editor supports this type of task by offering the option of editing the audio at the same time as the transcript.

Other forms of support for audio file editing and annotation found in tools such as Audacity [3], Wavesurfer [20] and Transcriber [4] (e.g. support for various signal processing algorithms, spectrogram display, multilevel annotation etc) has not been implemented in the current version of the system. Therefore this aspect of the tool has not been explicitly addressed in the evaluation reported below. However, we believe that the interaction style we propose to be well suited to audio alignment and editing tasks such as those targeted by existing specialised audio annotation tools.

4. EVALUATION
A two-stage evaluation method was employed in order to assess the effectiveness of the user interface techniques implemented in the transcript editor. The first stage consisted of an analytical evaluation of the effects of propagating user corrections on transcription accuracy. This was followed by a second stage consisting of a series of user trials designed to assess how the new interface functionality is perceived by the users, as well as the potential impact of that func-
The motivation for this two-stage strategy is admittedly practical. Since dynamic correction propagation can only be expected to have a measurable effect on relatively large transcripts, a meaningful empirical evaluation of this feature would require subjects to work on much longer tasks than the limited amount of time available for an experiment conducted in a usability laboratory would allow. We therefore decided to restrict the assessment of this feature to running a full-featured version of the speech transcript editor on a large audio file with focus on misrecognition due to pronunciation variants and out-of-vocabulary words. On the other hand, the assessment of the usability of error correction strategies implemented at the user interface level was performed on a stripped-down version of the system which had the dynamic correction propagation disabled, in order to prevent potential effects of this feature from interfering with the results. A more comprehensive longitudinal study of the fully functional system in more realistic settings is planned as future work.

4.1 Analytical evaluation: the effect of propagating user corrections

The purpose of this preliminary evaluation exercise was to assess the effects of dynamic propagation of word additions on the accuracy of the system’s underlying speech recogniser. For this test we focused on the fact that one of the main difficulties speech recognisers face is recognising proper nouns, which tend to occur regularly in cases such as broadcast news, medical and legal applications. This happens because such nouns are often out-of-vocabulary words (dictionary errors) or vary greatly in pronunciation (certain types of spelling and homonym errors).

For this experiment, six audio files were selected from the CMU ARCTIC database. The speech files chosen were produced by two US female speakers, two US male speakers, a Canadian male speaker and a Scottish male speaker. Each speech file contained 1,132 sentences, consisting of around 10,000 words. Five proper nouns (“Philip”, “Saxon”, “Gregson”, “Jeanne”, “MacDougall”) were chosen which together occurred 76 times in each of the recordings (accounting for 0.8% of the transcribed content). One instance of each of these nouns from one of the US female speaker’s recording was used to train the system. In the case of the words “Gregson” and “Jeanne” which were not present in the system’s initial vocabulary, the respective entries and phonetic spellings had to be added during the test.

Accuracy scores for the tested files and targeted vocabulary items per speaker before and after just four iterations of error correction are shown in Figure 2. Addition of out-of-vocabulary words, specification of alternative phonetic spellings and training of the recogniser on the selected words improves recognition across different speakers on average from 41% to 90%. This clearly indicates the benefits of adding words and training on even a single case of each word. Further investigation of the errors after word additions and training in this small test set shows that 100%

Figure 2: Accuracy of proper noun recognition by speaker before and after four error-correction iterations

accuracy can be achieved by additionally including the possessives for each of the missing nouns. While it is unlikely that such optimal outcomes will hold for speech recorded under less favourable conditions, the test nevertheless shows that correction propagation is a viable repair technique for speech transcription.

4.2 Usability evaluation

In order to further assess the effectiveness and usability of the user interface features implemented, we designed a within-subjects experiment in which each subject was asked to transcribe two short audio recordings using alternatively a baseline ASR-assisted transcription system and a transcription prototype with functionality similar to that of the system described in Section 4. For this test we focused on the fact that one of the main difficulties speech recognisers face is recognising proper nouns, which tend to occur regularly in cases such as broadcast news, medical and legal applications. This happens because such nouns are often out-of-vocabulary words (dictionary errors) or vary greatly in pronunciation (certain types of spelling and homonym errors).

The green editor consists of a plain text editor coupled to an audio playback component (Figure 3) which allows playing, pausing and jumping to different parts of the recording through a slider. The user transcription process is initiated when the user selects an audio file and presses the green arrow at the bottom of the editor’s window. The audio file is run through the speech recogniser and the recognition results appear on the editor pane, along with sentence markers and “noise phones” such as +COUGH+, +Uh+ etc. The subjects were instructed to simply ignore those symbols. Once transcribed sentences start to appear on the editor pane, users can begin correcting them by listening to the audio (using the audio playback component) and freely editing the transcript using keyboard and (possibly) mouse.

http://festvox.org/cmu_arctic/
We hypothesised that the transcript correction strategy users would adopt when using the green editor would be to re-play the audio sequentially while inspecting the initial transcription results, stop the audio playback to make corrections as needed, resume the playback, and so on.

The blue editor (Figure 4) implements essentially the same functionality as the fully-fledged system, except for dynamic error propagation and audio waveform display, as mentioned above. Display of aligned waveforms has been omitted because this feature targets more specialised user groups, such as linguists and speech scientists, and would find no natural counterpart in the baseline (green) transcriber. Although the waveform components are not displayed, the blue editor can display timestamps around each token, as shown in Figure 4, thus providing feedback on how speech and text are actually aligned. Similarly, the word splitting component (Figure 5) allows the user to find the correct alignment by moving a slider along the timeline below the word segments. The system initially attempts to guess the relative length of each typed subsequence by looking up a table of phonetic symbols and allocating time proportionally to the duration of the symbol sequence. Where precise alignment to the audio signal is needed, the user can define the sequence alignment by adjusting it through the horizontal slider while replaying the audio. For this trial, however, users were told that alignment was not part of the task and that these features could be ignored. Accordingly, the editor used in the trials was set up so as to hide the timestamps.

The functionality tested therefore encompassed: the editor-controllable playback feature, the highlighting of low-confidence segments, the drop-down menus for choosing among recognition alternatives, the auto-complete facility, the word merge and split functions and the keyboard shortcuts. As with the full-fledged prototype described in Section 3 clicking on a highlighted segment (or selected sequence of segments) on the blue editor activates a menu which displays the recognition alternatives according with the current ASR word lattice as well as the possible actions allowed by the editor depending on the context. These actions include merging of selected words into a single word, splitting of a single word into a sequence of words and manual correction. Both the word splitting and the manual editing components have word completion capabilities which simultaneously accesses the recogniser’s vocabulary and the phonetic symbol table. The system also allows the user to enter their own phonetic spellings, overriding the initial suggestion, if needed.

All trials were conducted on a computer lab, using the same hardware configuration. A total of 14 subjects took part in the evaluation. Twelve of these subjects were undergraduate students and the remaining two were post-doctoral researchers, all native speakers of English. They were given headphones and two printed documents containing brief descriptions of the blue and green editors. The trial started with an explanation of the task and a short tutorial on how to operate each version of the software. The subjects were then given a short audio file to transcribe using both editors.
to become familiar with their functionality. This familiarisation phase used a sample audio file different from the audio file used for the actual trial. Once the users felt comfortable enough with the software they were asked to transcribe one audio file using the green editor and another using the blue editor. The order in which the editors were to be used and the audio file to be used with each editor were selected randomly for each trial so as to minimise potential order effects.

We assessed transcription task accuracy for these trials in terms of word error rate (WER), macro precision ($\pi_M$) and macro recall ($\rho_M$) scores. The WER score is defined as the ratio between the minimum number of (word) insertions, deletions and substitutions needed to convert the transcript into its aligned reference text (Levenshtein distance) and the total number of words in the reference text. Alternative spellings of certain reference words and phrases were allowed in the user-produced transcripts during alignment to avoid arbitrarily penalising legitimate spellings entered through free typing (green editor) as opposed to menus. Examples of such allowed alternatives include: “thorough going” and “thorough going”, “upper hand” and “upperhand”, etc. The precision and recall scores were also included in order to provide a more nuanced view of user performance, following the approach proposed in [13]. Precision is defined for a single word as the proportion of occurrences of that word in the transcript which matches the corresponding word positions in the reference. Formally, $\pi_i = \frac{|T_i \cap R_i|}{|T_i|}$, where $T_i$ and $R_i$ are the sets of positions occupied by word $i$ in the transcript and reference text respectively. Conversely, recall indicates the proportion of the reference that is correctly matched in the transcribed text, or $\rho_i = \frac{|T_i \cap R_i|}{|R_i|}$. The global macho scores are obtained by averaging over the scores for each word in the vocabulary of the transcript (for $\pi_M$) and reference (for $\rho_M$).

The two audio files used in this experiment were selected from the Voxforge corpus and assessed as being of similar length and difficulty. They contained readings of English texts by a native speaker. The total number of words in the reference transcripts for these files were 288 and 278. The initial word error rates (WER) for ASR-generated transcripts for these files were 28.8% and 21.5%.

Users were allowed to spend as much time as they needed on the task. They were asked to try to transcribe as accurately as possible, and instructed to press the save button when they were satisfied with the transcript they had produced. The trials were visually monitored by the experimenter. Transcription times were recorded automatically by the system and accuracy statistics were compiled at the end of the experiments. At the end of each session, users were asked to fill out an evaluation questionnaire, and this was followed by a brief interview. The questionnaire included questions that asked the users to rate (on a 5-point Likert scale ranging from “strongly disagree” to “strongly agree”) certain features of the blue editor in terms of helpfulness and usefulness, compared to the green editor. These questions are shown in Table 1.

| 1. The ability to choose between alternative words from a menu was helpful |
| 2. Being able to play back the audio for selected parts of the text was helpful |
| 3. Using the menu-based editing in the blue editor was as efficient as typing freely in the green editor |
| 4. The auto-complete function was helpful |
| 5. I found the keyboard shortcuts in the Blue editor beneficial |
| 6. I enjoyed using the blue version |
| 7. I enjoyed using the green version |

In addition, users were asked to estimate their usage of keyboard shortcuts and the menu for correcting words, and to indicate which of the editors they preferred using, if any.

![Figure 6: Answers to the questionnaire regarding the usability of error correction mechanisms.](http://www.voxforge.org/)
question numbers shown in the aforementioned table.

Figure 6 shows that the ability to play back the audio for selected parts of the text (question 2) was the feature the users found most valuable. The pop-up menus and keyboard shortcuts (questions 1 and 5) were also positively evaluated. Users were divided as to whether using menu-based editing in the blue editor was as efficient as typing freely in the green editor (question 3), with a trend towards disagreement. Perhaps surprisingly, most users were neutral with respect to the auto-complete functionality. The following is typical of the comments made by the users: “I felt the Blue version was more efficient than typing freely in the Green editor. Both editors were useful - but I preferred the Blue with right menu options. I found the keyboard shortcuts less useful than expected - but I expect I would use them more, with practice.”

User observations showed that, as we had hypothesised, the preferred strategy adopted by users when using the green editor was to play the audio and correct sequentially. Contrary to our expectation, however, a predominantly sequential strategy was also adopted when the blue editor was used. This is probably due to the fact that, not having heard the audio recording prior to the beginning of the transcript correction task, users could not assess how reliable the ASR-generated transcript was to begin with, and therefore were unable to take full advantage of the visual cues and random audio access facilities provided by the blue editor. This also suggests as to why the users were slightly faster when using the green editor, despite the fact that most users preferred the blue editor.

Table 2 summarises the performance results in terms of average word error rate (WER) for the final transcripts, the percentage of errors corrected per editor, the macro precision and recall scores, and the transcription speeds expressed in words per minute (WPM). As regards task performance, results indicate that the user interface components employed by the blue editor produced more accurate final transcripts at the expense of transcription time. Furthermore, the higher differences found in the \( \pi_M \) scores indicate that most remaining errors in transcripts produced with the green editor were due to user failure in spotting misrecognitions.

The speed results can be partially explained by the task setup, which precluded listening to the audio while ASR was taking place, as explained above. Another contributing factor, however, is the perceived “ease” of one of the texts (text 2) which required less edits and therefore less menu-based interaction. As Table 2 and Figure 7 show, users editing text 2 with the green editor in general took less time than those who used the blue editor, even though the error rates for the former were quite spread while the latter tended to cluster towards the lower end of the error scale. For text 1, the completion time differences were less evident but the blue was clearly superior in terms of accuracy.

### Table 2: User performance comparison

<table>
<thead>
<tr>
<th></th>
<th>blue editor</th>
<th>green editor</th>
</tr>
</thead>
<tbody>
<tr>
<td>text 1</td>
<td>WER</td>
<td>4.49%</td>
</tr>
<tr>
<td></td>
<td>% corrected</td>
<td>82%</td>
</tr>
<tr>
<td></td>
<td>speed (WPM)</td>
<td>22.4</td>
</tr>
<tr>
<td>text 2</td>
<td>WER</td>
<td>3.81%</td>
</tr>
<tr>
<td></td>
<td>% corrected</td>
<td>86.4%</td>
</tr>
<tr>
<td></td>
<td>speed (WPM)</td>
<td>24.2</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>4.30%</td>
</tr>
<tr>
<td></td>
<td>( \pi_M )</td>
<td>0.85</td>
</tr>
<tr>
<td></td>
<td>( \rho_M )</td>
<td>0.87</td>
</tr>
<tr>
<td></td>
<td>% corrected</td>
<td>83.4%</td>
</tr>
<tr>
<td></td>
<td>speed (WPM)</td>
<td>23.3</td>
</tr>
</tbody>
</table>

Figure 7: Error rates per user (after user editing) against task completion time expressed as ratio of completion time to maximum completion time per text.

### 5. CONCLUSION

The results of evaluation show that the error correction mechanisms implemented by the prototypes described in this paper (and related systems) can potentially improve user performance at computer-assisted transcription tasks. Dynamic propagation of corrections can improve performance by modifying the working transcript continuously as error correction feedback is received by the system. Other user interface improvements such as menu-based correction (coupled with keyboard shortcuts) and text-driven playback are also well received by users and likely to result in more accurate transcripts.

However, the results of usability evaluation also suggest that certain design improvements need to be made in order for the new functionality (blue editor) to match the naturalness of free typing (green editor) for transcript correction tasks. This is indicated by the perception expressed by some users (through the questionnaires and interviews) that menus and other mouse-activated components tended to slow them down while producing little benefit. For users with this sort of profile, which is likely to include expert
users and professional transcribers, interface designs that allow for keyword-controlled menus activated automatically as the cursor moves over low-scoring words and merge-split functions embedded in the editor pane might help increase transcription speed.

We plan to conduct further experiments with a more focused user group and task (e.g. speech corpus production) in order to evaluate the segmentation, alignment and waveform display features. We have also devised alternative forms of animated visualisation for transcripts (described in detail elsewhere) which we intend to incorporate into the transcription editor in order to encourage users to adopt a non-sequential correction strategy, which could be particularly useful in contexts where transcripts serve an information retrieval purpose in conjunction with other audiovisual sources.

6. ACKNOWLEDGEMENTS
S. Luz’s research was part-funded by Science Foundation Ireland through the CNGI/CSET grant.

7. REFERENCES