An Integrated Toolkit Deploying Speech Technology for Computer Based Speech Training with Application to Dysarthric Speakers

Athanassios Hatzis, Phil Green, James Carmichael,
Stuart Cunningham+, Rebecca Palmer°, Mark Parker°, and Peter O'Neill+

Department of Computer Science, University of Sheffield, United Kingdom
+ Barnsley District General Hospital NHS Trust, Barnsley, United Kingdom
° Institute of General Practice and Primary Care, University of Sheffield, United Kingdom

Abstract

Computer based speech training systems aim to provide the client with customised tools for improving articulation based on audio-visual stimuli and feedback. They require the integration of various components of speech technology, such as speech recognition and transcription tools, and a database management system which supports multiple on-the-fly configurations of the speech training application. This paper describes the requirements and development of STRAPTk (www.dcs.shef.ac.uk/spandh/projects/straptk-Speech Training Application Toolkit) from the point of view of developers, clinicians, and clients in the domain of speech training for severely dysarthric speakers. Preliminary results from an extended field trial are presented.

1. Introduction

Various computer based speech training systems with applications to speech therapy and second language learning [6], [13], [14], [18], have attempted to improve *intelligibility*. In contrast this study has the aim of improving speech *consistency* so that severely dysarthric clients can use automatic speech recognition to control assistive technology. This work was commissioned within the STARDUST project (Speech Training and Recognition for Dysarthric Users of Speech Technology). Software development and speech technology deployment in this area should adopt a holistic approach, taking into account the requirements of the trainee, the instructor and the tool designer. The designer, in the attempt to meet a range of specifications and requirements from various instructors and trainees, requires a development tool that supports rapid prototyping and usability testing. This paper discusses at a general level the software requirements of such a speech training toolkit, then we describe the specific components and structure of STRAPTk; finally we report encouraging initial results from field testing for severely dysarthric speakers.

2. General Speech Application Requirements

There are numerous speech technology toolkits, specialising either in speech processing (SFS, [12], Snack, [17]), speech recognition (ISIP, [7], HTK, [19]), dialogue, (CSLU, [15]), annotation (EMU, [2], AGTK, [1]), speech analysis-visualisation (SAPPHIRE, [11], Wavesurfer, [16]). STRAPTk shares several common characteristics with these other toolkits and yet it is different in many aspects because of the specialisation in the domain of speech training. In the following sections we discuss the requirements for the software development of speech technology toolkits

2.1. System Requirements

Coding: Two levels of programming are required, a low level, usually C/C++, to create components that require computationally expensive processing, reuse of code, data structure flexibility , and a scripting language, in this case Tcl/Tk, for integrating and extending components, managing interaction, rapid prototyping, and flexible development of graphical user interfaces.
Interoperability: The components must function both independently and in combination with other components.
Open architecture: allowing developers to extend the toolkit's functionality and facilitate research
Portability/Compatibility: The toolkit application must be easy to install on different platforms and demonstrate backward compatibility.
Configurability: All the components have to be customizable, allowing the developer to access any functional aspect of them to be further developed at the end user level.
Graphical User Interface (GUI): GUIs should exhibit such usability properties as attractiveness, comprehensibility, adaptability, and intuitiveness.

2.2. Speech Training Requirements

Speech training tools: Tools should be provided for speech training at all different phonetic levels (sound, segment, word, and sentence), and for various speech features (voicing, pitch, nasality, etc…). The tools must allow also the instructor to set exercises for the client to complement his/her normal training schedule.
Database Management System (DBMS): The toolkit should allow systematic collection and warehousing of data as well as storage of user-customized configurations.
Graphical User Interfaces: We consider the following graphical tools to be essential for speech training:
DBMS GUI for accessing the database, retrieving training exercises, examining results, etc.
Recorder GUI for recording speech data for various tasks either for diagnostic, assessment, or client training purposes.
Transcription GUI for viewing, editing, and automatic or manual labelling of the utterances recorded.
Recording Browser GUI for fast access to the recorded utterances of a client and for creating collections of selected samples.
Recognizer GUI for recognizer training, monitoring and predicting the performance of a customised recognizer.

This will be mainly used for providing feedback at a word/sentence level.
Recognition GUI for providing recognition-based feedback at a word/sentence level.
Visual maps GUI for client training, and obtaining feedback at the sound/segment level, [10].
Progress GUI for monitoring the progress of a client based on a specific training feature and a metric of the training tool.

2.3. Commercial speech training software

Considering these requirements, commercial systems (such as IBM Speech Viewer, [14], Accent Coach, [18], ISTRA, [13], and Video Voice, [6]) which are not based on an open architecture cannot be extended or configured easily. They usually do not readily integrate with other speech technology toolkits and their database structures usually store only information about the history record and the progress of clients. There are also significant limitations concerning the kind of visual feedback they provide: see [9] for an extensive analysis. Their limitations notwithstanding, it must be acknowledged that the abovementioned applications have proved beneficial in speech therapy and thus encouraged further research and development.

3. STRAPTk

3.1. Software architecture

Unlike most applications for the Windows environment, STRAPTk decouples functionality from the graphical user interfaces: the developer can manipulate the operation of the application during run-time using a console (command-line environment), whereas instructors and clients may equivalently work at a higher level (graphics environment). The following subsections discuss these functions and the application's structure in more detail.

3.1.1. The console

We have coded classes to encapsulate the functionality of the main entities found in the speech training domain such as client, exercise-task, stimuli, recording session, utterances, utterances collections, and recognition engines as well as their relationships. All the GUIs we have developed are linked to those classes, providing effective communication and operation based on the common entities they share. This allows the appearance and operation of GUIs to be modified manually or automatically to suit the circumstances. Any instance of a class can be accessed at the console level, providing a powerful tool for the developer, enabling rapid programming of prototypes and the ability to check the internal running state of the application.

3.1.2. The Database and the Resources Manager GUI

The current database structure stores permanently the public properties of any object created with an interface for reading, writing, and querying operations, together with code for configurations of recognition engines and GUIs. This feature of STRAPTk allows the dynamic reconfiguration of any available tool depending on the task and the required customisation for a specific client. The non-expert user may manipulate the database with the assistance of a resources manager GUI.

3.1.3. The Chameleon Recorder

As its name indicates, this is the primary GUI that is used as a tool for the completion of several tasks. Thanks to its multi-component structure and close integration with the database, the user may select on-the-fly among predefined configurations that determine the overall appearance and operation of the recorder, and the more specific configuration or properties of its components. Currently, the Recorder serves as the main interface for the following functions: speech acquisition and corpus creation (§3.1.4), transcription, speech visualisation, speech recognition, speech consistency training (§3.1.6), and the interface to assistive technology (§3.1.7). For instance, the configuration shown in **[Fig. II]** is set up for speech consistency training: the user speaks words in response to prompts and receives visual feedback based on the forced-alignment Viterbi match to HMMs trained for these words.

3.1.4. The Recording Browser

In speech training applications it is important for the instructor to be able to rapidly review recordings of speech material made by a client, for instance to play back and select those utterances to be used in recognizer training sets. The Recording Browser **[Fig. I]** provides this: typically a client records a 'session' consisting of 10 repetitions of 10 words. The recording browser presents a corresponding 2D grid, each cell corresponding to 1 utterance in the session. Clicking on the cell plays back the word and associates that utterance with the other tools (recognizer, chameleon recorder) for further processing and visualisation of the result (speech recognition, spectrograms, waveforms, etc). Additionally the browser can be used to select utterances to create *collections* for various purposes, e.g. training a recognizer (§3.1.5).

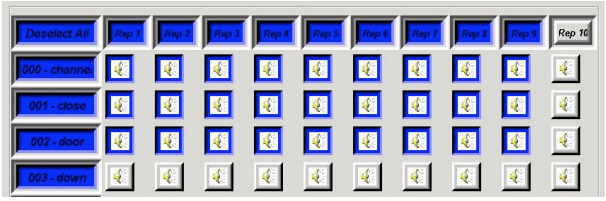

Figure I: The recording browser showing a 4 words x 10 repetitions recording session

3.1.5. The Recognizer and its GUIs

STRAPTk provides a wrapper class for the HTK toolkit [19] and GUIs for configuration of speech recognition parameters and performance appraisal. This enables the non-expert user, to create and configure a fully functional speaker-dependent recognition engine with minimal effort. A selection of baseline recognizer configurations (specifying signal processing specification, number of HMM states, number of mixtures, training/testing/evaluation collections §3.1.4…) are stored in the database. The user can select one of these and adapt it as necessary, or alternatively define a configuration from scratch. The aim is to allow a non-expert user to train a recognizer using a standard configuration and at the same time provide an expert user with the ability to experiment with the recognizer set-up. This dual purpose is typical of

the software engineering challenges addressed in STRAPTk. The performance of any recognizer that is fully built may be examined with the decoding of any collection (training/testing/evaluation) selected from the database or with a single utterance. STRAPTk provides a visualisation of the recognizer's performance on a collection of utterances with an html-formatted report that presents confusion and confusability matrices and the likelihood scores, the subject of a companion paper [8]. Based on this visualisation, speech samples which deviate significantly from the model may be removed from the training set so that the recognizer can be subsequently retrained for more accuracy, [8]. These tools are necessary when there is insufficient data available to assess a recognizer on an independent test set in the conventional way.

3.1.6. Speech Consistency Training with the Recognizer

The operation of any recognizer is accessed through the chameleon recorder and visual feedback from the recognition of an utterance (either live or pre-recorded) is presented to the client graphically **[Fig. II]**. The right panel compares the likelihood of any of the recognizer's models of generating the utterance just spoken. The objective here is to provide immediate word error rate feedback and assist in identifying any confusability patterns, [8]. The left panel is used during training to display visual feedback to the client. The bar on the left corresponds to the forced-alignment likelihood of the utterance in the training set that best-matches the model for the target word: it typifies what the recognizer has been trained to accept. Similarly, the bar on the right represents the forced-alignment score of the current attempt of the client. The client's task is to modify his/her production of the prompted word to minimise the difference in height between the two bars. In addition to this visual feedback, the client may also receive audio feedback by playing back the maximum likelihood utterance from the training set ('the recognizer wants you to say it like this…'). The success of this audio-visual feedback in demonstrating the proximity of the spoken utterance to the best fit model is explored in the following sections.

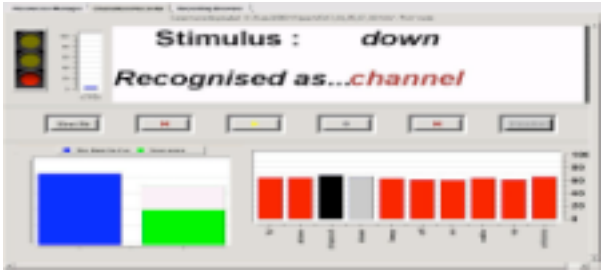

Figure II: Utterance recognition with visual feedback

3.1.7. Extensions for assistive technology

The overall goal of the STARDUST project is to investigate the possibility of providing a speech-driven interface to assistive technology. This technology enables people with severe physical disabilities to use a variety of methods to gain access to devices for communication, mobility and environmental control tasks, e.g., turning on the television or hi-fi, illuminating a room, etc. For people with physical disabilities, a speech interface to such technology offers an attractive alternative to other control methods. For people with neurological conditions such as cerebral palsy, multiple sclerosis or head-injury, their physical disabilities are often accompanied with some limitations of speech production, a condition known as dysarthria. Dysarthria is the most common acquired speech disorder, [4]. In its severest form, dysarthric speech can be unintelligible to others. It has been shown that commercially available ASR systems conventionally trained perform poorly, [5]. We mentioned (§3.1.3) that the chameleon recorder can be configured in such a way as to provide a speech-driven interface to assistive technology for dysarthric users. For that purpose, the recorder GUI is linked to a second program developed in the project, called ECTalk. This is the program that interfaces with external home devices and allows the control of the recorder using a switch. ECTalk also incorporates a communication aid that, in the event of unsuccessful recognition provides the user with a more conventional means of accessing their assistive technology via a switch-controlled scanning-interface.

4. Speech training field trial

STARDUST supports a longitudinal study with 8 severely dysarthric clients. Each client undertakes a period of speech training using a speaker-dependent recognizer trained from her/his initial recordings. This period of client's training is motivated by the observation that dysarthric speakers' productions can exhibit a far greater degree of variability than those of normal speakers, [8]. We intend to investigate the effect of training on the consistency of a client's productions. If a client can produce a target command word with a close phonetic proximity to the recognizer target model with a high degree of frequency, then it should be possible to train a speaker-dependent recognizer with a high level of recognition accuracy.

4.1.1. The client's training scheme

Currently, there are 8 subjects participating in a trial of the speech training system. All subjects have been assessed as severe dysarthric – less than 30% intelligible as rated by naïve listeners on the Frenchay Dysarthria Assessment, [3]. Prior to the client's training phase, a speech and language therapist uses the STRAPTk recorder to make several recordings of the command vocabulary, typically consisting of around 10 words. These recordings are then used to train a recognizer. Recognizer performance is encouraging [8]. A major innovation in the STARDUST project is that the speech training can be conducted in the client's own home, without requiring the presence of a therapist for every session. In each practice session the client produces each word in the recognizer's vocabulary for a specified number of repetitions, usually 4 or 5, as determined by the therapist. During the practice session the client exercises complete control over the program via the manipulation of a single switch. This switch mechanism enables the client not only to operate the recognizer and iterate through the stimuli set, but also to listen to the best fit utterance. The client is given feedback as to whether the word is correctly recognized or not. If it is recognized, a bar-graph is presented showing the user how close the pronunciation attempt was to the target **[Fig. II]**. The user then has the option of making another attempt – possibly after listening – to produce an utterance closer to the maximum likelihood target.

4.1.2. Results

We have preliminary results from five users. Two users did not achieve greater recognition probabilities for their productions over three weeks of practice. The remaining three users showed an increase in the mean recognition probabilities for most words. Further analysis is required to investigate whether these higher scores are actually indicative of a greater level of consistency in their productions. For one client, the general trend of recorded recognition probabilities continued to increase in a period of three weeks of training, (right panel of **[Fig. III]**). For another client, recognition probabilities increased rapidly over the early sessions, and then began to plateau in subsequent sessions, (left panel of **[Fig. III]**). For the users who achieved higher recognition probabilities with most of the words in the vocabulary, this pattern was not replicated with one or two words. Further investigation and analysis of these results might explain why the apparent improvement in the production of some words was not replicated in others.

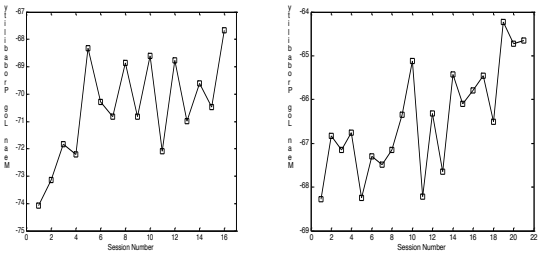

Figure III: The trend of mean log-probability recognition scores over several training sessions for two clients.

5. Conclusions

It has become apparent from this discussion that research and development in computer-based speech training systems require a flexible environment where ideas can be tested with a minimum effort and prototypes can be built by non-computer experts with a modular approach. STRAPTk has been developed by a multi-disciplinary team with experience in the domain of speech training and follows a careful study of the state of the art in the software design of similar systems. The application of this toolkit in the domain of speech recognition for severely dysarthric speakers may signal the start of a prosperous continuation in other problem domains and by various other experts. We have expanded the design of this software in the OLP (Ortho-Logo-Paedia) 5th European framework project (www.xanthi.ilsp.gr/olp/).

6. Acknowledgements

STARDUST is funded by the UK Department of Health NEAT programme.

7. References

[1] Bird S., Maeda K., Ma X., Lee H., Randall B., and Zayat S. (2002). *"Tabletrans, multitrans, intertrans and tree-trans: Diverse tools built on the annotation graph toolkit"*. In Proceedings of the Third International Conference on Language Resources and Evaluation.

[2] Cassidy S., Harrington J., "*Multi-level Annotation of Speech: An Overview of the EMU Speech Database Management System"*. Speech Communication, 2000.

[3] Enderby P.M., "*Frenchay Dysarthria Assessment"*, Pro-Ed, Texas, 1983.

[4] Enderby, P.M. & Emerson, J., *Does speech and language therapy work?* Whurr: London, 1995.

[5] Ferrier, L.J., Shane, H.C. Ballard, H.F., Carpenter, T., Benoit, A., Dysarthric speech: A comparison of computerized speech recognition and listener intelligibility, *Journal of Augmentative and Alternative Communictation*, 11, 165-174.

[6] Fitzgerald M., Gruenwald A., Stoker R., (1989), *"Software review – Video Voice Speech Training System"*, Volta Review, vol. 89, pp. 171-173

[7] Ganapathiraju A., Deshmukh N., Hamaker J., Mantha V., Wu Y., Zhang X., Zhao J., and Picone J., (1999), *"ISIP Public Domain LVCSR System,"* Proceedings of the Speech Transcription Workshop.

[8] Green P., Carmichael J., Hatzis A., Enderby P., Hawley M., (2003),*"Automatic speech recognition with sparse training data for dysarthric speakers"*, In Proc: (Eurospeech'03), September 2003.

[9] Hatzis A., (1999), *"Optical Logo-Therapy (OLT) Computer-Based Audio-Visual Feedback Using Interactive Displays for Speech Training"*. Unpublished PhD thesis, Dept. of Computer Science, Sheffield University.

[10] Hatzis A., P.D. Green, (2001), *"A two dimensional kinematic mapping between speech and acoustics and vocal tract configurations"*, In Proc: Workshop on Innovation in Speech Processing, (WISP'01).

[11] Hetherington, L., McCandless, M., 1996. *"SAPPHIRE: an extensible speech analysis and recognition tool based on Tcl/Tk"*. In: Proc. (ICSLP'96), pp. 1942-1945.

[12] Huckvale, M., 1987-1998. *"SFS Speech Filing System"*. [http://www.phon.ucl.ac.uk/resource/sfs.html].

[13] Kewley - Port D., Watson C.S., and Cromer P.A. (1987), *"The Indiana Speech Training Aid ISTRA: A microcomputer-based aid using speaker-dependent speech recognition"*. Synergy '87, The 1987 ASHF Computer Conference, Proceedings, pp. 94 - 99.

[14] Pratt S., Heintzelman A.T., and Deming S. Ensrud (1993), *"The efficacy of using the IBM Speech Viewer vowel accuracy module to treat young children with hearing impairment"*, Journal of Speech and Hearing Research, vol. 36, pp. 1063-1074.

[15] Schalkwyk, J., de Villiers, J., van Vuuren, S., Vermeulen, P., 1997. *"CSLUsh: an extensible research environment"*. In: Proc. (Eurospeech 97), pp. 689-692.

[16] Sjölander K., and Beskow J., *"Wavesurfer – an open source speech tool"*, In Proc. ICSLP 2000.

[17] Sjölander, K., Beskow, J., Gustafson, J., Lewin, E., Carlson, R., Granström, B., (1998), "*Web-based Educational Tools for Speech Technology"*. In: Proc. (ICSLP'98), pp. 3217-3220.

[18] Taylor R.P, (1999), *"Accent Coach English Pronunciation Trainer"*, Software Review for the Computer Assisted Language Instruction Consortium.

[19] Young S.J., Woodland P.C., Byrne W.J. (2002), *"HTK: Hidden Markov Model Toolkit V3.2"*, Cambridge University Engineering Department, Speech Group and Entropic Research Laboratories Inc.