Building an interface between EMU and Praat: a modular approach to speech database analysis

Jonathan Harrington [†], Steve Cassidy [‡], Tina John[†] and Michel Scheffers [†]

[†] Institute of Phonetics and Digital Speech Processing, University of Kiel, Germany

[‡] Centre for Language Technology, Macquarie University, Sydney, Australia

ABSTRACT

In this paper, we demonstrate the advantages of combining the largely complementary systems of Praat, a computational system for doing phonetics, with the EMU system for speech database analysis. The interface applies to the annotations in which a Praat TextGrid is converted into an EMU hierarchical annotation structure and vice-versa. With the exception of annotations in EMU that are not explicitly linked to times, we show that there is no loss of information in this conversion. The interface between the Praat and EMU systems provides a flexible labelling system: the data can be labelled as segments or events in Praat and various kinds of structures between annotation tiers can be defined and then queried within EMU. We argue that both the variety of existing speech databases as well as the multitude of different possible types of speech analysis require a modular approach allowing the integration of a number of different stand-alone components that are adapted to different aspects of creating, annotation, querying and analysing speech data.

1. INTRODUCTION

In recent years, there has been a considerable growth in the usage of corpora in both speech and language research and recent surveys show that there are several hundreds of corpora available worldwide [1,2]. The growth in speech corpora was initially largely driven by the need to develop training material for speech technology systems, but with the improvements in the technology for storing, accessing and exchanging large corpora, many speech databases have been created for various aspects of basic research such as prosody and intonation [3], coarticulation and speech production [4] and endangered languages [5]. Although large speech databases are usually established with the aim of being shared across the research community, they are sometimes created with proprietory or platform specific software (e.g. the earlier versions of TOBI which required Entropic's Waves system running on SUN hardware) or special purpose file formats (e.g., [4]).

A speech corpus is usually not just a collection of signal files, but it is also likely to contain several different types of *annotation*. Whereas a good deal of effort has been devoted to creating speech tools for the manipulation of sound files and for deriving signal files in the time and frequency domains, the problem of how to represent annotations has only been addressed recently [6]. There are therefore few annotation systems used in speech research that allow hierarchical relationships between annotation levels to be represented, or that distinguish between hierarchical and autosegmental tiers, or that can represent intersecting sets of hierarchically structured labels from two or more persons participating in a dialogue, or that allow annotation structures of this kind to be semi-automatically constructed – and yet all these kinds of representations are fundamental to very many types of enquiries in phonetic and phonological theory. Recently, some progress towards the issue of how to represent annotated speech corpora has been made in the annotation graph system [7] implemented within the ATLAS project, the XML-based Mate annotation workbench [8], the heterogeneous relation graph of the Festival system [9] and the EMU system which is discussed in further detail below.

Once a database has been annotated, there has to be a mechanism for *querying* the annotations and their associated signal files. Such queries need to take account of the fact that the symbolic structure of speech is inherently hierarchical with perhaps multiple structures linked to the same or different time signals (some of which are discussed below). The development of query languages must be closely related to types of annotation that are declared to be computationally feasible. There are very few query languages that are adapted to speech analysis. With the exception of the EMU system, a query language Q4M based on an XML representation has been developed within the Mate workbench [8] and an annotation graph query language is under development [10]. Finally, the extracted data might be *analysed* in an environment for its graphical and numerical analysis. This functionality might be provided either as an inherent (as in Praat) or independent (as in EMU) part of the system in which the speech data or corpus was created.

In this paper, we discuss the adequacy of the Praat and EMU systems for creating, annotating, querying and analyzing speech data. We then present some tools that provide an interface between the annotation structures of the systems. In the final section, we argue for a modular approach to speech tool development in which the user can piece together platform-independent tools that are designed for specific tasks in speech and language research.

2. PRAAT AND EMU

The Praat system is a tool for 'doing phonetics by computer' [11]. Praat accepts a wide variety of sound formats, it is platform-independent and it has extensive facilities for displaying, segmenting and annotating

ISBN 1-876346-48-5 © 2003 UAB

speech signals. It includes numerous routines for digital signal processing, as well as pitch and formant trackers and a facility for resynthesising speech. There are also programs that are specifically designed for phonetic and phonological analyses such as vocal tract modelling, articulatory synthesis and analysing data within optimality theory. Praat provides a GUI for the display of speech data through which all signal processing operations can be applied. In addition to this, Praat includes its own scripting language for signal processing, acoustic phonetic analysis and other types of applications.

A speech signal can be annotated in Praat using a number of different tiers either as segments (intervals) or as events. There is a query system for extracting parameters from signal files (such as mean F0). The system has justifiably become very popular both as a research and teaching tool.

The EMU speech database system has its origins in the development by Gordon Watson and Jonathan Harrington in the late 1980s of the 'Acoustic Phonetics in S' system at CSTR, Edinburgh University and has undergone numerous extensions and transformations since that time at SHLRC, Macquarie University [12-15]. The Emu speech database system is an integrated set of tools for creating, querying and analysing annotated speech corpora. The core of Emu is implemented as a C++ library and a set of extensions to the Tcl scripting language. This core is augmented with other components which deal with sound file input/output and signal processing and analysis to form an integrated toolkit for corpus based speech research.

The EMU approach can be characterised as leveraging existing general purpose tools where appropriate for speech data analysis. Hence it uses the Splus [16] statistical language (or more recently the open source R implementation [17]) for numerical analysis of speech data; it uses Tcl [18] as the scripting language with which to automate corpus generation and analysis tasks; third part signal processing tools such as ESPS, the Edinburgh Speech Tools [19], Snack [20] and the Kiel-Xassp system [21] can be used for speech signal processing. Hence, while signal processing is limited in EMU, its architecture makes it easy to interface to other tools which perform more extensive DSP operations.

A key difference between EMU and Praat (and indeed most other speech signal processing systems) is that EMU is designed for building speech databases whose characteristics have to be pre-defined by the user in a *template file*. The template provides information about the directory location of all files, the types of signal files (their format, their file-extension name) and details of how the signal files are to be annotated. The template also specifies the number and types of annotations that are to be included, their hierarchical relationships, and how they are related to time signals. One of the practical advantages of defining a template file is that it provides an efficient way for multiple users possibly working on different platforms at different sites to use the same sets of operations and commands on a database. It also allows a user to specify the kinds of display that are likely to be needed if the EMU graphical tools are used. The theoretical motivation for the template file is that it encourages database creators to plan the structure of a database carefully in terms of the types of queries and analyses that are likely to be required of it.

EMU allows annotations to be specified as intersecting hierarchical systems as shown in Fig. 1. A user may construct some of the hierarchical annotations semi-automatically (e.g., words become accented if one of their syllables is associated to a pitch-accent; phonemes are syllabified according to the maximum onset principle, etc.) using the Tcl scripting language or in C++ using the EMU core library. EMU outputs label files in a format compatible with Waves and can read annotation files from Waves, Transcriber [22], ACCOR [4] and the Kiel-Xassp system [21]. Support for new file formats can be added to the core relatively easily.

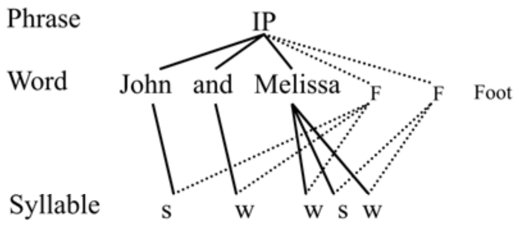

Fig. 1. An example of intersecting hierarchies. An intonational phrase is made up of words and in a separate plane of stress-feet (defined here as a strong syllable followed by any number of weak syllables, irrespective of word boundaries). The Word and Stress-Foot tiers are then linked at the Syllable tier.

EMU provides an extensive query language which can be used to locate annotations together with their associated times within a corpus. Queries can be based on sequential structure (e.g. ‘find all /p/ phonemes between vowels’), hierarchical structure (find all H* tone targets in phrase-final words) or a combination of these (e.g., ‘find the word following an accented word that dominates a H* tone target’). The query language also allows number (find all /p/ phonemes in 3 syllable words) and position (find all word-initial syllables) to be specified.

Another central difference compared with Praat is that the extracted data is exported into a separate system for graphical and numerical analysis. Since its initial development as APS at CSTR Edinburgh, the preferred environment for the analysis has been the S (Splus) programming language and more recently its open-source derivative R [17], which runs on all major platforms. The EMU system includes a set of extensions to this environment specific to speech analysis and its visualisation. Almost all of these are centered around a *segment list*, i.e. a collection of segments defined by start and end times that are extracted from the database using the query language. Signal file data is then typically extracted with reference to segment list boundaries. This approach provides an extensive library of third party statistical and numerical methods [17] to the speech researcher as well as allowing a user to write scripts in a widely known programming language.

3. THE PRAAT-EMU-PRAAT INTERFACES

To take advantage of the different strengths of the two systems, an interface has been developed for transfering annotations between Praat and EMU. A user may segment and label data in EMU and then export the annotations into Praat and vice-versa. Currently, derived files such as formant or pitch data can't be transferred between the two systems. The interface consists of Tcl-scripts that are used to convert between a Praat TextGrid (Fig. 2) and a set of hierarchically structured EMU annotations (Fig. 3).

The conversion script is executed using a simple graphical interface in which the user selects the annotations that are to be converted as well as the directory location to which the output files should be saved. For converting annotations from EMU to Praat, the user must also provide the name of an EMU template file. The scripts can also convert an entire directory of annotation files.

In Praat, the notion of hierarchy as inclusion can only be implicitly expressed through the relationship of annotations to times. For example, in Fig. 2, we may say that 'noch' ('still') dominates the three segments [n O x] because the start and end times of 'noch' are the same as first and last segments of [n O x]. Hierarchies can be explicitly built from these types of time-dependent relationships in EMU, but there is no requirement that this should be so.

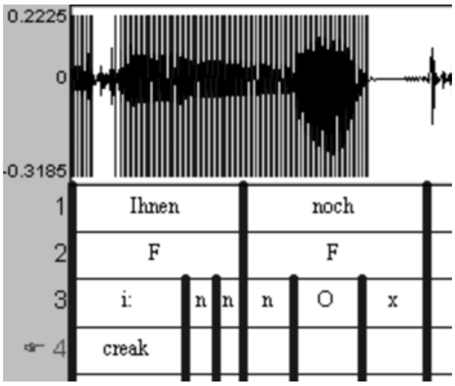

Fig.2. A Praat TextGrid of a fragment of the Kiel-Corpus of spoken recordings in German [21].

For example, in the tree-structure in Fig. 3, which is taken from the Kiel-Corpus of Spoken German [21], the links between the 'Kanonic' (citation-form) and 'Phonetic' levels are used to express the relationship between citation-form, dictionary-generated pronunciations and their phonetic realizations in continuous speech. In this particular fragment of the tree, the deletion of a citation-form glottal stop, which typically occurs before citation-form vowel-initial morphemes in German, is represented by its presence at the Kanonic level and its absence at the Phonetic levels. This is therefore an example of a word 'Ihnen' ('to them') that dominates a segment (a glottal stop) without that segment being linked to any time signal. The advantage of such an analysis is that it leaves open the possibility of subsequently querying and analysing (perhaps a large number of) word-initial vowels, in order to determine whether the glottal stop has left an acoustic signature which is not immediately apparent at the labelling stage. 'Timeless' labels can also be useful in rapidly annotating a corpus in which it is important to be able to query according to the context but without the context necessarily being linked to times. In a TOBI-style annotation for example, it might be sufficient to associate pitch-accent F0-targets with accented words, but only mark the start and end times of the words at the beginning and end respectively of each phrase.

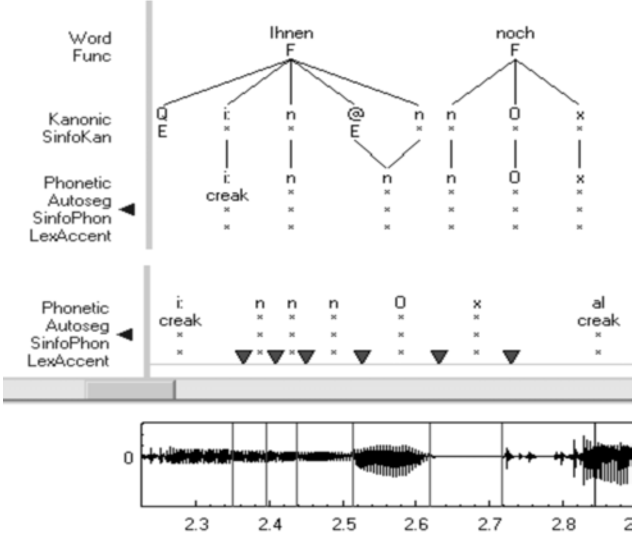

Fig. 3. The same fragment of the Kiel Corpus as in Fig. 2 represented hierarchically in EMU. The /Q/ label (glottal stop) at the Kanonic level is not associated with time to express segment deletion.

In the Waves-Entropic system by contrast, which is still widely used for TOBI-markup, all the word boundaries have to be explicitly associated with times, even though word boundary information and word duration may be rarely used in the subsequent prosodic analysis.

In the Praat-to-EMU conversion, the separate tiers of the Praat TextGrid are converted into an equivalent set of time-linked EMU-tiers: a hierarchical structure can then be superimposed on top of these using the Tcl-scripting language, together with some EMU-Tcl tree-building scripts. No information is lost in this conversion. On the other hand, whereas all time-linked EMU-annotations are also faithfully converted, timeless annotations are not preserved in a Praat TextGrid.

4. CONCLUSIONS

The Praat segmentation, labelling and in particular the signal manipulation and resynthesis facilities are unquestionably superior to any corresponding functionality that currently exists, or that we are likely to want to develop ourselves, in EMU. On the other hand, EMU allows a considerably richer annotation structure to

ISBN 1-876346-48-5 © 2003 UAB

be represented, together with rules for its construction as well as an extensive query language for the extraction of annotations. We are not proposing any new speech tools here, but instead an interface which provides the user with a set of platform-independent modules that extend over a very broad range of the tasks involved in creating, annotating, querying and analysing speech corpora. At least as far as the acoustic analysis of speech is concerned, we believe that there is currently at least as much payoff from building interfaces between complementary systems such as these as developing new tools for speech analysis.

More generally, a modular approach seems to us to be an essential part of speech tool development: the highly multidisciplinary nature of phonetics and speech analysis would seem to preclude designing a single all-encompassing system which is adapted to the huge range of experimental paradigms that are found in speech research. This is one of the main reasons why, in contrast to Praat, we prefer to make use of independently supported programming environments that can be adapted to speech analysis such as Splus/R and Tcl, rather than designing an EMU-specific scripting language. Such an approach has the immediate practical advantage that we can borrow numerous functions that are useful for speech analysis, such as polynomial fitting, principal components analysis and other basic functions for graphical and statistical analysis from other packages. A longer-term advantage of this modular approach is that if EMU were no longer maintained, it might be replaced with some version of the Bird & Liberman annotation graph toolkit [7] combined perhaps with the MATE query language [8], but importantly without disrupting irrevocably the other speech analysis tasks that are devolved to other modules. If Praat development were to cease, a user stands to lose not only a very versatile system for speech segmentation, labelling, processing and synthesis, but also the considerable investment in writing Praat-scripts for speech analysis that might not easily carry over into other environments.

REFERENCES

[1] S. Bird and M. Liberman, "Linguistic annotation resources,". www.ldc.upenn.edu/annotation

[2] ELRA, www.icp.grenet.fr/ELRA.

[3] M.E. Beckman and G.M Ayers. "Guidelines for TOBI labelling'" http://www.ling.ohio-state.edu/~tobi/.

[4] EUR-ACCOR: www.cstr.ed.ac.uk/artic/accor.html

[5] S. Bird, "Multidimensional exploration of online linguistic field data," *Proceedings of the 29th Annual Meeting of the Northeast Linguistics Society*, pp. 33–47, 1999.

[6] S. Bird and J. Harrington, Eds. "Special issue on speech annotation and corpus tools," *Speech Communication*, Vol. 33, pp. 1-174, 2001.

[7] S. Bird and M. Liberman, "A formal framework for linguistic annotation," *Speech Communication*, Vol. 33, pp. 23-60, 2001.

[8] D. McKelvie et al., "The MATE workbench – an annotation tool for XML coded speech corpora," *Speech Communication*, Vol. 33, pp. 97-112, 2001.

[9] P. Taylor, A. Black and R. Caley, "Heterogeneous relation graphs as a formalism for representing linguistic annotations," *Speech Communication*, Vol. 33, pp. 153-174, 2001.

[10] S. Cassidy and S. Bird, "Querying databases of annotated speech," *Proc. 11th Australasian Database Conference*, Vol. 22, pp. 12-20, 2000.

[11] P. Boersma and D. Weenik, "Praat, a system for doing phonetics by computer", Tech. Report 132, Inst. Phonetic Sciences, Univ. Amsterdam. www.praat.org, 2001.

[12] J. Harrington et. al., "The mu+ system for corpus-based speech research," *Computer Speech and Language*, Vol. 7, pp. 305-331, 1993.

[13] S. Cassidy, "Compiling multi-tiered speech databases into the relational model: experments with the EMU system," *Proc. Eurospeech*, pp. 2238-2242, 1999.

[14] S. Cassidy et al., "Testing the adequacy of query languages against annotated spoken dialog," *Proc. Australian Speech Science and Tech. Conference*, pp. 428-433.

[15] S. Cassidy and J. Harrington, "Multi-level annotation in the Emu speech database management system", *Speech Communication*, Vol. 33, pp. 61-77, 2001. http://emu.sourceforge.net/

[16] W. Venables and B. Ripley, S-Progamming, London: Springer-Verlag, 2000.

[17] The R archive network. http://cran.r-project.org/

[18] J. Ousterhout, *Tcl and the Tk Toolkit*, Wokingham: Addison-Wesley, 1994. http://www.tcl.tk/

[19] P. Taylor et al. "Edinburgh speech tools library", www.cstr.ed.ac.uk/projects/speech_tools, 1998.

[20] K. Sjölander, "The Snack sound extension for Tcl/Tk", http://www.speech.kth.se/snack/, 2000.

[21] A. Simpson et al. "The Kiel Corpus of Read/Spontaneous speech. AIPUK, Vol. 32, pp. 31-115. www.ipds.uni-kiel.de/forschung/xassp.en.html, 1997.

[22] C. Barras et al. "Transcriber: development and use of a tool for assisting speech corpora production," *Speech Communication*, Vol. 33, pp. 5-22, 2001.