EMU: an Enhanced Hierarchical Speech Data Management System

Steve Cassidy and Jonathan Harrington
Speech Hearing and Language Research Centre, Macquarie University, Sydney

ABSTRACT: EMU is a system for labelling, managing and retrieving data from speech databases such as the Australian ANDOSL database or the US TIMIT.

EMU is a re-implementation of the earlier MU+ system (Harrington, Cassidy, Fletcher, and McVeigh 1993) with the aim of providing a more flexible environment. The hierarchical structures and database query facility have been generalised and the system has been extended to include an interactive labeller with spectrogram and waveform displays. EMU incorporates the Tcl/Tk scripting language which can be used to extend the labeller and to perform many automated operations on databases; as an example, scripts have been written to automatically construct hierarchical descriptions given Phonetic level labels.

The need for increased flexibility was driven largely by the desire to use the system on languages other than English. This paper concludes by describing a database for Cantonese, and a database used in a kinematic study of vowel lengthening, both of which include facilities for automatically generating hierarchies.

IMPROVING ON MU+

The MU+ speech database system (Harrington, Cassidy, Fletcher, and McVeigh 1993) was developed at Macquarie University over a number of years to serve the research needs of speech scientists working with large collections of speech data. It provided the ability to associate hierarchical labels with speech data files and allowed retrieval of speech segments based on sequential and hierarchical criteria. Although MU+ provided a powerful set of tools, the evolutionary manner in which it was developed meant that much of the underlying code was specific to the databases it had been used with and was difficult to adapt to new projects. It was decided to re-design and re-implement the core functionality of MU+ to provide a flexible platform for future development; the new system, called EMU, generalises the functionality of MU+ and provides a new set of tools for labelling and querying speech databases.

The EMU system manages speech data with associated label files which record either the start and end times of segments (for example phonemes or words) or the times of events (for example vowel targets or tongue displacement maxima). These label files, which describe the sequential structure of an utterance, can be integrated into and augmented by an hierarchical description which, for example, groups phonemes into words and words into phrases. EMU then allows queries of the database which utilise both the sequential labels (find all vowels following stop consonants) and the hierarchical description (find all stop consonants in word initial position).

Compared with its predecessor, EMU allows a richer variety of structure to be imposed on an utterance and allows the simple integration of multiple label files. The query facility has also been overhauled to simplify the most common queries while allowing complex queries to be expressed in a reasonably transparent syntax. Whereas the MU+ system was bound up with the Splus statistical package, EMU exists as a set of stand-alone tools and so can be used to extract data for import into a variety of analysis packages.

The remainder of this paper discusses the main features of EMU and concludes with examples of two databases built using the new software.

DATABASE STRUCTURE

An EMU database is a collection of speech data and label files corresponding to a number of utterances. The structure of the database, in terms of the files that it contains and the structure of the labels for each utterance, are defined in a single database template file. This file defines the following:

- The different kinds (levels) of segments/events which are labelled in the database (for example, phoneme, word and syllable segments) and the relationship between these levels.
- The different kinds of label which can be placed at each level (for example, words might be labelled with their text and their part of speech)
- Information about each set of external label files, these files contain segment/event times and labels created, for example with Waves+ or the EMU labeller.
- A search path for the different kinds of file making up the database. This allows, for example, speech data files to reside on CD-ROM while label files reside on hard disk.
- The file extensions associated with different speech data tracks (for example samples, formants).
- Finally, arbitrary information can be recorded for use with user scripts.

EXTENSIBILITY

EMU is implemented as a C++ library which is used to build the various tools which make up the system. A major lesson from the MU+ project was that it is impossible to foresee the detailed requirements of future speech database projects. Building new database or analysis tools would normally require writing new C or C++ code which used the core library; as an alternative, the EMU system incorporates the Tcl/Tk scripting language (Ousterhout 1994) which enables new extensions to be built quickly and easily. An advantage of this approach is that graphical user interfaces to various database functions can be built using the Tk windowing toolkit which is part of Tcl/Tk. Since Tcl/Tk is portable across all major platforms, the user interfaces built with the toolkit are also portable. All of the EMU user interface tools including the labeller are built using Tcl/Tk and their components, including spectrogram and waveform display, are available for use in new applications.

An important use of the Tcl/Tk scripting language is to automatically build hierarchical descriptions for utterances which have been hand-labelled at one or more levels. For example, a procedure can be written to group phoneme segments into syllables or to locate word boundaries in a phonetic level transcription. Modules have been written to allow simple rules and look-up tables to be used for the most common level-to-level transformations, for example, building a phonemic level given a set of phonetic level labels. An example script taken from a Cantonese database application is shown in Figure 1, this script creates a new segment at the Syllable level which dominates all segments at the Mu level and shows how a script can query the current state of the hierarchy and add new structure to it.

THE EMU LABELLER

The fundamental principle of the EMU system is that speech data has both sequential and hierarchical structure. The normal mode of labelling of speech data only deals with the sequential structure: for example, the sequence of Phonetic elements or the position of prosodic events for an utterance. In the older MU+ system, hierarchical structure was built after sequential labelling had been completed. The EMU system integrates both kinds of labelling in the EMU labeller which offers most of the facilities of a traditional sequential labeller (spectrogram and waveform displays etc.) and at the same time allows the user to construct a hierarchical labelling either manually or automatically.

The advantage of integrating these two kinds of labelling can be illustrated with a simple example. Consider labelling a simple database collected from children which is to be labelled at the phonetic level and phoneme, syllable and word labels are added semi-automatically. Because of the variablilty of children's speech it is desirable to record some aspects of voice quality in the utterance description, however this information generally does not belong at the phonetic level but at the Word or Utterance level. With the EMU labeller the user can augment the automatically

```
proc CantonSyllableLevel {hier} {
    ## make all segments at the Mu level children of one segment at
    ## the syllable level, the label is the concatenation of the Mu labels
    set segments [$hier segments Mu]
    set labels {}
    foreach s $segments {
        lappend labels [$hier seginfo $s label Mu]
    }
    set label [join $labels ""]
    ## create a new segment with this label at the syllable level
    set newseg [$hier append Syllable $label]
    ## and add this new segment as the parent of all Mu segments
    $hier seginfo $newseg children Mu $segments
}
```

Figure 1: An example Tcl script which adds Syllable labels to hierarchies in the Canton database. Lines beginning with # are comments.

derived word or utterance level labels with voice quality labels to indicate, for example, whispering of the relevant segment. Without the integration of hierarchical labelling this kind of mark-up would require some kind of non-phonetic label to be inserted at the phonetic level, or for the user to make a note of the problem and annotate the utterance at a later time.

Figure 2 shows the hierarchical and sequential labellers side-by-side for an utterance from the Cantonese database.

DATABASE QUERIES

The core function of the EMU system is to query and extract data from a database. The result of a query to the database is a segment or event list which records the start and end time (or event time) for the matching segments or events along with their label and the name of the utterance in which they were found. Given this list, EMU can then be used to extract whatever data has been stored in the database, for example sampled speech data, kinematic data or derived data such as formant tracks.

A query to the database specifies conditions on the segments or events which should be returned. The simplest query just gives the required label or class of labels, for example `Phonetic=p` would match all Phonetic level segments labelled `p`. Queries can also include constraints involving sequence and hierarchical relations within the utterance. Sequence queries match more than one segment and the resulting segment list can contain either all or part of the sequence matched. For example, to find all stop+vowel pairs we could use the query `[Phoneme=stop -> Phoneme=vowel]`, to find only the vowels preceded by stops the query is modified by marking the target segment with a hash sign: `[Phoneme=stop -> #Phoneme=vowel]`.

Hierarchical queries constrain the target segment based on parent/child relations defined in the hierarchical utterance description. As an example, to find all stop consonants which start a syllable where that syllable is stressed we would use the query: `[Phoneme=stop&Start(Syllable,Phoneme) ^ Syllable=S]`. Hierarchical and sequence queries can be combined to constrain the target segment further.

In some cases, it is difficult or impossible to express the desired segmental constraint in the EMU query language; here the scripting features of EMU become useful again as a simple procedural script can be written to extract the desired segments according to arbitrary criteria.

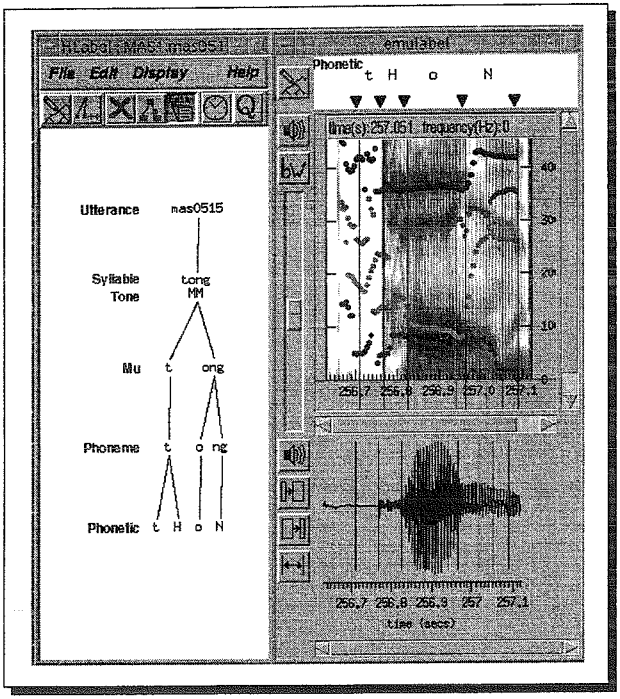

Figure 2: The EMU labeller showing both hierarchical and sequential views of an utterance.

DATA ANALYSIS

Having selected a set of speech segments from the database, the next step is to import the data into an analysis tool. In MU+ the only choice was to use the extensive set of routines developed for the Splus statistical/graphing package for data analysis. EMU is entirely compatible with MU+ in respect of these routines and the Splus environment forms the backbone of speech data analysis within SHLRC. However Splus is a commercial package and is not available to all laboratories, nor is it necessarily the best analysis tool for all applications. With EMU we have detached the speech database functionality from the data analysis tools and it is now possible to import data from the database to any tool via text files. Special purpose scripts could be written to interface to any data analysis package (for example MatLab or Microsoft Excel).

So, for example, one could query the database for all vowel segments in strong syllables and extract the formant values at the midpoint of each vowel. This data could then be imported into a graphing application and used to generate a plot of F1 vs. F2 for the given vowels.

In addition to the library of Splus routines tailored for speech data analysis we have implemented a similar set of extensions to the freely available XlispStat package (Tierney 1990). This system has then been used extensively in teaching and forms part of the software distributed with our forthcoming book (Harrington and Cassidy 1997). One advantage of XlispStat over Splus is that it allows one to build dynamic graphical applications; we have used this facility in our teaching materials to build interactive displays showing, for example, the different normalisation strategies that can be applied to vowel formant data.

It is always possible to interface directly to the EMU C++ library and perform data analysis directly in C++. Alterna-

tively scripts could be written in Tcl/Tk to accomplish less speed intensive tasks.

EXAMPLE DATABASES

Finally we will briefly describe two databases which have been constructed recently using the facilities of EMU. The first is a database of Cantonese collected by researchers in Hong Kong (Harrington and So 1994), the second is a collection of kinematic data which is being used in an ongoing study of vowel lengthening (Harrington, Fletcher, and Roberts 1995)

Cantonese

The structure of the Cantonese database is simple as can be seen from the example utterance shown in Figure 2: the data is hand labelled phonetically and higher level labels are generated automatically, except for the Tone label which is selected manually once the hierarchy is in place. This database is interesting in the context of this paper because it was a primary motivation for the re-design of the MU+ system. The problems we encountered when attempting to build this database using the older software included difficulties with specifying the mapping from Phonetic to Phoneme levels (MU+ allowed a certain kind of automatic mapping which didn't fit this problem) and syllabification.

Using EMU, the hierarchies for this database were built using a combination of rewrite rules and simple procedures (one procedure is shown in Figure 1 above). The rewrite rules are used to map the Phonetic level onto the Phoneme level, some examples are shown in Figure 3. The procedures are used to create the Mu, Syllable and Utterance levels and in all cases are quite simple.

```
G H -> gw
N -> ng
y -> yu
I -> i
<s=stop> H -> <s>
<a=affricative> H -> <a>
```

Figure 3: Examples of the rewrite rules used to map the Phonetic level onto the Phoneme level for the Cantonese database. The final two rules use variables to match categories of labels defined in the database template. In the first of these for example, any stop segment followed by an `H` segment will be rewritten as the stop. The effect of this can be seen in Figure 2 where `t H` at the Phonetic level has been re-written as `t` at the Phoneme level.

Kinematic

This database is being used in a study of vowel lengthening and consists of a number of utterances all of which correspond to a simple pattern. Only three words in each utterance are labelled at the Phoneme level, spaces are marked here with a hash sign (#). The research involves analysis of a trace of jaw displacement over time and in addition to marking the start and end times of each phoneme segment, the jaw trace and a computed trace of jaw velocity are used to mark prosodic events. Opening and closing gestures are marked for each target vowel at the JawGesture level. For each gesture, the time of the peak velocity is marked as an event on the JawVelocity level. These segments and events are then integrated into an hierarchical description as illustrated in Figure 4.

In this case the hierarchical description is build using a series of lookup tables since the structure of each utterance is fixed. Notice that in addition to the Text of each word, segments at this level are labelled as to whether they are nuclear accented, phase final and pre-pausal. The pre-pausal label is pre-set to a default value but may be modified by the labeller based on listening to the utterance. This then allows queries to the database to select open/close gesture pairs based on any of these criteria.

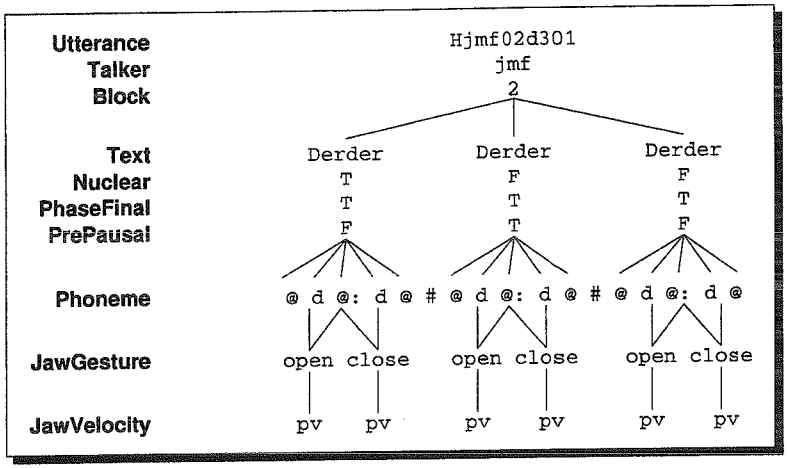

Figure 4: A hierarchy from the kinematic database.

One development of this application would be to semi-automatically place the jaw gesture labels given the phoneme labels – this is a simple peak/trough finding problem with some heuristics since there are known to be a fixed number of peaks/troughs for each gesture. Currently, for historical reasons,the project uses an Splus program to do this and hence it is not integrated with the labeller as it could be if it were written in Tcl/Tk.

SUMMARY

The re-implemented EMU system is inherently more flexible than its predecessor and offers a new range of possibilities in corpus based speech research. The integration of hierarchical and sequential labelling is a key feature of this system which allows a much richer annotation to be made on any speech signal. The querying and data extraction facilities have been streamlined and could now interface to any data analysis package although special interfaces exist for Splus and XlispStat. The system is portable to the major computing platforms and currently exists in versions for Unix and Microsoft Windows.

For more information about EMU, contact the authors or visit the SHLRC home-page http://www.shlrc.mq.edu.au/.

References

Harrington, J. and S. Cassidy (1997). Techniques in Speech Acoustics. Kluwer. Forthcoming.

Harrington, J., S. Cassidy, J. Fletcher, and A. McVeigh (1993). The mu+ speech database system. Computer Speech and Language 7, 305–331.

Harrington, J., J. Fletcher, and C. Roberts (1995). Coarticulation and the accented/unaccented distinction: evidence from jaw movement data. Journal of Phonetics, 305–322.

Harrington, J. and L. So (1994). Some design criteria in segmenting and labelling a database of spoken cantonese. In Proceedings of the Fifth International Conference on Speech Science and Technology, Volume 1, pp. 215–220.

Ousterhout, J. K. (1994). Tcl and the Tk Toolkit. Addison Wesley.

Tierney, L. (1990). Lisp-Stat: an object oriented environment for statistical computing and dynamic graphics. Wiley.