Managing speech databases with emuR and the EMU-webApp

Raphael Winkelmann

1Institute of Phonetics and Speech Processing, Ludwig Maximilian University of Munich

raphael@phonetik.uni-muenchen.de

Abstract

As is the nature of the discipline, a majority of speech and language researchers spend a large amount of their time acquiring and transforming data into analyzable and interpretable forms to gain a better understanding of a certain subject matter. In this paper we present a collection of tools that aid the researcher in this sometimes tedious and error-prone process. The tools presented here are part of the next iteration of the EMU speech database management system which aims to be as close to an all-in-one solution for generating, manipulating, querying, analyzing and managing speech databases as possible.

Index Terms: EMU, speech databases, web application

1. Introduction

Anyone fortunate enough to work with speech databases knows how laborious the chore of generating, manually annotating and eventually analyzing them is. Researchers usually have to familiarize themselves with a plethora of different tools to perform various tasks at the different stages of the process, which can be a very cumbersome and sometimes daunting process.

In this paper we present researchers with a start-to-finish set of tools for segmentation, visualization, signal processing, querying of hierarchical and sequential structures and statistical analysis of speech segments. These tools are part of the next iteration of the EMU speech database management system (see [1] & [2] for information on the legacy EMU system) and are mostly accessible directly within the R language for statistical computing [3]. This allows researchers to work in a familiar environment and facilitates the direct statistical analysis of the data. In addition, the system has been designed for maximum efficiency in collaborative research efforts in which a number of group members work simultaneously on annotating speech databases.

2. System architecture overview

As mentioned above, the core of the new system is based in R. We have developed two R packages (the gray items in Figure 1); one being the wrassp package [4] to handle the speech signal processing and the other being the emuR package [5], which handles database manipulation and querying as well as serving as a local websocket server to connect to the EMU-webApp. This infrastructure provides researchers with a self-contained system to query databases, perform signal processing and perform statistical analysis all within the same environment.

3. DB design

The database structure is basically a set of files and folders that adhere to a certain structure and naming convention (see Fig. 2).

The database root directory contains a single _DBconfig.json file which, as the name implies, contains the configuration options of the database such as its level definitions, how these levels are linked in the database hierarchy and what is displayed in the EMU-webApp. The database root folder also contains session folders (ending with _ses). These session folders can be used to logically group the _bndl folders described below.

Each session folder can contain any number of _bndl folders. All the files belonging to a bundle, for example a recorded sentence, are stored in the corresponding bundle folder. This includes the actual recording (.wav) and derived signal files in the SSFF format [6] such as formants (.fms) or the fundamental frequency (.f0), both of which can be generated using the wrassp package. Each bundle folder also contains the annotation file (_annot.json) of that bundle. This file, as the name implies, contains the actual annotations including the hierarchical linking information. JSON schema files are provided to ensure the syntactic integrity of the database (https://github.com/IPS-LMU/EMU-webApp/tree/master/dist/schemaFiles).

4. Managing / analyzing the database

The emuR package allows the researcher to perform CRUD (Create, Read, Update and Delete) operations for various properties and entities of the database. This allows for complete control over the database structure from within the user’s R session as well as allowing the user to inspect the structure of the database.

One of the most powerful features of the package is the ability to query a database for speech segments based on the
Figure 2: DB file structure

exampleDB/
  exampleDB
  DBconfig.json
  0001
    ses/
      bundle1_bndl/
      bundle2_bndl/
        bundle2.wav
        bundle2_annom.json
        bundle2.fms
        bundle2.f0
      ...
  bundle3_bndl/
  ...
  0003_sess/
  ...

sequential and hierarchical structure of the utterance in which they occur. Querying the database is performed using a slightly updated version of the Emu Query Language [2] called EQL2, which enables the researcher to use a simple query syntax to extract the desired items. Based on the extracted items, emuR provides a mechanism to extract derived speech signals such as formant values either from pre-calculated SSFF files but also has the ability to calculate these signals on-the-fly using functions provided by the wrassp package. This enables the researcher to quickly extract relevant information depending on her or his focus of interest. Further, emuR provides a large amount of functions that aid the researcher in inspecting and analyzing the data. These functions where largely adopted from the legacy EMU system and are described in [1].

5. Visualizing / editing the database

Researcher can easily host databases to the EMU-webApp (http://ips-lmu.github.io/EMU-webApp/) from within their R session using the functionality of the emuR package to act as a websocket server that implements the EMU-webApp websocket protocol [7]. The EMU-webApp is a fully fledged browser-based labeling and correction tool that offers a multitude of labeling and visualization features. These features include unlimited undo/redo, formant correction capabilities, snap preselected boundary to nearest top/bottom boundary, snap preselected boundary to nearest zero crossing and many more. The web application is able to render everything directly in the client’s browser, including calculating and rendering the spectrogram, as it is written entirely using HTML, CSS and JavaScript. This means it can also be used as a standalone labeling application as it does not require any server-side calculations or rendering. Further, it is designed to interact with any websocket server that implements the EMU-webApp websocket protocol, which enables it to be used as a labeling tool for collaborative annotation efforts. A database hosted by a single server can be made available to multiple users working on separate accounts logging individual annotations, time and date of changes and other activities such as comments added to problematic cases. Tasks can be allocated to and unlocked for each individual user by a central person such as the project leader. As such, user management in collaborative projects is substantially simplified and trackable compared with other currently available software for annotation and labeling.

A major advantage of the EMU system is its ability to deal with complex hierarchical annotation structures. The EMU-webApp enables the user to visualize and edit these annotation structures while the emuR package provides the necessary mechanism to query them. Utilizing these hierarchical annotations by linking various items to one another enables more powerful queries to be performed and reduces the error-prone procedure of trying to keep boundaries aligned.

6. Discussion / future work

In this paper, we presented a brief overview of tools for managing speech databases that ease the speech and language researcher’s interactions with the database. In future, we hope to offer simple interoperability with a variety of different tools to ease the integration of the tools presented here into the workflow of creating speech databases. Tools that will include this interoperability include the SpeechRecorder software [8] as well as WebMAUS [9]. This will enable a complete chain of tools from the initial prompted recordings (SpeechRecorder) to automatically annotating the database using forced alignment (WebMAUS) to manually correcting / enhancing and then eventually inspecting and evaluating the data (emuR, EMU-webApp, wrassp).

7. Acknowledgements

Research supported by the European Research Council Grant (295573) to Jonathan Harrington.

8. References