Towards an Acoustic-Phonetic HUES-HONOR Database

Latané Bullock
latanebullock@gmail.com

LING499 Final Project Report
Rice University

May 7, 2020

Abstract

We address one of the primary obstacles in acoustic phonetics research – ample, high-quality phonetic measurements – by designing a system that leverages open-source software and the discourse transcriptions of a pre-existing corpus to create a highly structured, richly annotated queryable database. Interviews recorded as part of the Harvey Oral Narratives on Record (HONOR; Englebretson, Kemmer, & Niedzielski, 2020) project represent a large body of data on which phonetic analysis is currently impractical. The present paper proposes a software system to translate the discourse-level orthographic transcription of the HONOR spoken language corpus into a segment-level phonetic database. The low-resolution two-minute time stamps of the current database, along with the discourse-level transcript and audio, are used as inputs to an open-source forced alignment software package, the Penn Phonetics Lab Forced Aligner (P2FA). The resulting output is a series segment-level Praat TextGrids. The TextGrids are converted to a more structured, queryable format with the EMU Speech Database Management System (EMU-SDMS; Winkelmann, Harrington, & Jänsch, 2017), a domain-specific tool which integrates the spoken lanuage research workflow into a single software environment. The EMU-SDMS database format is customizable to dataset-specific annotations, but, as a minimum, each segment includes a label, onset and offset time stamps, and its hierarchical relationship to other annotated linguistic levels. Stored in an EMU-SDMS format, the HONOR data is highly accessible for phonetic investigations. We justify our methodology by detailing software design choices and discuss the tradeoffs of various data formats of the new database.

For demonstration purposes, we convert and load five HONOR recordings into an EMU-SDMS database, resulting in more than 130,000 labels corresponding to lexical items and phones. The integrity of the generated annotations is then confirmed with evidence in the database for a well-known phenomenon: prenasal /æ/ raising.

While the conversion system and resulting database are not in their final form, this work paves most of the way towards an acoustic phonetic HUES-HONOR database. The combination of ample data, complex query capabilities, and efficient signal processing makes the proposed HONOR EMU-SDMS database a valuable resource for phonetics researchers. We foresee the database benefiting future research in a) American English phonetics, b) dialectal shift in the United States, and c) local variation in the Greater Houston area, as the HONOR corpus provides a snapshot of Houston's intricate linguistic landscape. Finally, the methodology and software tools detailed in this report are of use to those tasked with generating and managing a spoken language corpus.

1 Introduction

In the wake of Hurricane Harvey in 2017, the Rice Linguistics Department launched an effort to document experiences of Houstonians in the time leading up to, during, and after the storm. To document the narratives, nearly 100 interviews were conducted with citizens in the Greater Houston area, eventually spanning a wide range of ages, ethnic backgrounds, and geographic areas. The resulting high-quality audio recordings were then

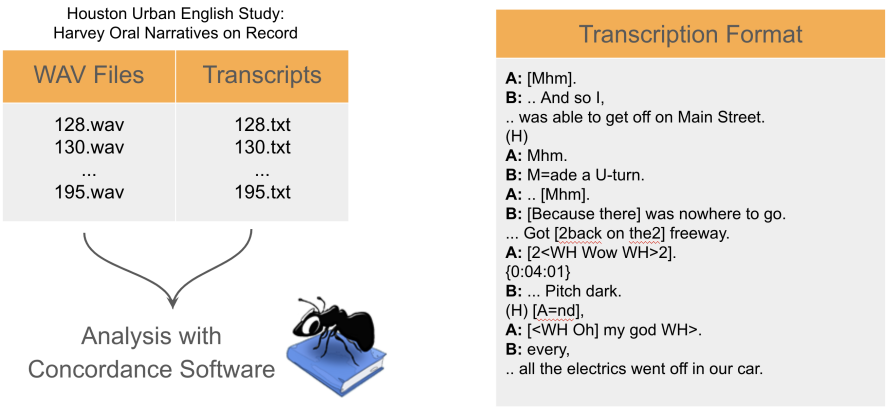

Figure 1: The structure of the existing HUES-HONOR corpus. Left: Each WAV recording is paired with a text file transcription. Discourse analyses are performed with concordance software such as AntConc (Anthony, 2004). Right: Excerpt from a transcript to demonstrate the format of the transcriptions. Note the time stamp {0:04:01} approximately four minutes into the recording.

compiled into a corpus now known as the Harvey Oral Narratives on Record (HONOR) and made part of the longer-term Houston Urban English Study (HUES; Englebretson et al., 2020). Students and faculty with linguistic training transcribed the recordings for discourse-level analyses, encoding laughter, exhalations, inhalations, tongue clicks, and other information relevant to each intonation unit. The current database structure and transcription format are depicted in Figure 1.

For the researcher interested in acoustic phonetic analyses, the HONOR interviews represent a rich body of data to draw on for empirical experimentation. The high-quality audio, breadth of speaker socioeconomic backgrounds, and casual setting result in a set of recordings of great interest for sociolinguists and phoneticians alike. However, the lack of phone-level annotation presents a large barrier to testing phonetic hypotheses. In fact, the present project stemmed from a question regarding phonetic convergence: *Is there evidence to support the hypothesis that the interviewer and interviewee's vowel spaces tend towards each other as an interview progresses?* When designing a pilot experiment to tackle this question, we realized the current state of the dataset offers little way to obtain the measurements of interest in a reasonable amount of time. One would have to manually peruse the raw transcriptions, select individual words that contain tokens of interest, find the nearest time stamps (every two minutes), listen to the corresponding audio recording near the stamp, wait for the word to be said, and finally take note of the perceived onset and offset times of the phone. Only then could one use Praat (Boersma & Weenink, 2007) to extract the desired measurements.

Rather than perform this tedious single-use-case experiment, we elected to take on a more generalized project: generating a set of phonetic annotations for the complete HONOR database. Although not directly advancing the understanding of language, we considered the project an opportunity to contribute a valuable tool to greatly facilitate future linguistics research. This report outlines the design process of a system that leverages

open-source software and the discourse transcriptions of the HONOR corpus to create a structured, queryable database to be used in the acoustic-phonetic research community.

2 Background

In this section, we touch on literature and related work that addresses the question of how to store annotated spoken language data. Much of the work in the literature on 'phonetic databases' is concerned with compiling speech data with the intent of using it to train automatic speech recognizers and speech synthesizers (e.g., Lamel, Kassel, & Seneff, 1989; Moreno Bilbao et al., 1993). We, however, will focus on work that is concerned with compiling speech databases for linguistic research. While there is some superficial overlap in these domains, their differing end-goals result in conflicting software design decisions. For example, user friendliness is paramount in linguistics research software, whereas usability might be sacrificed for increased statistical or modeling power in automatic speech recognition.

Recently, a wave of new tools aimed at providing a more integrated, streamlined experience has emerged in the spoken language research community to facilitate experimentation. Among these are Speech Corpus Tools (McAuliffe & Sonderegger, 2016), LaBB-CAT (Fromont & Hay, 2012), and the iterations of the Phon software (Hedlund & Rose, 2019; Rose & MacWhinney, 2014; Rose et al., 2006). Speech Corpus Tools (SCT) is an "application for working with speech datasets, with a focus on large-scale speech corpora" (McAuliffe & Sonderegger, 2016). The creators of SCT emphasize the system's flexibility with input formats, supporting Buckeye (Pitt, Johnson, Hume, Kiesling, & Raymond, 2005), TIMIT (Pitt et al., 2005), and force-aligned TextGrids (Boersma & Weenink, 2007). The underlying data structure in SCT is provided by PolyglotDB (McAuliffe, Stengel-Eskin, Socolof, & Sonderegger, 2017), allowing a user to conveniently query for a set of tokens. PolyglotDB is a Python package designed only for data storage, requiring a user interface to be built on top of its application program interface. Together, SCT and PolyglotDB offer an easy-to-use solution to large-scale speech data storage and querying. The primary drawback of these systems is that they do not offer statistical analysis. Users must export the queries and find third-party software to perform the digital signal processing required for measurements such as formant values (McAuliffe & Sonderegger, 2016). LaBB-CAT (for Language, Brain and Behaviour – Corpus Analysis Tool) is the most recent generation of the 'ONZE Miner' software (Fromont & Hay, 2008). Most notably, these tools pioneered browser-based user interfaces for spoken language databases. Phon is a database annotation and management tool developed as a part of the CHILDES (MacWhinney, 2000) project to facilitate child development research by providing "functionality for the elaboration, analysis and sharing of phonological data" (Rose et al., 2006). Like SCT, however, both LaBB-CAT and Phon expect users to export data queries for analysis.

One of the most recent contributions to spoken language corpora management is the release of the updated EMU Speech Database Management System (EMU-SDMS) (Winkel-

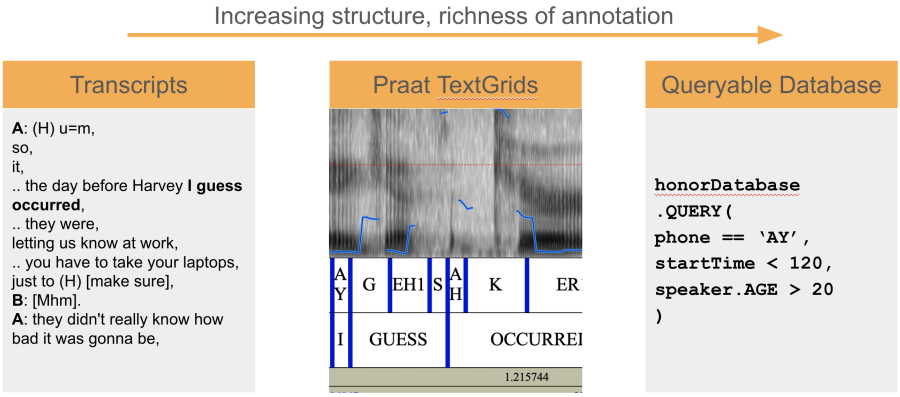

Figure 2: High-level depictions of the format of transcript and audio data at the start, middle, and end of processing. The raw transcripts are transformed into a queryable database via forced alignment. The far-right panel is a toy example to illustrate the types of queries that a researcher could perform on the dataset: 'find all instances of /aɪ/ spoken by someone over 20 years old which occur in the first two minutes of a recording.'

mann et al., 2017). In development since the early 90s (Cassidy & Harrington, 2001; Harrington, Cassidy, Fletcher, & McVeigh, 1993), the primary advantage of this database system is that it centralizes all tools required for speech corpus analysis - from annotation generation to storage and querying to signal processing to visualization. In this way, it reduces the burden on the researcher of becoming familiar with many steep-learning-curve software tools. The technical specifications of the software are more fully discussed in Section 3, as the EMU-SDMS is the technology employed in this work.

3 Methodology and Technologies

In this section, we detail the technologies employed to convert the available HONOR recordings and transcripts into a queryable phonetic database.

3.1 System Overview

A series of scripts were chained together to create a 'data pipeline' that takes a WAV file and corresponding transcript, aligns them with high-resolution time stamps, and integrates them into a database to be used for analysis. This is illustrated in Figure 2, where the format of the data at each step is depicted. The process begins with simple WAV and text files, and results in a set of richly annotated recordings suitable for both low-level acoustic and phonetic analysis, and lexical frequency and co-occurrence analyses. Section 3.2 presents the first step in the process (converting raw transcripts to time-aligned Praat TextGrids) and Section 3.3 presents our methodology for the second (converting TextGrids to an all-in-one speech sciences research database).

3.2 Transcripts to Praat TextGrids

The first step towards a phonetic database is generating a sequence of time-aligned phones from the recordings. The most familiar tools designed for this purpose are called 'forced aligners.' Although there are many open-source software resources performing forced alignment, one of the most common is the Penn Phonetics Lab Forced Aligner (P2FA) (Yuan & Liberman, 2008). P2FA is an "automatic phonetic alignment toolkit based on Hidden Markov Toolkit (HTK)," containing Python scripts which utilize acoustic models of American English. More information on the software can be found here in the README.

P2FA outputs Praat TextGrids with two tiers: 'phone' and 'word'. This not only encodes word-level time stamps (previously unavailable), but also specifies the phonemes within the word and their onset and offset times. The word-to-phone mapping makes use of the CMU pronunciation dictionary (Carnegie Mellon University, 2000). As no dictionary is comprehensive, entries must be added manually for novel lexical items present in the transcript. In the HONOR interviews, there are many cases of Spanish words which need to be added to the dictionary. Otherwise, they are omitted from the TextGrids. Listing 1 presents a few of the words and corresponding pronunciations that were added to the dictionary.

```
HEALTHWISE  HH EH1 L TH W AY2 Z
SUBE S UW1 B EH1
MEDICAMENTO M EH1 D IH0 K AH0 M EH1 N T OW0
MENTALITIES M EH1 N T AE1 L IH1 T IH0 Z
```

Listing 1: Examples of lexical entries added to the CMU Dictionary located in the P2FA code repository at `p2fa/model/dict`. Both Spanish and English entries were added with their ARPABET phoneme encoding.

The primary challenge with forced aligners is data wrangling – transforming and formatting preexisting data into a format that is most useful for downstream processes. In our case, P2FA accepts a very specific format of transcript. It requires every word to be capitalized and on a new line in the text file, with no standard orthographic markers or special characters. We must, therefore, remove all special characters like commas, parentheses, and brackets while preserving the words around and inside of them. An excerpt of an unprocessed transcript along with its resulting P2FA formatting is shown in Figure 3. While this may seem like a simple task, implementing a script to perform this task can be challenging. Any inconsistencies in the raw transcripts result in either software errors while processing or in data flaws in the output files. Other complicating factors include overlapped speech, dash characters used for both hyphens (as in 'far-off') and to denote interruption in an intonation unit, and the need to translate special ways of encoding laughter, breathing, and silence. The primary commands used to format the transcripts are `sed` and `tr`. Two lines from the conversion script are shown below in Listing 2. The full code can be found at the project's public Github repository.

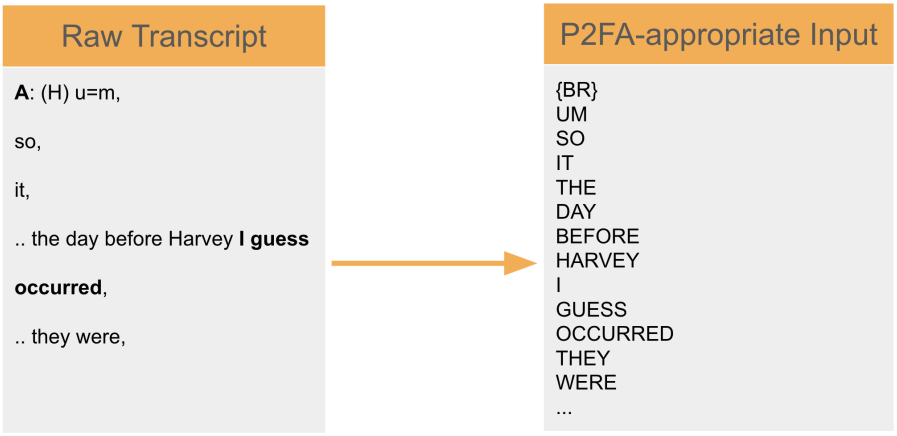

Figure 3: Transcript format before and after passed to `strip-transcript.sh`. The output format matches that expected by the Penn Phonetics Lab Forced Aligner. In the first line, "(H)" becomes "{BR}" in the P2FA format. Every word in raw transcript is presented on a new line in the P2FA transcript, with all special characters and speaker labels removed. This transformation is primarily done with the `sed` (**s**tream **ed**itor) and `tr` (**tr**ansliterate) Unix commands.

```
...
tr -d ',=[].<>?%~' < $1 | \
sed -E 's/[^A-Za-z][@]+[^A-Za-z]/ {LG} /' |
...
```

Listing 2: Excerpt of the Bash script `transcript2textGrid-p2fa.sh`, used to format transcripts before passing them to P2FA. The first line uses the `tr` command to remove all commas, equal sings, square brackets, periods, question marks, and other characters. The stream of text is then piped to the `sed` command to convert laughter in the raw transcripts (one or more '@' characters) into P2FA's expected encoding ('{LG}'). The piping continues until the text is appropriately formatted.

In addition to altering the transcript format for input, P2FA requires shortening the length of input audio. The longer the recording, the more difficult the alignment task is and the more errors (misalignments) are generated. With this consideration, the two-minute time stamps from manual transcription are used to chunk the audio files and text transcripts, each of which are individually fed to P2FA. After obtaining the full set of two-minute TextGrids for a full recordings, they are loaded as Python objects using the `PraatIO` package (Mahrt, 2016) to be spliced together. After concatenation, the full-length TextGrid is saved back to a text file.

The data processing described in this section is encapsulated within a single script, which calls other scripts to perform formatting, chunking, alignment, and splicing. To process a new WAV-TXT interview pair, simply call `$ ./transcript2textGrid-p2fa.sh [interview number]`. The WAV file and text file must be named `{interviewNumber}.wav` and `{interviewNumber}.txt`, and reside within the `data` directory specified inside of

the script. The default location for a Unix-like machine is `~/Documents/data/HONOR`.

3.3 TextGrids to Queryable Database with EMU-SDMS

TextGrids, even by themselves, represent a reasonably structured set of data on which one could perform phonetic research. They are, in fact, sometimes used to generate and store annotations for a set of audio recordings. Further, because they are Praat's 'native language', some signal processing can be performed without external tools. However, complex queries on the data remain impractical. This final step, of transforming the TextGrids into a queryable, more accessible format, warranted a significant amount of research into database systems. Eventually, a few speech-specific databases surfaced, and along with them other tools for not only storing but also querying and manipulating speech corpora (see Section 2). Among these, The EMU Speech Database Management System (EMU-SDMS; Winkelmann et al., 2017) arose as the clear winner because it is the only all-in-one solution to speech database management. The authors of the software describe it as a collection of tools centered around the R language (R Core Team, 2017) that are designed in user-friendly manner while maintaining the flexibility and statistical power of advanced speech database tools. The EMU-SDMS is advantageous to alternative databases in a number of ways. Its key features, which support our original goal of expediting and facilitating phonetic research, include:

- **TextGrid compatibility**: TextGrids are one of a few import options to generate an emuDB database. Similarly, users can export an emuDB to a series of TextGrids if it is necessary.
- **Built-in Signal Processing**: The EMU-SDMS `wrassp` R package provides researchers with both bulk and on-the-fly signal processing, incorporating many of Praat's implementations via the `PraatR` R package. It "[handles] speech signal files in most common audio formats and [performs] signal analyses common in the phonetic and speech sciences" (Winkelmann, 2016). It includes functions like frequency filtering, Cepstral-smoothed Fourier analysis, and pitch analysis.
- **Web-based User Interface**: While analyses are performed in R, annotations are generated and manipulated via Web application. The EMU-webApp serves the database over web protocols (`websocket`), allowing both local and remote users to access the data. Along with the advantage of widespread familiarity with browser interface, the Web application GUI enables collaborative annotation and verification efforts. In our case, this might be useful if a few research assistants are generating speaker labels or verifying existing alignments at one time or in remote locations.
- **Updated Technologies and Ongoing Development**: EMU-SDMS is an ongoing project, with code developments every few weeks. This is an advantage over other deprecated projects, which suffer from outdated software and little technical support if

bugs are found. Because the team recently developed the new EMU-SMDS, it adheres to modern software standards. For example, they use the JSON format to store data in the emuDB as this renders the database compatible with other technologies, including Web.

- **Extensive Documentation**: The EMU-SDMS is very well documented in the EMU-SDMS online manual (Winkelmann, 2016). This organized, accessible plain-text presentation of the underlying software is a rarity in large-scale coding projects. It makes for quicker orientation, debugging, and feature exploration. The manual provides users with digestible technical specifications regarding each of EMU-SDMS's subsystems, import and export options, use case examples, and even supplementary speech-specific plotting routines in R.

Once EMU-SDMS and its dependencies are installed, the conversion from TextGrids to the emuDB format is performed in just a few lines of code, as shown in Listing 3. The `convert_TextGridCollection()` function provided by the emuR package works behind the scenes to load each WAV-TextGrid pair and generate the corresponding internal emuDB representation. This creates a permanent directory on the filesystem that houses the database. These data are then modified and accessed in efficient, systematic ways via the emuDB user interfaces (Emu-WebApp and emuR).

```
# load the emuR package from EMU-SDMS
library(emuR)

# set path to the data directory containing the wav files and TextGrids
data_dir = "~/Documents/data/HONOR/wav-textgrids-comb"

# convert TextGrid collection to the emuDB format
convert_TextGridCollection(dir = data_dir,
                           dbName = "HONOR",
                           targetDir = "~/Documents/data/HONOR",
                           tierNames = c("word", "phone"))
```

Listing 3: Excerpt from the `emuR-hueshonor.R` script. Once the `emuR` package is loaded, we specify the location of the WAV files and their P2FA-generated TextGrids. EMU-SDMS provides the `convert_TextGridCollection()` routine to import the TextGrids and convert them to the native emuDB format. The TextGrid annotations do not encode hierarchy explicitly and therefore, when imported, do not take full advantage of emuDB's annotation richness. However, the annotations could be added manually on a portion of the database, should such hierarchical links be necessary to answer a specific research question.

Once the data are in the EMU-SDMS database, the researcher is not required to use any other software. One can annotate a new recording in the EMU-webApp, visually inspect existing annotations and modify them, query the data for tokens of interest in the

EMU Query Language (which is housed in an R package), perform on-the-fly signal processing to obtain token measurements, and employ all of the statistical and visualization techniques available in the R language (R Core Team, 2017; Winkelmann et al., 2017).

4 Proof of concept

In this section, we test an investigative question on five HONOR interviews to a) establish the quality of the proposed system and b) illustrate the efficiency with which future researchers may employ the system. We are not able to address our original curiosity regarding phonetic convergence in the HONOR recordings, as the data are not speaker-labeled. We will attempt to answer a different question: *is there evidence to suggest speakers raise /æ/ in the prenasal environment in the HONOR interviews?* Whereas phonetic convergence is not always empirically evident, nasalization is very predictable and is therefore perhaps more suitable for this demonstration because we can verify the result. As with any nasalized vowel, we expect to find prenasal /æ/ has a lower F1 than /æ/ realized in other environments (Johnson, 2004; Mielke, Carignan, & Thomas, 2017).

As a preliminary step, five HONOR interviews were loaded into the EMU-SDMS with the pipeline described in Section 3, resulting in a database with over 130,000 time-stamped labels (consisting of a lexical tier and a phone tier). To launch our investigation, we first query the database on the 'phone' level for instances of /æ/. There are 2886 such segments. We then obtain the right context of each segment, using it to filter the /æ/ segment list into prenasal instances of /æ/ and all other instances. The excerpt in Listing 4 implements this dataflow.

```
# query all segments containing the phone label AE1
segList_AE = query(honorDB_handle,
                   query = "phone == AE1")
# => results in a dataframe with 2886 segments
#    with onset and offset times of /ae/

# get the right context (phone to the right) of each of the AE segments
AE_rightContext = requery_seq(honorDB_handle, segList_AE,
                              offset = 1,
                              ignoreOutOfBounds = TRUE)
# => also results in a dataframe with 2886 segments

# separate the AE segment list into prenasal and other
nasals = c('N', 'M', 'NG')
AE_N = segList_AE[ (AE_rightContext$labels %in% nasals), ]
AE_nonN = segList_AE[ !(AE_rightContext$labels %in% nasals), ]
```

Listing 4: Retrieving instances of /æ/ in an emuDB database from five HONOR interviews. The database is queried on the 'phone' level for instances of /æ/ , which are then filtered into prenasal (`AE_N`) and non-prenasal contexts (`AE_nonN`).

Next, we compute the formants of those segments with emuR's `get_trackdata()` function. The `forest` (**for**mant **est**imation) parameter passed to the function specifies the measurement to be extracted for each segment (see Chapter 8 of the EMU-SDMS manual for a complete enumeration of available signal processing routines (Winkelmann, 2016)). The formants are then normalized and preprocessed for plotting. Finally, we make use of R's `ggplot` package for visualization. This is shown in Listing 5, and the resulting vowel space plot is shown in Figure 4. As expected, we note a lower F1 (raising) for prenasal /æ/ compared to other realizations of the phoneme. Thus, in just a few lines of R, we have compelling evidence to answer the question at hand. This ease of analysis scales and generalizes even with more nuanced investigations. We hope to have shown that the combination of ample tokens, complex queries, and quick signal processing makes the proposed HONOR EMU-SDMS database a valuable, even exciting, resource for acoustic phonetics researchers.

```
# get the formants for these segments,
# "forest" = just-in-time formant computation
AE_N_formants = get_trackdata(honorDB_handle,
                      AE_N,
                      onTheFlyFunctionName = "forest")
AE_nonN_formants = get_trackdata(honorDB_handle,
                                 AE_nonN,
                                 onTheFlyFunctionName = "forest")

# time normalize the formant values
AE_N_formants_norm = normalize_length(AE_N_formants)
AE_nonN_formants_norm = normalize_length(AE_nonN_formants)

...

# plot the prenasal and non-prenasal ellipses and their centroids
ggplot(AE_summary) +
  aes(x = T2, y = T1, label = labelsAE, col = labelsAE) +
  stat_ellipse() +
  scale_y_reverse() + scale_x_reverse() +
  labs(x = "F2 (Hz)", y = "F1 (Hz)") +
  theme(legend.position = "none") +
  geom_text(data = AE_centroid)
```

Listing 5: Extracting formant values for /æ/ instances retreived in Listing 4. The formants are computed with emuR's built-in functions. Average formants across the prenasal and nasal categories, along with covariance ellipses, are plotted with `ggplot` (see Figure 4).

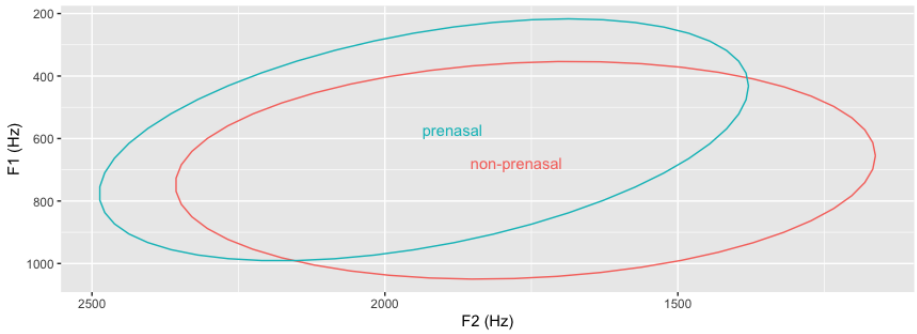

Figure 4: Prenasal and non-prenasal /æ/ average formant values plotted in the F1-F2 vowel space. As expected, the prenasal realizations are raised (lower F1) compared to other realizations. These data are from five HONOR recordings, independent of speaker. The plot is the result of the `ggplot` command in Listing 5.

5 Limitations and Future Work

While the present work has laid the groundwork for a phonetic HONOR database, we were unable to achieve a final, polished set of data. To perform robust analyses with more confidence and higher statistical power, a number of issues need to be resolved. This section outlines the current limitations of the system, serving as a guide to future work on the project.

The only issue identified in terms of data integrity lies in TextGrid splicing after forced alignment. A small error accumulates when concatenating the two-minute TextGrids into a full-interview TextGrid. The TextGrid thus lags behind the audio in the later portions of the recording. The time differences are less than a half-second even at the very end of a recording, but this clearly results in defective alignment when considering the timescale of a single phone (as short as 25 milliseconds). The error will be resolved by ensuring that the TextGrids are concatenated more precisely at the two-minute marks, rather than current process of chaining them back-to-back (using `PraatIO`). Once the alignment accumulation error is resolved, the remaining interviews can be added to the database. The only manual task to be performed for a new interview is to add previously-unseen lexical items to the CMU dictionary. Afterwards, the automated conversion takes less than five minutes to go from raw transcript to the emuDB database form.

A second issue – related to breadth of possible analysis rather than erroneous data – is that we were unable to retain the intonation-level speaker labels in the raw HONOR transcripts through P2FA's forced alignment. As a result, the database does not include annotations that would be required to inform socio-phonetic research or any other speaker-dependent questions. Lack of precise time-stamps in the transcripts also precludes the possibility of automatically generating them and integrating them into the final database. Two possible solutions can be explored to address this. The first, most reliable option is to hand-label the speakers. Annotations could be added directly to the database with the EMU-webApp or indirectly via Praat TextGrids. The clear downside for manual annotation is time and effort. Another option is to develop an optimization algorithm which

minimizes the 'distance' (in terms of a cost function) between the raw transcripts and the final aligned output by testing the different permutations of word-to-speaker mappings. The benefit of an algorithmic approach is generalizability to new transcripts and databases. This would, however, be a noisy process and warrant manual inspection/verification.

Finally, a more subjective limitation is the seemingly drastic departure from familiar tools to the EMU-SDMS. The time-honoured tool central to the field, Praat, is notably absent in the user experience of our proposed system. Praat's drawbacks (e.g., plain-text corruptible data storage and inability to query tiers), warranted looking into other, more integrated solutions to eliminate the burden of learning four to five different software environments and enabling efficient queries on the dataset. Praat's signal processing is certainly at work under the hood, though, and its familiar waveform-formant-annotation editor interface is reflected in the EMU-webApp. Overall, we hope to have shown that the advantages of the all-in-one EMU-SDMS outweigh the comfort of more familiar software.

6 Conclusion

The present work addresses one of the primary obstacles in acoustic phonetics research – ample, high-quality phonetic measurements – by generating an easy-to-use phonetic database from preexisting data. Although not directly advancing the understanding of language, we considered the project an opportunity to contribute a valuable tool to greatly facilitate future linguistics research. Our proposed system leverages open-source forced-alignment software and the discourse transcriptions of the HUES-HONOR interviews, and uses the EMU-SDMS as a means of storing and querying the phonetic data. This report has detailed the motivation and background for such a project, presented the data pipeline and corresponding code required to generate the database, demonstrated the use of EMU-SDMS in a proof-of-concept case example, and commented on the limitations of the system as it stands today.

The present work does not achieve a final, polished HUES-HONOR database. We have, however, paved most of the way towards such a resource by contributing foundational research and functional code to the effort. The combination of ample data, complex query capabilities, and efficient signal processing will make the proposed HONOR EMU-SDMS database a valuable resource for acoustic phonetics researchers. We foresee the database facilitating future research in a) American English phonetics, b) dialectal shift in the United States, and c) local variation in the Greater Houston area as the HONOR corpus provides a snapshot of Houston's intricate linguistic landscape.

References

Anthony, L. (2004). AntConc: A learner and classroom friendly, multi-platform corpus analysis toolkit. *Proceedings of IWLeL*, 7–13.

Boersma, P., & Weenink, D. (2007). PRAAT: Doing phonetics by computer (Version 5.3.51).

Carnegie Mellon University. (2000). *The CMU pronunciation dictionary.* Retrieved from http://www.speech.cs.cmu.edu.

Cassidy, S., & Harrington, J. (2001). Multi-level annotation in the Emu speech database management system. *Speech communication*, *33*(1-2), 61–77.

Englebretson, R., Kemmer, S., & Niedzielski, N. (2020). *HONOR (Harvey Oral Narratives on Record): A Corpus of Interviews from Hurricane Harvey.* (Online Resource)

Fromont, R., & Hay, J. (2008). ONZE Miner: the development of a browser-based research tool. *Corpora*, *3*(2), 173–193.

Fromont, R., & Hay, J. (2012). LaBB-CAT: An annotation store. In *Proceedings of the Australasian Language Technology Association Workshop 2012* (pp. 113–117).

Harrington, J., Cassidy, S., Fletcher, J., & McVeigh, A. (1993). The mu+ system for corpus based speech research. *Computer Speech & Language*, *7*(4), 305–331. (Publisher: Elsevier)

Hedlund, G., & Rose, Y. (2019). *Phon 3.0.* Retrieved from https://phon.ca

Johnson, K. (2004). Acoustic and auditory phonetics. *Phonetica*, *61*(1), 201.

Lamel, L., Kassel, R., & Seneff, S. (1989). Speech database development: Design and analysis of the acoustic-phonetic corpus..

MacWhinney, B. (2000). *The CHILDES Project: Tools for analyzing talk. Transcription format and programs.* (Vol. 1). Psychology Press.

Mahrt, T. (2016). *PraatIO.* Retrieved from https://github.com/timmahrt/praatIO

McAuliffe, M., & Sonderegger, M. (2016). *Speech Corpus Tools (SCT).* Retrieved from http://speech-corpus-tools.readthedocs.io/

McAuliffe, M., Stengel-Eskin, E., Socolof, M., & Sonderegger, M. (2017). Polyglot and Speech Corpus Tools: A System for Representing, Integrating, and Querying Speech Corpora. In *INTERSPEECH* (pp. 3887–3891).

Mielke, J., Carignan, C., & Thomas, E. R. (2017). The articulatory dynamics of pre-velar and pre-nasal/æ/-raising in English: An ultrasound study. *The Journal of the Acoustical Society of America*, *142*(1), 332–349.

Moreno Bilbao, M. A., Poig, D., Bonafonte Cávez, A., Lleida, E., Llisterri, J., Mariño Acebal, J. B., & Nadeu Camprubí, C. (1993). Albayzin speech database: design of the phonetic corpus. In (pp. 175–178). EUROSPEECH.

Pitt, M. A., Johnson, K., Hume, E., Kiesling, S., & Raymond, W. (2005). The Buckeye corpus of conversational speech: Labeling conventions and a test of transcriber reliability. *Speech Communication*, *45*(1), 89–95. (Publisher: Elsevier)

R Core Team. (2017). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

Rose, Y., & MacWhinney, B. (2014). The PhonBank Project: Data and software-assisted methods for the study of phonology and phonological development.

Rose, Y., MacWhinney, B., Byrne, R., Hedlund, G., Maddocks, K., O'Brien, P., & Wareham, T. (2006). Introducing Phon: A software solution for the study of phonological ac-

quisition. In *Proceedings of the Boston University Conference on Language Development* (Vol. 2006, p. 489). NIH Public Access.

Winkelmann, R. (2016). *The EMU-SDMS Manual.* Retrieved from https://ips-lmu.github.io/The-EMU-SDMS-Manual/index.html

Winkelmann, R., Harrington, J., & Jänsch, K. (2017). EMU-SDMS: Advanced speech database management and analysis in R. *Computer Speech & Language*, *45*, 392–410. Retrieved 2020-04-28, from https://linkinghub.elsevier.com/retrieve/pii/S0885230816302601 doi: 10.1016/j.csl.2017.01.002

Yuan, J., & Liberman, M. (2008). Speaker identification on the SCOTUS corpus. *Journal of the Acoustical Society of America*, *123*(5), 3878.