Open-source, automatic speech analyses for long audio-recordings

Authors to be defined[1]

[1] LSCP, Département d'études cognitives, ENS, EHESS, CNRS, Université PSL

Author Note

Correspondence concerning this article should be addressed to Authors to be defined, 29, rue d'Ulm, 75005, Paris, France. E-mail: alecristia@gmail.com

Abstract

Recent years have seen the emergence of hardware facilitating daylong, child-centered audio-recordings. A major roadblock remains: Open-source, improvable software for annotating such massive datasets. In this paper, we present a first attempt to overturn this roadblock. Our system uses Virtual Machine technology, which allows packaging diverse computer code into a unit that is both self-installing and multi-platform. The current version contains a competitive set of tools including voice activity detection, talker kind diarization, classification of adult speech into child- versus adult-directed, unsupervised diarization routines, and syllable count estimation, which themselves have been drawn from open-source software. Performance of this system is compared against a current benchmark, the LENA^{TM} system. The present report together with the open source code it is based on could be used to extend this approach to other populations.

Keywords: Audio-recordings, Speech technology, Language production

Word count: The median number of published pages is 12 in this journal, so aim for 24 in this format, counting references

Open-source, automatic speech analyses for long audio-recordings

Info for us

Authorship.

- Author-level contributions: To be completed in discussion with people contributing code, analysis, writing...
- Acknowledgements/citations only:
  - AlF, ER, & JK contributed ancillary code still in use; note that all three are coauthors in Interspeech paper, so they could be acknowledged in BRM without being coauthors
  - All tools should be properly cited in the DiViMe site and the paper so their authors will not by default be coauthors of this paper
  - All datasets should be properly cited (in the DiViMe site if routinely used) and the paper so their authors will not by default be coauthors of this paper

Todo/done.

- run the analyses for talker diarization just to check it works
- run the ana for VCM just to check it works
- run the ana for WCE just to check it works
- run the eval just to check it works
- report any bugs or inaccuracies
- add the 4-class model (perhaps remove the yunitators) with Marvin
- run the analyses and calculate the results for role diarization
- compare that to LENA
- incorporate ADS/CDS into divime with Wassim
- run the ana for ADS/CDS just to check it works
- update the recipe section

Introduction

Research projects in linguistics, speech pathology, and other language sciences often collect and compare ecological data from different cultures and settings with a diverse set of recording devices. Recent years have seen the emergence of "daylong recordings", whereby researchers build datasets containing hundreds, often thousands of hours, of audiorecordings gathered with device worn by a person as they go about their normal day. The resulting, highly heterogeneous speech corpora truly deserve the "in the wild" label, and have been shown to test the limitations of even state-of-the-art speech processing algorithms ("The first DIHARD speech diarization challenge," n.d.). Moreover, there is basically no tool that is both open source and easy to use. In this paper, we present DiViMe, a virtual machine which (a) contains tools allowing broad, first-pass annotation of audiofiles in a completely automatized manner, (b) is easy to use, and (c) can be augmented in the future, in an open source manner.

The LENA™ Revolution

The field of language development has a long and fruitful history of reliance on data, including recorded data when the technology became available. With just a few exceptions, the most common format for data collection involved visiting families in their home and taping the child for an hour or so, often repeating such visits periodically. An interest in the child's production often led to prioritizing language-rich moments, by asking caregivers to interact with the child using some prop chosen to stimulate talk and interaction. While such data are invaluable, they cannot capture faithfully the sparse interactions that are probably most common in children's lives.

This changed two decades ago, when the LENATM Foundation developed a sturdy recording device that could collect audio for 16 hours at a time, and a set of software that can automatically analyze such long recordings. While there were other researchers who had started collecting these person-centric, long-form recordings (**???**, **???**), the LENA

Foundation's research was groundbreaking because it brought this technology within the reach to many more researchers, notably those who could not afford to do manual annotation of these extensive audio data.

With it came also a change in mindset, shifting focus from the socially, pragmatically, and linguistically informed annotations that could only be produced by highly trained assistants, to a view more appropriate to big data, and a focus on verbal behavior more generally. When viewed in this way, verbal behavior in particular, and the analysis of participant-centered sound scapes, may be relevant at all stages of life, but particularly useful for individuals diagnosed with psychological or neurodegenerative disorder. All of these goals would be aided by a broad, first-pass annotation that singles out vocalizations by the person wearing the recorder versus others, which would further provide broad strokes on everyday language use by and around that person.

Producing broad, first-pass annotations is not a trivial task

The difficulties in processing data such as child speech in a daily-life environment have been highlighted at by the results of a recent Interspeech Challenge called DiHARD, aimed at assessing the state-of-the-art performance when parsing audiorecordings containing spontaneous conversations. If a clip with such a conversation is played back to a human, they'll be able to easily tell how many people are talking, and when each of them starts and stops speaking (particularly if they are speaking a language known to the hearer, and if they themselves are well-known to the listener). Machines faced with the same task can perform dismally. Participants to the DiHARD Challenge scored blah blah

Available tools

LENA is available for a fee. A systematic review and targeted data analysis reveals performance not very good, particularly for other children and male adults. Some derived metrics are also not very good

TABLE SUMMARIZING LENA SCORES ON DIFFERENT METRICS, AGAINST OUR SCORES ON SAME METRICS table can also convey that we have some metrics lena doesn´t and vice versa

Thus, there is room for improvement both in terms of availability for researchers with limited funding, and of accuracy. Complex tools might lead to excellent performance, but do not benefit the larger scientific community as they should if they cannot be easily applied to reproduce experiments and to build on top of them.

Some excellent tools available, flexible, etc. But challenging to use. Kaldi recipes. installing and running such code is not always straightforward. In particular, integrating an open-source project into a local processing pipeline is a challenging task since file formats and environment settings might differ from one tool to another. Moreover, we have found training to be crucial as this is a very new and different domain, and thus free solutions do not work well. One definitely needs to retrain.

Other tools available and very easy to use (WebMaus), but cannot cope with these specific recordings, with their speaker changes, mumbling, lack of transcription, etc.

- WebMAUS
- LIUM in VM

Introduce the Speech Recognition Virtual Kitchen repository ("The speech recognition virtual kitchen," n.d.).

Present project goals

These observations motivated us to develop the ACLEW Diarization Virtual Machine - DiViMe for short. DiViMe follows in the Speech Recognition Virtual Kitchen's Plummer et al. (2014); ("The speech recognition virtual kitchen," n.d.) footsteps in that it is a virtual machine (VM) gathering speech processing tools inside a unified computational environment. As a result, it can be deployed on most host computer systems and offers a simple interface to run the integrated models within a global pipeline.

The global pipeline itself is directly inspired by the LENA™ products. Specifically, we wanted to provide tools for analyzing very long, highly ecological recordings. Daylong recordings are particularly interesting for the present project because they present a difficult diarization problem (and in the case of acquisition data, probably the hardest case imaginable), and they are a natural test case for VM use because these data are typically difficult or impossible to share broadly, and thus must be analyzed *in situ*.

We intended to require only minimal computing power and programming skills from users, so as to bring these systems within the reach of the general language scientist. Given the success of the LENA™, it was reasonable to build on their user base and start with child-centered recordings. We were ideally positioned to do so because we are part of a large international collaboration grant, "ACLEW: Analyzing Child Language Experiences Around The World" ("ACLEW - analyzing child language experiences around the world," n.d.). The scientific goal of this grant is to document patterns of variation and stability in young children's language experiences, and their subsequent development, as documented via daylong recordings. Additionally, our collaborator network includes some members with very limited or no previous programming experience, allowing us to beta test that instructions are clear and usable. Moreover, much research in this field employs a unified recording device and software toolkit for automatic speech processing developed by the LENA™ Foundation. While this product is not open source, it provides an interesting benchmark to compare our work against since it was specifically designed to process children's speech.

In this paper, we present our latest stable release. At the time of writing, DiViMe contains a set of algorithms which were designed to automatically detect and label speaker turns in naturalistic audio recordings, as well as a number of other tools. Two main tasks are distinguished to achieve this goal. A first category of tools perform *Voice Activity Detection* (VAD). The output of such tools is typically a file of time labels with `vocal' or`non-vocal' tags. Once the vocal stretches are located in the audio files, a second category of tools can be applied to attribute each vocalization stretch to a specific speaker. This second task is named

Talker Diarization (TD). Some of the tools in DiViMe perform both tasks jointly, receiving as input the raw recording and returning a set of talker labels with onset and offset timing.

INSERT HERE SOMETHING ABOUT OPEN SOURCE REPRODUCIBILITY REPLICABILITY CUMULATIVE SCIENCE

Description

Workflow.

Installation. The VM is designed using Vagrant ("Vagrant by hashicorp," n.d.), which is a tool enabling to build and manage virtual machine environments. The VM is completely specified through a Vagrantfile script which contains the core architecture of the computing system to be deployed. Based on this file, Vagrant runs the virtual environment on top of usual providers such as VirtualBox ("Oracle vm virtualbox," n.d.) (a Docker ("Docker - build, ship and run any app, anywhere," n.d.) version is under development). We provide a stable Vagrantfile which automatically builds and run a Ubuntu virtual machine isolated from your local hosting computer. The resulting environment runs on any local machine regardless of its operating system. It contains all required dependencies and speech processing tools introduced in this paper. The detailed instructions for installing the VM are in https://divime.readthedocs.io/.

The VM is completely isolated from the local host, but can exchange files via shared folders that are viewed by both the VL and the local host. For instance, the `data` folder enables to transfer data from the host to the VM and results from the VM back to the host. The basic workflow of the VM is summarized in the schematic diagram of Figure~**??**.

Application. Once the installation is complete, the tools that the VM provides can be applied to data files on the user's host machine with a series of simple shell commands (e.g., `vagrant ssh -c "tools/TOOLNAME data/"`). We provide users with an on-line documentation in https://divime.readthedocs.io/.[1]

[1]There is also a local version of the documentation in the shared directory `docs`. The source files are in `docs/source`, and the html in `docs/build/html/index.html`. It can be recompiled through the VM

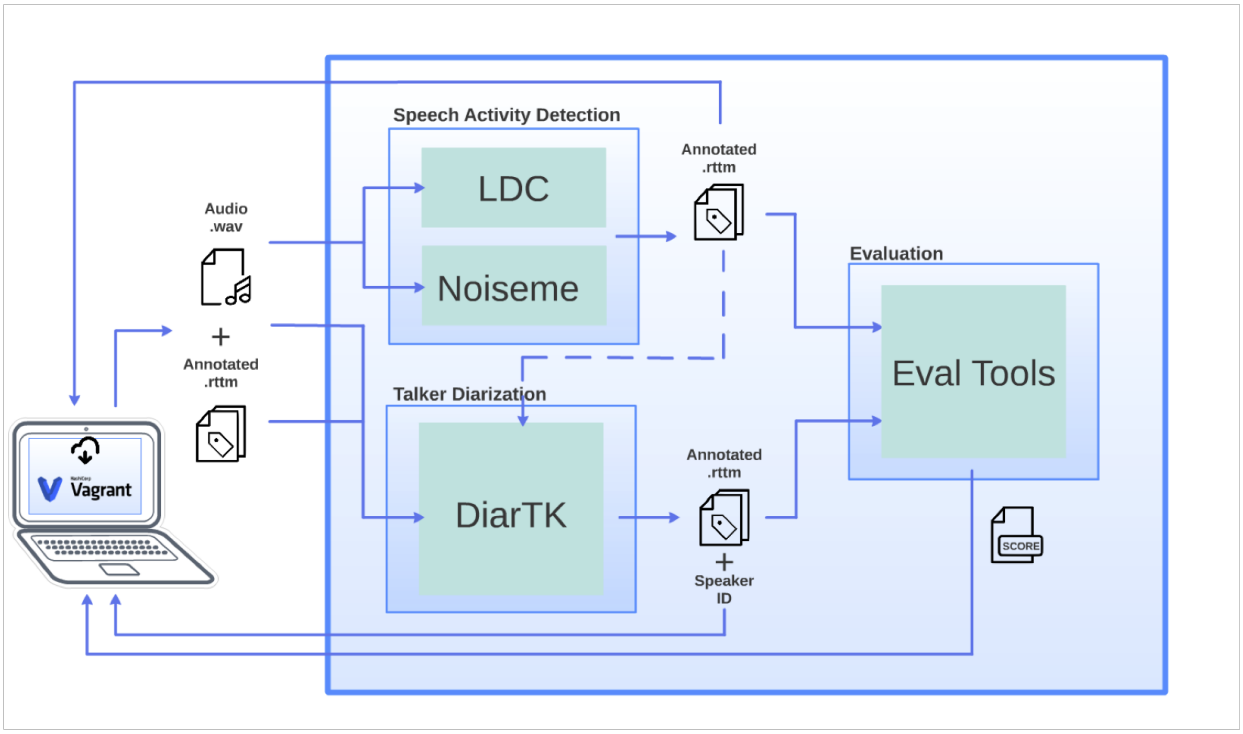

Figure 1. Schematic diagram of data flow in DiViMe. Significant inputs and outputs (as well as log files) of the individual tools are being read from and written to synced folders of the host computer. Processing is triggered via shell commands, on the host machine.

```
# wake the machine up
vagrant up
# apply VAD to all wav files in the data folder
vagrant ssh -c "tools/ldc_sad.sh data/"
# apply DiarTK on wav+LDC_SAD files
vagrant ssh -c "tools/DiarTK.sh data/ ldc_sad"
```

through `vagrant ssh -c "cd /vagrant/docs; make html"`.

```
# evaluate the above
vagrant ssh -c "tools/eval.sh data/ diartk ldc_sad
# put the machine back to sleep
vagrant halt
```

Checks. Steps taken to ensure reproducibility

Vagrant includes routines for checking that all parts are in place and working reasonably well

for instance, pulls sample data (aclew starter, vandam CITE) and runs whole pipeline, checks that results are consistent

NOTE: Some of our tools are non-deterministic; how do we decide the system passed?

Input and output files. Although some of the tools can receive formats other than short .wav files as input (.mp3, audiofiles lasting more than 10h), it is simpler for now to assume that at worst a conversion step can take place at the onset to get all input files into the same format. Audio files are expected to be in .wav format. If the user has annotations at either the speech activity or diarization levels, for simplicity we only require the RTTM ("An object oriented description of speech for EARS," n.d.) format. That is, if the user wants to evaluate the VAD performance, then he/she will need to provide the RTTM label for each wav file containing the human-annotated reference annotation. Notice that this gold RTTM can also be provided for the diarization tools, so as to assess talker diarization performance in the absence of VAD errors.

The system returns all annotations in the RTTM format, with the name of the tool that produced them appended to the original file name. Evaluations are returned in a dataframe format, with wavs as rows, and metrics as columns.

Tools in the current DiViMe release. The current DiViMe builds exclusively on tools that have been developed, documented, and made available by independent researchers. We therefore keep the descriptions very short, and instead provide links to the original resources, where readers will be able to find the full technical descriptions.

Voice Activity Detection. We currently provide four options for Voice Activity Detection (VAD) tools. The first is the *LDC SAD* (Ryant, n.d.), which relies on HTK Young et al. (2002) to band-pass filter and extract PLP features, prior to applying a broad phonetic class recognizer trained on the Buckeye Corpus Pitt, Johnson, Hume, Kiesling, and Raymond (2005) using a GMM-HMM model. An official release by the LDC is currently in the works, and should be ready by the time Interspeech is held.

Our second VAD tool will be referred to as *Noiseme SAD* because it draws from a broader `noiseme classifier'' @wang2017first, a neural network that can predict frame-level probabilities of 17 types of sound events (called`noisemes'"Burger, Jin, Schulam, and Metze (2012)), including speech, singing, engine noise, etc. The network consists of one single bidirectional LSTM layer with 400 hidden units in each direction. It was trained on 10h of HAVIC data Strassel et al. (2012) with the Theano toolkit which we will change in the future since this framework is no longer maintained. The OpenSMILE toolkit Eyben, Weninger, Gross, and Schuller (2013) is used to extract 6,669 low-level acoustic features, which are reduced to 50 dimensions with PCA. For our purposes, we summed the probabilities of the classes `speech'' and`speech non-english"' and labeled a region as speech if this probability was higher than all others.

Talker Diarization. We currently provide one Talker Diarization (TD) tool. The *DiarTK* model imported in the VM is a C++ open source toolkit Vijayasenan and Valente (2012). The algorithm first extracts MFCC features, then performs non-parametric clustering of the frames using agglomerative information bottleneck clustering Vijayasenan, Valente, and Bourlard (2007). At the end of the process, the resulting clusters correspond to identified speakers. The most likely Diarization sequence is computed by Viterbi realignement.

Speech quantification. We include one tool that provides an estimate of the number of syllables found in each turn.

MISSING REFERENCE TO PAPER BY OKKO ET AL INTRODUCING THE WCE IN TECHNICAL TERMS

Joint tools and secondary analyses. Once the talker diarization phase has been completed, the next phase of analysis will depend completely on the user base. Often, the next step must be to decide which speaker is the person who wore the recorder, and to assign typical roles to those around him/her. For the user base we are building on, this entails distinguishing the "target child" from other children and from adults, so as to be able to perform specialized analyses of language produced versus experienced by the target child. Other user bases will have other typical roles, for instance "patient" versus "caregiver". Since an adult patient and an infant have very different voices, it is impossible to build a "role assignment" tool that will work for every user base. This step can only be done cooperating with the relevant research community, so that the latter provides annotated data on which acoustic models can be trained to make the classification.

Therefore, we focus here on the classification that has already been trained and is made available within DiViMe today, namely one in which the audio is parsed into the following categories: target child (the one wearing the device), other children, female adult, male adult.

MISSING REFERENCE TO PAPER INTRODUCING THE CLASSIFIER IN TECHNICAL TERMS

Similarly, other classification tasks may be desirable for specific talker roles. DiViMe includes three such tools.

MISSING REFERENCE TO PAPER BY FRANK ET AL INTRODUCING THE ADS/CDS CLASSIFIER IN TECHNICAL TERMS

MISSING REFERENCE TO PAPER BY ZIXING ET AL INTRODUCING THE VCM IN TECHNICAL TERMS

ADD DESCRIPTION OF STATISTICS/SUMMARY TOOL

Evaluation. Finally, we have evaluation tools for each task noted above. We have tried to stay close the speech technology literature, and re-use available code for evaluation as well. This was important to us because we wanted to make sure that our results were widely interpretable and comparable against established benchmarks.

For voice activity detection, we employ code included in pyannote (**???**), which returns the false alarm (FA) rate (proportion of frames labeled as vocalizations that were not vocalizations in the gold annotation) and missed speech rate (proportion of frames labeled as non-vocalizations that were in fact vocalizations in the gold annotation). Frames are always 10 ms long.

For role and talker diarization, we report, in addition to false alarm and miss rates, the confusion rate, which establishes the ratio of frames labeled as the wrong speaker.

One important consideration is in order: What to do with files that have no speech to begin with, or where the system does not return any speech at the VAD stage or any labels at the TD stage. This is not a case that is often discussed in the literature because recordings are typically targeted at moments where there is speech. However, in naturalistic recordings, some extracts may not contain any speech activity, and thus one must adopt a coherent framework for the evaluation of such instances. Notice that since all of the ratios above have the amount of speech in the reference as denominator, all of them are undefined when there is no speech.

We opted for the following decisions. If the gold annotation was empty, and the VAD system returned no speech labels, then the FA = 0 and M = 0; but if the VAD system returned some speech labels, then FA = 100 and M = 0. Also, if the gold annotation was not empty and the system did not find any speech, then this was treated as FA = 0 and M = 100.

As for talker diarization evaluation, the same decisions were used above for FA and M, and the following decisions were made for confusion. If the gold annotation was empty, regardless of what the system returned, the mismatch rate was treated as 0. If the gold annotation was empty but a pipeline returned no TD labels (either because the VAD in that system did not detect any speech, or because the diarization failed), then this was penalized via a miss of 100 (as above), but not further penalized in terms of talker mismatch, which was set at 0.

The same DER system was applied to the evaluation of the other classification tasks,

namely talker role attribution, addressee estimation, and vocal maturity. By and large, misses and false alarms will carry over from the front-end segmentation tasks, and categorization error reflects inaccuracies of frame labeling.

Finally, there is no standard system for evaluating syllable count estimations. EXPLAIN WHAT WE DO IN THE END OR REFER TO PAPER

Experiments

We conducted several experiments to test and benchmark the tools currently included in DiViMe. To this end, we used XXX TO BE REVISED datasets, as follows.

- Tsimane (9h): A total of 537 1-minute clips were extracted from 1-2 daylong recordings gathered from 27 children learning Tsimane in rural Bolivia Scaff, Stieglitz, and Cristia (n.d.). Of these, 227 came from LENA™ recordings (henceforth Tsi-LENA), and the remaining 310 from other devices (USB or Olympus; henceforth Tsi-other). Clips were sampled periodically throughout the day to avoid sampling bias. Speakers were labeled using broad classes (children, female adults, male adults), with the exception of the child wearing the recorder and the most common female adult voice. The annotator did not know the recorded families.

Results for VAD at the time of final submission are shown on **??** and **??**; those for TD are shown on **??**. Recordings not collected with LENA™ hardware cannot be analyzed with the LENA™ software, and thus such combinations are shown as NA below.

```
mydirs=c("../evaluations/ASE+", "../evaluations/Casillas", "../evaluations/dihard", "../
alldi=NULL
allsad=NULL
for(thisdir in mydirs){
  print(thisdir)
  dir(path=thisdir, pattern="diartk")->fs
```

```
  for(tf in fs) {
    print(tf)
    thisfile=read.table(paste0(thisdir,"/",tf),sep="\t",header=T)
    thisfile$filename=rownames(thisfile)
    alldi=rbind(alldi,cbind(thisdir,tf, thisfile ) )
    }

  dir(path=thisdir, pattern="_sad")->fs #this is broken due to inconsistent naming sche
  for(tf in fs) {
    print(dim(allsad))
    print(tf)
    allsad=rbind(allsad,cbind(thisdir,tf, read.table(paste0(thisdir,"/",tf),header=T,sep

  if(thisdir %in% c("../evaluations/ASE+","../evaluations/tsimane/tsi-lena")){
    tf="lena"
    thisfile=read.table(paste0(thisdir,"/lena_diar_eval.df"),sep="\t")
    thisfile$filename=rownames(thisfile)
    alldi=rbind(alldi,cbind(thisdir,tf, thisfile ) )
    allsad=rbind(allsad,cbind(thisdir,tf, read.table(paste0(thisdir,"/lena_sad_eval.df"

  }
}
for(j in c("DCF","FA","MISS")) allsad[,j]=gsub("%","",allsad[,j],fixed=T)

allsad[,"tf"]=gsub("_sad_eval.df","",allsad[,"tf"],fixed=T)
allsad[,"tf"]=gsub("/","-",allsad[,"tf"],fixed=T)
```

```
alldi[,"tf"]=gsub("sad_eval.df","",alldi[,"tf"],fixed=T)
alldi[,"tf"]=gsub("Sad_eval.df","",alldi[,"tf"],fixed=T)
alldi[,"tf"]=gsub("diartk_","",alldi[,"tf"],fixed=T)
alldi[,"tf"]=gsub("_","",alldi[,"tf"],fixed=T)

write.table(alldi,"alldi.txt",row.names=F,sep="\t",quote=T)
write.table(allsad,"allsad.txt",row.names=F,sep="\t",quote=T)
```

```
##                  thisdir                    tf          DER
##  ../evaluations/ASE+:18   diartk_opensmile:18   Min.   :  70
##                                                 1st Qu.: 109
##                                                 Median : 160
##                                                 Mean   : 246
##                                                 3rd Qu.: 222
##                                                 Max.   :1394
##   B3Precision      B3Recall         B3F1        TauRefSys
##  Min.   :0.18   Min.   :0.24   Min.   :0.22   Min.   :0.016
##  1st Qu.:0.47   1st Qu.:0.27   1st Qu.:0.35   1st Qu.:0.045
##  Median :0.53   Median :0.30   Median :0.37   Median :0.070
##  Mean   :0.51   Mean   :0.31   Mean   :0.38   Mean   :0.081
##  3rd Qu.:0.56   3rd Qu.:0.34   3rd Qu.:0.42   3rd Qu.:0.118
##  Max.   :0.82   Max.   :0.42   Max.   :0.48   Max.   :0.161
##    TauSysRef          CE             MI            NMI
##  Min.   :0.03   Min.   :0.51   Min.   :0.06   Min.   :0.040
##  1st Qu.:0.08   1st Qu.:1.16   1st Qu.:0.15   1st Qu.:0.103
##  Median :0.16   Median :1.22   Median :0.27   Median :0.137
##  Mean   :0.16   Mean   :1.39   Mean   :0.30   Mean   :0.144
```

```
##  3rd Qu.:0.22   3rd Qu.:1.49   3rd Qu.:0.42   3rd Qu.:0.179
##  Max.   :0.38   Max.   :3.08   Max.   :0.61   Max.   :0.312

##                thisdir           tf          DER        B3Precision
##  ../evaluations/ASE+:90   gold     :18   Min.   :  54   Min.   :0.16
##                           ldc      :18   1st Qu.:  93   1st Qu.:0.41
##                           noisemes :18   Median : 101   Median :0.53
##                           opensmile:18   Mean   : 161   Mean   :0.51
##                           tocombo  :18   3rd Qu.: 153   3rd Qu.:0.60
##                                          Max.   :1394   Max.   :0.82
##                                                         NA's   :1
##     B3Recall         B3F1        TauRefSys      TauSysRef          CE
##  Min.   :0.24   Min.   :0.22   Min.   :0.02   Min.   :0.01   Min.   :0.5
##  1st Qu.:0.31   1st Qu.:0.37   1st Qu.:0.06   1st Qu.:0.08   1st Qu.:1.1
##  Median :0.42   Median :0.45   Median :0.09   Median :0.13   Median :1.3
##  Mean   :0.48   Mean   :0.47   Mean   :0.10   Mean   :0.15   Mean   :1.4
##  3rd Qu.:0.63   3rd Qu.:0.55   3rd Qu.:0.13   3rd Qu.:0.20   3rd Qu.:1.6
##  Max.   :0.92   Max.   :0.84   Max.   :0.27   Max.   :0.45   Max.   :3.3
##  NA's   :1      NA's   :1      NA's   :1      NA's   :1      NA's   :1
##        MI             NMI                          filename
##  Min.   :0.04   Min.   :0.04   BER_0485_12_07_09123: 5
##  1st Qu.:0.14   1st Qu.:0.11   BER_0713_07_02_21041: 5
##  Median :0.25   Median :0.14   BER_2224_16_11_01293: 5
##  Mean   :0.27   Mean   :0.15   CAS_0643_02_01_25764: 5
##  3rd Qu.:0.35   3rd Qu.:0.17   CAS_2625_32_01_11145: 5
##  Max.   :0.65   Max.   :0.36   CAS_8787_11_01_14005: 5
##  NA's   :1      NA's   :1      (Other)              :60
```

tabsadFA False alarm (FA) rates in VAD as a function of the dataset and the VAD

tool. Lower is better.

tabVADM Miss (M) rates in VAD as a function of the dataset and the VAD tool. Lower is better.

tabdia:

DER in TD as a function of the dataset. The LENA column indicates diarization performance for the LENA algorithm as a whole. For all other columns, diarization was done with DiarTK, and the column label indicates the VAD annotation used as input. Gold column gives the results of applying DiarTK to the human annotated VAD. The DiHARD results are as provided by the Challenge organizers. Lower is better.

Experiment 1: How well do included tools fare against the current field standards?}. LENA has data for VAD, TD, WCE, VCM

VAD and TD. The LENA™ software performs joint segmentation and classification with acoustic models developed on the basis of an open source ASR toolkit in addition to being trained with 150 hours of hand-annotated data from English-learning American children growing up in urban settings. It returns a segmentation of the audio into categories: key child, other children, female adult, male adult, TV noise, other noise, silence, and overlap (which is overlap between any of the non-silence categories). For the purposes of our experiments, we declared as non-speech all the non-human categories as well as the speech categories that the system classified as far from their acoustic models, because in pilot analyses the VAD performance was better without than with these far items.

To focus on differences that were stable rather than averages like the ones reported on the Tables above, we fit a mixed regression model (in R Team and others (2013)}, package lme4 D. Bates, Maechler, Bolker, Walker, and others (2014)}), declaring corpus, system, and their interaction as fixed effects and the clip ID as random effect. Given the question addressed in this experiment, we focus on the two corpora gathered with a LENA™ device. We declared ASE+ as the baseline for corpus (since it is closer to what the LENA™ system was developed on), and LENA™ as the baseline for system. Results are shown on Table **??**.

Effects of corpus will be discussed in the next subsection. Turning to the current key interest, LDC SAD led to a significantly higher FA and lower Miss rates than LENA^{TM} , whereas Noisemes led to a non-significantly lower FA and higher Miss rates. Given that the reduction in Miss is smaller than the gain in FA with the LDC SAD system, this appears like a competitive alternative to the LENA^{TM} system, as does Noisemes which performed no better or no worse than LENA^{TM}. The results of the DER analyses, which compound errors over the VAD and TD phases, confirm these conclusions, as neither of our systems differed from the LENA^{TM} significantly for ASE+, and there was only an interaction between LDC and corpus at t = 2.1.

INSERT TABLE HERE

tab:reg1

Mixed model regressions predicting performance from the system, corpus, and their interaction. Each cell shows the estimate (and its standard error) corresponding to the crossing of the predictor and the dependent variable. An asterisk indicates an effect with t > 2.

WCE. to add

VCM. to add

Experiment 2: How well do tools do with audio collected with other devices and untrained populations? Should the flow go like this or shall we change the overall structure of the paper?

Language acquisition researchers are often put off by the cost of the LENA^{TM} devices and software (about 13k US$ for 2 devices and the PRO version of the software). However, perhaps this is worthwhile. Indeed, it may be the case that recordings carried out using other recording devices than the LENA^{TM} hardware lead to better automatic annotations than daylong recordings gathered using other hardware.

The main effect of corpus in Table **??** shows a significantly higher FA and significantly lower M for Tsi-lena than ASE+, due to the fact that there was a great deal more silence in

the former files (in fact, nearly half of the Tsimane clips had no speech in them). This effect is caused by the Tsimane clips being randomly sampled throughout the day and night, whereas the ACLEW Starter set clips were selected because there was speech in them.

To provide a broader picture, we fit another mixed model predicting DER (which represents global performance for a given pipeline), this time with the 4 child corpora. As before, fixed effects were corpus, method, and their interaction, with baseline levels ASE+ and LENA^{TM}. Only two of the interactions (LDCxTsi-LENA, and NoisemesxTsi-LENA, indicating lower DERs in this corpus when our systems were used rather than the LENA system) had t$>$2, suggesting that all systems performed similarly to each other and across corpora. As for the impact of the hardware, the results of Tsi-LENA and Tsi-other (recorded with non-LENA devices) do not highlight better performances on Tables **??**, **??**, **??** when using the LENA hardware.

Experiment 3: Benchmarking against the extant challenges. I like the idea of having a final benchmark section – others?

VAD and TD. We had two goals by using the DiHARD Challenge data. First, the performance of the same tools across our child language acquisition data versus the DiHARD data indirectly speaks to how comparably difficult our datasets are. The DiHARD test data contains an heterogeneous mix of data, whereas all of the other datasets we tested here are children-centered, collected in a completely ecological fashion. We observe that DER is higher for the non-DiHARD datasets than the DiHARD Challenge dataset, regardless of the tool.

Second, we can compare the tools in DiViMe against the leaderboard of the Challenge on the DiHARD data so as to assess to what extent our tools are competitive. Our primary purpose was to offer a quick and easy access to speech processing tools to conduct research. Therefore, we did not expect the tools we introduced so far to outperform the state-of-the-art of VAD and TD. This expectation was confirmed: Our systems score at the bottom of the DiHARD chart for both tracks. This is not only the case due to our VAD

being underperforming, as clear from the fact that TD with gold VAD still led to a very high error rate. However, we did not retrain our tools on our testing datasets to reflect an "out of the box" use of the VM. While we feel that DiViMe fits its function in terms of usability, we look forward to incorporating better-performing VAD and TD tools in the future.

Other tools. They are state of the art and thus not evaluated here.

Conclusions

We presented a Virtual Machine that almost anybody can use to detect speech segments using various advanced techniques. We outlined the VM's use, its internals, and provided pointers to currently available algorithms. Our benchmarks showed that ecological language acquisition data are particularly hard even when compared with the DiHARD Challenge data. We would look forward to integrating better-performing VAD and TD systems. In next steps, we will incorporate models that can be retrained inside the VM. In the meanwhile, for several tasks and dataset combinations, we remain competitive against the LENA^{TM}, which is the current go-to system in the language acquisition field, making DiViMe a competitive open-source solution for this audience. Additionally, the algorithms currently included are robust to variation in the recording hardware used and the population from which data are collected, which are crucial features for our target users. In sum, DiViMe is a promising tool that makes complex processing models accessible to non-technical users.

ACLEW - analyzing child language experiences around the world. (n.d.). https://sites.google.com/view/aclewdid/home.

An object oriented description of speech for EARS. (n.d.). https://catalog.ldc.upenn.edu/docs/LDC2004T12/RTTM-format-v13.pdf.

Bates, D., Maechler, M., Bolker, B., Walker, S., & others. (2014). Lme4: Linear mixed-effects models using eigen and s4. *R Package Version*, *1*(7), 1–23.

Burger, S., Jin, Q., Schulam, P. F., & Metze, F. (2012). *Noisemes: Manual annotation*

of environmental noise in audio streams (No. CMU-LTI-12-07). Pittsburgh, PA; U.S.A.: Carnegie Mellon University.

Docker - build, ship and run any app, anywhere. (n.d.). https://www.docker.com/.

Eyben, F., Weninger, F., Gross, F., & Schuller, B. (2013). Recent developments in opensmile, the munich open-source multimedia feature extractor. In *Proceedings of the 21st acm international conference on multimedia* (pp. 835–838). ACM.

Oracle vm virtualbox. (n.d.). https://www.virtualbox.org/.

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.

Plummer, A., Riebling, E., Kumar, A., Metze, F., Fosler-Lussier, E., & Bates, R. (2014). The speech recognition virtual kitchen: Launch party. In *Proc. interspeech*. Singapore: ISCA.

Ryant, N. (n.d.). LDC sad. https://github.com/Linguistic-Data-Consortium.

Scaff, C., Stieglitz, J., & Cristia, A. (n.d.). Daylong recordings from young children learning Tsimane in Bolivia. https://nyu.databrary.org/volume/445.

Strassel, S., Morris, A., Fiscus, J. G., Caruso, C., Lee, H., Over, P. D., ... Michel, M. (2012). Creating havic: Heterogeneous audio visual internet collection. In *Proc. lrec*. Istanbul, Turkey: ELRA.

Team, R. C., & others. (2013). R: A language and environment for statistical computing.

The first DIHARD speech diarization challenge. (n.d.). https://coml.lscp.ens.fr/dihard/index.html.

The speech recognition virtual kitchen. (n.d.). https://github.com/srvk.

Vagrant by hashicorp. (n.d.). https://www.vagrantup.com/.

Vijayasenan, D., & Valente, F. (2012). Diartk: An open source toolkit for research in multistream speaker diarization and its application to meetings recordings. In *Thirteenth*

annual conference of the international speech communication association.

Vijayasenan, D., Valente, F., & Bourlard, H. (2007). Agglomerative information bottleneck for speaker diarization of meetings data. In *Automatic speech recognition & understanding, 2007. asru. ieee workshop on* (pp. 250–255). IEEE.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., ... others. (2002). The htk book. *Cambridge University Engineering Department*, *3*, 175.