Skeleton of analyses

Contents

Conceptual explanation

This project aims at establishing the reliability of measures extracted from long-form recordings. Reliability here means several things:

1. Inter-rater reliability, comparing (a) one main human annotator against two other annotators who only saw a small portion of the data, and (b) the two other annotators against each other. These are calculated only on the data that the two other annotators coded: 79 1-minute chunks. This tells us how much trust we can place on the human annotations.
2. Machine-human reliability, comparing an auto-annotator system (VTC) output against the main human annotator. This is calculated only on the data that the main human annotators coded: 89 recordings x 1-minute chunks x number of hours in each recording. This tells us how much trust we can place on the auto annotations relative to the human annotations.
3. Cross-day reliability using the main human annotator data, comparing estimates derived from two days drawn at random[1]. See section on derived estimates for full list. This is calculated only on the data that the main human annotators coded: 89 recordings x 1-minute chunks x number of hours in each recording. This tells us how much trust we can place on our derived estimates if we only had one recording for a given child/visit, and about as much coding as that human annotator did.
4. Cross-day reliability using the auto annotator system output, comparing estimates derived from two days drawn at random. This is calculated on all data: 89 recordings.[2] This tells us how much trust we

[1] In the 2018 analyses, odd and even days were used, but this is not ideal for two reasons: 1. We want to help readers learn how stable measurements can be if they only draw one recording for a given child, so our measures based on multiple odd days may provide an overestimate of that; 2. Different quantity of data go into such an analysis depending on the child (so it's messy).

[2] Lucas recently cleaned these files, and I think he found extra days that had not been human-annotated, so there may be more days. Also, Lucas was finishing the VTC on 6 children who were missing from Paul Valois' folder.

can place on our derived estimates if we only had one recording for a given child/visit.

Missing from the above is some form of external validation: We would ideally want to see that measures that are highly reliable also explain meaningful individual variance. One option in these data is to use longitudinal prediction of children's outcomes – but this may be more akin to long-term stability than true prediction. For now, therefore, this document does *not* cover convergent/divergent validity.

How reliability is calculated

Inter-rater reliability and machine-human reliability can be evaluated based on diarization success: to what extent the methods converge in stating that one "frame" (a 10 ms slice of the recording) contains a given speaker. We have a routine for evaluating this in the team, but it has proved fragile, because it had to be adapted across different projects. For this particular project, they have been calculated by an intern (Paul Valois), who created the subfolder `valois_reliability` (formerly called `vtc_lena`). In a nutshell, the process for this is:

- preprocessing: human and machine annotation get converted into a common format
- evaluation: speaker types get homogeneized, confusion matrix and diarization evaluation statistics are computed

Lucas is currently fixing the preprocessing step so that it's robust to different formats of data (which will contain speaker type homogeneization, since conceptually it belongs to preprocessing and not evaluation). Afterwards, Nicolas will be able to use this re-run the reliability evaluation.

Cross-day reliability cannot be analyzed in the same way because it's not two annotations of the same day, but annotations for two different days. So we need to use a method that looks at the stability of derived metrics (see next section). For now, this code has been written by Alex and is in this document. However, Nicolas may think of better ways of doing this than simple correlations.

Derived metrics

For cross-day reliability, we want to have measures that represent children's language development and their language input (ie the experiences from which the child can learn and that are provided by others). For now, we are calculating all combinations of 1. how vocalization events are "integrated"; 2. their sources; and 3. their context.

How vocalization events are "integrated":

- vocalization counts (ie how many vocalization events)
- vocalization duration (ie total time vocalized)

Sources of vocalizations:

- key child (CHI)
- female adults (FEM)
- male adults (MAL)
- all adults (summing FEM and MAL, ADU)
- other children (OCH)

Context of vocalizations:

- unmarked (ie all vocalizations regardless of the context)
- for non-CHI child-directed speech or "cds" (only vocalizations that occur in temporal proximity of the key child's vocalizations)
- for the child's vocalizations, that occur in temporal proximity of someone else's vocalizations – i.e. in "conversations"

Finally, some authors have popularized the notion of "turns", counting the number of times an adult and a child vocalize in close temporal proximity. These measures tend to have terrible machine-human reliability,

but for comparability with previous work (and since our auto algorithm is much better than that used in previous work), we have also extracted turns.

This is the full current list of derived metrics:

- counts (unmarked): vc_chi,vc_fem,vc_mal,vc_adu,vc_och,
- duration (unmarked): voc_dur_chi,voc_dur_fem,voc_dur_mal,voc_dur_adu,voc_dur_och,
- cds: voc_dur_och_cds,voc_dur_fem_cds,voc_dur_mal_cds,voc_dur_adu_cds,
- conv(ersations): voc_dur_chi_conv,
- turns: turns_chi,turns_och,turns_fem,turns_mal,turns_adu

Notes:

1. We could have extracted N of vocalizations for cds and conv types, and this should be easy to do. To this end, one would need to modify `extract_quantities.R`.
2. We currently extract sums over the whole file and then compute hourly averages – but it is unclear that this is optimal. In the literature, some people use instead 5-minute block estimates, and then extract the top 3 per child – but Alex suspects this is just because they then do human annotation. Nonetheless, the case could be made that a better unit takes into account internal structure in the day, at least using hourly blocks, and eg removing outlier hourly blocks, or analyzing the data as a time series.

Plans

Project advancement is noted on this github projet, but for clarity Alex lists all issues to do or in progress in this document, so that it's clear what they are for and when they become relevant.

Data generation

This goes from raw annotation data to data frames we can use for analyses.

Convert main human data (textgrid) into common format

- Waiting for Lucas to finish the routine for standardizing data
- then Nicolas uses that routine to standardize the main human annotator's data

Convert two other human data (csv) into common format

- Waiting for Lucas to finish the routine for standardizing data
- then Nicolas uses that routine to standardize the two other human annotator's data

Convert auto-annotator data (rttm) into common format

- Waiting for Lucas to finish the routine for standardizing data
- Waiting for Lucas to finish the analyses of 6 babies who were missing VTC
- then Nicolas uses that routine to standardize all rttms

Extract derived metrics.

This is only necessary for goals 3 & 4 (cross-day reliability using the main human/auto output). This is fully written (see below), but it may need to be adapted to the new data format that Lucas will create. However,

for now, everything works, and improvements can already be generated since they'll apply to the new data as it gets added.

Preprocessing

Just data shuffling here – this does not need to be re-done, but do revisit this when (a) data are standardized; (b) we get the 6 babies who were missing VTC

```
myfolder="/Users/acristia/Dropbox/its2pops/_nam_stab_chi/vtc_lena/output/"

#only needs to be redone when Lucas has finished VTCing the other kids
#break up single rttm per child into one per file
for(subfolder in dir(myfolder)) if(subfolder!="extracted_pa") for(thisf in dir(paste0(myfolder, subfolde
  read.table(paste0(myfolder,"/",subfolder,"/",thisf))->thisrttm
  for(thisrec in levels(thisrttm$V2)){
    write.table(thisrttm[thisrttm$V2==thisrec,],file=paste0("one_rttm_xrec/",thisrec,".rttm"),quote=F,r
  }
}
```

Get all stats at the level of the recording

The code pillages from Christof Neumann's Uruguay analyses and his avutils package.

```
source("extract_quantities.R") #load functions
source("read_rttm.R")


mydat=NULL

for(thisrec in dir("one_rttm_xrec/",pattern=".rttm")){
    fp_sad=paste0("one_rttm_xrec/",thisrec)

   # voc counts, voc durations, turns
  temp <- sapply(fp_sad,
                 extract_quantities,
                 turntakingthresh = 0.3,
                 simplify = FALSE)

   # voc counts
  temp1 <- unlist(lapply(temp, function(x)x$voc_count))
  vc_chi <- sum(temp1[grep(".KCHI", names(temp1),fixed=T)])
  vc_och <- sum(temp1[grep(".CHI", names(temp1),fixed=T)])
  vc_fem <- sum(temp1[grep(".FEM", names(temp1),fixed=T)])
  vc_mal <- sum(temp1[grep(".MAL", names(temp1),fixed=T)])
  vc_adu <- vc_fem + vc_mal
  rm(temp1)

  # voc durations
  temp1 <- unlist(lapply(temp, function(x)x$cum_dur))
  voc_dur_chi <- sum(temp1[grep(".KCHI", names(temp1),fixed=T)])
  voc_dur_och <- sum(temp1[grep(".CHI", names(temp1),fixed=T)])
  voc_dur_fem <- sum(temp1[grep(".FEM", names(temp1),fixed=T)])
  voc_dur_mal <- sum(temp1[grep(".MAL", names(temp1),fixed=T)])
```

```
  voc_dur_adu <- voc_dur_fem + voc_dur_mal
  if (voc_dur_fem == 0 | is.na(voc_dur_fem)) voc_dur_fem <- NA
  if (voc_dur_mal == 0 | is.na(voc_dur_mal)) voc_dur_mal <- NA
  if (voc_dur_chi == 0 | is.na(voc_dur_chi)) voc_dur_chi <- NA
  if (voc_dur_och == 0 | is.na(voc_dur_och)) voc_dur_och <- NA
  rm(temp1)

  #CDS
   temp1 <- unlist(lapply(temp, function(x)x$cds_dur))
  voc_dur_chi_conv <- sum(temp1[grep(".KCHI", names(temp1),fixed=T)])
  voc_dur_och_cds <- sum(temp1[grep(".CHI", names(temp1),fixed=T)])
  voc_dur_fem_cds <- sum(temp1[grep(".FEM", names(temp1),fixed=T)])
  voc_dur_mal_cds <- sum(temp1[grep(".MAL", names(temp1),fixed=T)])
  voc_dur_adu_cds <- voc_dur_fem_cds + voc_dur_mal_cds
  if (voc_dur_fem_cds == 0 | is.na(voc_dur_fem_cds)) voc_dur_fem_cds <- NA
  if (voc_dur_mal_cds == 0 | is.na(voc_dur_mal_cds)) voc_dur_mal_cds <- NA
  if (voc_dur_chi_conv == 0 | is.na(voc_dur_chi_conv)) voc_dur_chi_cds <- NA
  if (voc_dur_och_cds == 0 | is.na(voc_dur_och_cds)) voc_dur_och_cds <- NA
  rm(temp1)


  # turns
   temp1 <- unlist(lapply(temp, function(x)x$turns))
  turns_chi <- sum(temp1[grep(".KCHI", names(temp1),fixed=T)])
  turns_och <- sum(temp1[grep(".CHI", names(temp1),fixed=T)])
  turns_fem <- sum(temp1[grep(".FEM", names(temp1),fixed=T)])
  turns_mal <- sum(temp1[grep(".MAL", names(temp1),fixed=T)])
  turns_adu <- turns_fem + turns_mal
  if (turns_chi == 0 | is.na(turns_chi)) turns_chi <- NA
  if (turns_och == 0 | is.na(turns_och)) turns_och <- NA
  if (turns_fem == 0 | is.na(turns_fem)) turns_fem <- NA
  if (turns_mal == 0 | is.na(turns_mal)) turns_mal <- NA
  if (turns_adu == 0 | is.na(turns_adu)) turns_adu <- NA
  rm(temp1)

  rm(temp)



 #add data for this rec
  mydat=rbind(mydat,
              cbind(thisrec,vc_chi,vc_fem,vc_mal,vc_adu,vc_och, #counts
                    voc_dur_chi,voc_dur_fem,voc_dur_mal,voc_dur_adu,voc_dur_och, #dur
                    voc_dur_chi_conv,voc_dur_och_cds,voc_dur_fem_cds,voc_dur_mal_cds,voc_dur_adu_cds, #
                    turns_chi,turns_och,turns_fem,turns_mal,turns_adu))

  }

write.table(mydat,"data/data_visit_vtc.txt",row.names=F,quote=F,sep="\t")
```

Get all stats per hour (improvement – TODO)

Exactly as above, but instead of summing over the whole day, sum each recording hour separately.

Additional improvements to the metrics

See https://github.com/LAAC-LSCP/ChildRecordsData/issues/32

- take as input not (only) rttm's but (also) LG's new annotation and metadata file format: explained above
- provide option for dealing with overlapping speech: right now, speech gets summed even when you have two talkers or three overlapping with each other. This assumes babies can process overlapping speech perfectly (unlikely!!) and it yields to weird things when we estimate averages per hour (eg you could conceptually end up with more than one hour of speech per clock hour -silly!) – when you get to this point, we can talk about ways fo fixing this
- add metadata with start time: I haven't looked at the latest version of the metadata but I think there is a field which is at what time the recording was started
- add column with calculated local time: at the beg of the recording, this will be start time, and then you start adding time to it, so that this column places each vocalization on the actual daily time line
- derive stats for daytime or specific time range: instead of getting stats per hour of the audio time, this would mean getting stats per hour of the day

Inter-rater reliability

main human annotator against two other annotators

I think Paul Valois did not do this, but he did the next item, which is essentially the same. So check out that one and then come back here to adapt the code.

the two other annotators against each other

Paul Valois did this: see `valois_reliability/evaluations_ak` and `valois_reliability/evaluations_mm`. However, we need to re-do it with a more transparent and robust pipeline.

We are waiting for the data to be standardized (see above) to re-do it.

Once that is done, we need to adapt the following by correcting all paths and checking that we have correctly created an environment where pyannote is activated – but otherwise the code should work:

```
# File of interest
gold_data="./rttm_annotations_pa"
system_data="./rttm_annotations_mm"
other_system_data="./rttm_annotations_ak"

# 1) Create appropriate mapping ## NOTE 2020-10-17 COMMENTED THIS OUT BECAUSE LUCAS' STANDARDIZATION OF
#python scripts/labels_mapper.py -p $gold_data -m gold
#python scripts/labels_mapper.py -p $system_data -m gold
#python scripts/labels_mapper.py -p $other_system_data -m gold

# 2) Running evaluation metrics
python scripts/compute_metrics.py -ref $gold_data/mapped_gold \
-hyp $system_data/mapped_gold -t identification \
-m ider precision recall

rm -rf evaluations_mm
mkdir evaluations_mm
mv $gold_data/mapped_gold/gold_gold evaluations_mm
```

```
python scripts/compute_metrics.py -ref $gold_data/mapped_gold \
-hyp $other_system_data/mapped_gold -t identification \
-m ider precision recall

rm -rf evaluations_ak
mkdir evaluations_ak
mv $gold_data/mapped_gold/gold_gold evaluations_ak

# 3) Confusion matrices

python scripts/frame_cutter.py --i $gold_data/mapped_gold/ --o framed_gold
python scripts/frame_cutter.py --i $system_data/mapped_gold/ --o framed_system
pythin scripts/frame_cutter.py --i $other_system_data/mapped_gold/ --o framed_other_system

Rscript scripts/conf_mat.r $system_data/framed_system $gold_data/framed_gold corpora
mv $gold_data/framed_gold/*.txt evaluations_mm

Rscript scripts/conf_mat.r $other_system_data/framed_other_system $gold_data/framed_gold corpora
mv $gold_data/framed_gold/*.txt evaluations_ak
```

Machine-human reliability

Paul Valois did this: see `valois_reliability/evaluations`. However, we need to re-do it 1. with a more transparent and robust pipeline, 2. with all the babies (Paul only had 6 VTC'd).

We are waiting for the data to be standardized (see above) & for the missing 6 VTC babies to re-do it.

Once that is done, we need to adapt the following by correcting all paths and checking that we have correctly created an environment where pyannote is activated – but otherwise the code should work:

```
rm -rf data/reliability/*/rttm

#NOTE!! This section is here because the human annotations are currently incorrectly time-stamped (they
# if it doesn't go, I notice it says "eaf" which is a different format of human annotation, so this bit
# Extract rttm chunks from eaf
indices=(1 2)
for index in ${indices[*]}; do
    mkdir -p data/reliability/gold${index}/rttm
    for eaf in data/reliability/gold${index}/eaf/*.eaf; do
        eaf=$(basename $eaf)
        python scripts/adjust_timestamps.py data/reliability/gold${index}/eaf/${eaf} data/reliability/g
        ret=$?
        if [ $ret == 1 ]; then
            exit
        fi;
    done;
done;

# NOTE: look inside this script -- this may also be preprocessing rendered unnecessary by Lucas' new st
# Prepare reliability
python scripts/prepare_reliability.py

mkdir -p data/reliability/gold2/match
```

```
for rttm in data/reliability/gold2/rttm/*.rttm; do
    # Delete rely_ from the filename, so that pyannote can do the pairing
    new_path=${rttm/rttm\//match\/}
    new_path=${new_path/rely_/}

    cp $rttm $new_path
done


############################
##   1) pyannote metrics  ##
############################
source activate lena_eval

python scripts/labels_mapper.py -p data/reliability/gold1/match -m gold -o
python scripts/labels_mapper.py -p data/reliability/gold2/match -m gold -o

python scripts/compute_metrics.py -ref data/reliability/gold1/match/mapped_gold \
    -hyp data/reliability/gold2/match/mapped_gold -t identification \
    -m ider precision recall
python scripts/compute_metrics.py -ref data/reliability/gold1/match/mapped_gold \
    -hyp data/reliability/gold2/match/mapped_gold -t detection\
     -m deter precision recall --class_to_keep OCH
python scripts/compute_metrics.py -ref data/reliability/gold1/match/mapped_gold \
    -hyp data/reliability/gold2/match/mapped_gold -t detection\
     -m deter precision recall precision recall --class_to_keep CHI
python scripts/compute_metrics.py -ref data/reliability/gold1/match/mapped_gold \
    -hyp data/reliability/gold2/match/mapped_gold -t detection\
     -m deter precision recall --class_to_keep MAL
python scripts/compute_metrics.py -ref data/reliability/gold1/match/mapped_gold \
    -hyp data/reliability/gold2/match/mapped_gold -t detection\
     -m deter precision recall --class_to_keep FEM
python scripts/compute_metrics.py -ref data/reliability/gold1/match/mapped_gold \
    -hyp data/reliability/gold2/match/mapped_gold -t detection\
     -m deter precision recall --class_to_keep ELE
python scripts/compute_metrics.py -ref data/reliability/gold1/match/mapped_gold \
    -hyp data/reliability/gold2/match/mapped_gold -t detection\
     -m deter precision recall --class_to_keep OVL

rm -rf reliability_evaluations
mkdir reliability_evaluations

mv data/reliability/gold1/match/mapped_gold/gold_gold reliability_evaluations/metrics

###################################
##  3) Confusion matrices        ##
###################################

python scripts/frame_cutter.py  --i data/reliability/gold1/match/mapped_gold --o framed_mapped -d 60
python scripts/frame_cutter.py  --i data/reliability/gold2/match/mapped_gold --o framed_mapped -d 60
Rscript scripts/conf_mat.r data/reliability/gold1/framed_mapped data/reliability/gold2/framed_mapped rel
mv data/reliability/gold2/framed_mapped/*.txt reliability_evaluations
```

Cross-day reliability using the main human annotator data

This has not been done, but it's the same code as in the next section.

Cross-day reliability using the auto annotator system output

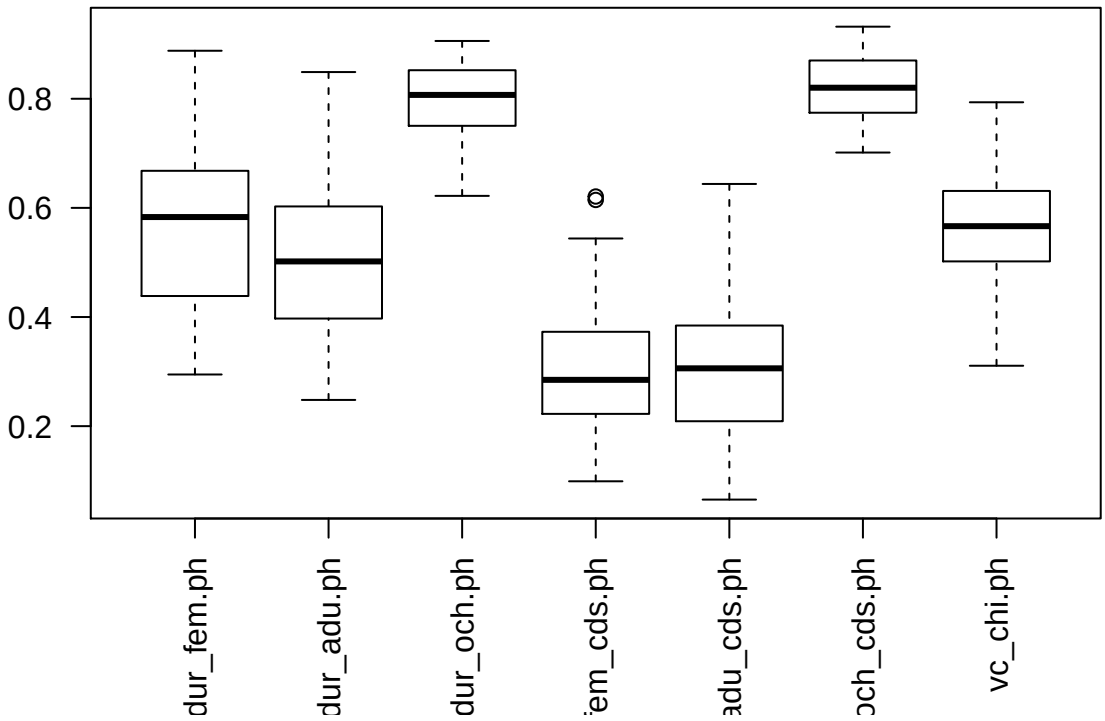
This has been done, but we want to re-do it when the annotation has been standardized (as per above).

Also, right now, Alex got file duration from VTC output – it would be more appropriate to get file duration from the actual wav duration, since silence isn't logged by VTC, so all the final silence in a recording is not being counted now.

```
# variables needed for ana:
# - total dur input
# - total dur CDS input
# - count output
selvars=c("voc_dur_fem.ph","voc_dur_adu.ph","voc_dur_och.ph","voc_dur_fem_cds.ph","voc_dur_adu_cds.ph",



# do_day_sampling
mysample<-function(myvar) sample(myvar,2)
nsamples=50
cortab=matrix(NA,nrow=nsamples,ncol=length(selvars))
colnames(cortab)<-selvars
for(i in 1:nsamples){
  twodays = aggregate(vtc$chidate,by=list(vtc$chivisit),mysample)
for(thisvar in selvars) cortab[i,thisvar]<-cor.test(vtc[vtc$chidate %in% twodays$x[,1],thisvar], vtc[vt
}

summary(cortab)
```

```
##  voc_dur_fem.ph   voc_dur_adu.ph   voc_dur_och.ph   voc_dur_fem_cds.ph
##  Min.   :0.2947   Min.   :0.2480   Min.   :0.6220   Min.   :0.09891
##  1st Qu.:0.4386   1st Qu.:0.3981   1st Qu.:0.7516   1st Qu.:0.22667
##  Median :0.5832   Median :0.5019   Median :0.8070   Median :0.28484
##  Mean   :0.5541   Mean   :0.5034   Mean   :0.7936   Mean   :0.30876
##  3rd Qu.:0.6672   3rd Qu.:0.6020   3rd Qu.:0.8523   3rd Qu.:0.37188
##  Max.   :0.8880   Max.   :0.8489   Max.   :0.9059   Max.   :0.62086
##  voc_dur_adu_cds.ph voc_dur_och_cds.ph   vc_chi.ph
##  Min.   :0.06542    Min.   :0.7014     Min.   :0.3107
##  1st Qu.:0.21024    1st Qu.:0.7753     1st Qu.:0.5026
##  Median :0.30598    Median :0.8202     Median :0.5665
##  Mean   :0.31006    Mean   :0.8202     Mean   :0.5569
##  3rd Qu.:0.38134    3rd Qu.:0.8691     3rd Qu.:0.6306
##  Max.   :0.64386    Max.   :0.9321     Max.   :0.7936
```

```
boxplot(cortab,las=2)
```


0.8
0.6
0.4
0.2
dur_fem.ph
dur_adu.ph
dur_och.ph
em_cds.ph
adu_cds.ph
och_cds.ph
vc_chi.ph