# Full Description of An Automated Pipeline for Providing Personalized Feedback Based on Audio Samples

*Loann Peurey*[1,*], *William Havard*[1,2,*], *Gwendal Virlet*[1,2,3,*], *Xuan-Nga Cao*[1,2], *Juanita Bloomfield Lescarboura*[4], *Ana Balsa*[4], *Alejandro Cid*[4], *Martín Ottavianelli*[5], *José Luis Horta Brasil*[5], *Alejandrina Cristia*[1]

[1]Laboratoire de Sciences Cognitives et de Psycholinguistique, Département d'études cognitives, ENS, EHESS, CNRS, PSL University, France
[2]Cognitive Marchine Learning Team, INRIA, France
[3]PEGASE, INRAE, Institut Agro, France
[4]Universidad de Montevideo, Uruguay
[5]Simpletech, Uruguay

loannpeurey@gmail.com, alecristia@gmail.com

## Abstract

Personalized feedback based on the automated analysis of audio samples could be useful in a wide range of intervention contexts, from early childhood to neurodegenerative programs, which target behaviors having vocal correlates. In this paper, we describe an automated pipeline that allows one to provide personalized feedback based on the automated analysis of audio samples of caregiver-child conversations captured using a smartphone. The pipeline relies on open-source packages and AWS in order to provide a cheap, reproducible, and considerably scalable solution for researchers and practitioners interested in early childhood development and caregiver-child interaction, and which could be adapted for other use cases. It processes conversation files that are 1-10 minutes long, with a cost of 0.20 US$ per hour of audio analyzed. It is currently operational in one large-scale experiment in Uruguay, where audio files are collected through a chatbot, whose implementation is not covered in this paper. Finally, we lay out limitations of our approach and potential improvements.

**Index Terms**: automated analyses, acoustic analyses, infant-parent interaction

## 1. Introduction

There are many situations in which it would be useful to provide personalized feedback based on the automated analysis of audio samples. For example, many early childhood interventions today focus on changing parental behaviors that have vocal correlates, such as inviting parents to use Parentese [1]; and audio samples can also help in assessing the progress of neurodegenerative disorders [2]. In this paper, we explain in detail a pipeline that we created that allows automated analyses of audio samples by focusing on the former use case, but we believe many of the ideas and technical solutions described here can be used in other cases.

* LP, WH, and GV are co-first authors. This work benefited from the support of the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (ExELang, Grant agreement No. 101001095), the J. S. McDonnell Foundation Understanding Human Cognition Scholar Award, the IADB and the Bernard van Leer Foundation

### 1.1. Related work

The idea that we can provide feedback based on automated analyses of speech, including that collected from a distance, is far from new [3]. More recently, however, the implementation of such systems has been dramatically facilitated by widespread access to the internet around the globe and the appearance of affordable and versatile cloud services.

Perhaps the most salient field in which this idea has been implemented is automated feedback system used for teaching. Countless solutions have been developed, in the hope of providing a learning experience close to what a human teacher would offer at much lower operating costs. For example, a recent review [4] evaluated and classified 109 such systems.

Automated feedback tools for speech also exist in the domain for which we developed the solution presented here, namely parent-child interaction. The most famous may be the LENA system, a solution proposed for perceived disparities in language stimulation in USA [5] by providing caregivers with numerical summaries of how much they talked around the child, how many conversational turns parent and child engaged in, and how much the child vocalized. Unlike the solution we propose below, children have to wear a recording device over a whole day, and the data are uploaded by connecting this device to a computer or the cloud. Although the system was initially conceptualized to be a "business to customer" solution, that would be purchased by individual families, who would receive feedback on the previous day's recording by connecting to the cloud, it is currently rather marketed in the context of public health approaches, and families do not receive immediate feedback. Notice additionally that the reliance on a purpose-built hardware (that costs about 300-400US$ per unit) also makes it a difficult choice for low-resource settings. Additionally, since data are analyzed by the LENA Foundation in their American clusters, this may entail some difficulties depending on country-specific legislation. For instance, in Europe, such a transfer would require written consent including several lines of information, which may cause families to worry and decline participation.

We have not been able to find a tool that provides caregivers with immediate feedback on their interactions with children, despite the fact that there are useful tools that have been developed to facilitate speech analyses. For instance, SoundCount [6] is an open-source web-based tool built to allow researchers to more easily analyze speech through third-party ASR systems, such as

Google Cloud or Sphinx, and return descriptive features such as number of words and transcript as well as gender, age and dialect of the speaker in the audio. The authors mention the potential use case of book reading. Although we think this is a fantastic initiative, the solution assumes a specific context for interactions that are not spontaneous and may not be the typical way in which a specific dyad interacts; indeed, some families find it unnatural to read a book to their infant, and prefer singing to/with their child, or merely conversing while playing with toys. Moreover, this system assumes that the initial recordings were previously split into smaller files each only containing speech from a single speaker, and thus the files need to be processed by researchers. Finally, the reliance on an ASR system means that only families speaking well-resourced languages can benefit from this system, and the fact that these are third-party systems also implies certain ethical and legal challenges related to data sharing.

Here, we present a tool that takes a rather different approach from SoundCount. To begin with, our tool does speaker diarization [7] to avoid having to perform human annotation tasks beforehand and which is more fitting to truly spontaneous parent-child interaction. We rely on open source models that have been trained multilingually, including on low-resourced languages and multilingual settings, to make the system more flexible (and in fact, the models we use can also be retrained with domain-specific data). It also avoids considerations of data sharing when dealing with third parties, especially given the sensitive nature of child-centered recordings.

### 1.2. Our use case

This pipeline was developed in the context of a larger randomized control trial (RCT), in which families whose children are traditionally at higher risk of language and cognitive delays were enrolled. This RCT was designed, implemented, and evaluated by a team separate from the one implementing the current solution, and therefore a full description of the RCT is beyond the scope of the present paper. Here, we describe only general features that can aid in the comprehension of the technical solution described here. There was a control group and several experimental treatment groups, totaling over a thousand families. All the treatment groups had in common a curriculum which provided caregivers with information about child development and caregiver behaviors that are beneficial to that development, according to previous work (e.g., [1]). Building on previous work by the team leading the RCT showing that messages reinforce caregiver behavior changes, a subset of families were assigned to a special treatment group, which additionally had the opportunity to hold "conversations" with a chatbot in WhatsApp, chatbot that was developed by the company SimpleTech. A subset of this subset (about 300 families) were further encouraged to upload audio messages via WhatsApp, in which case the audio was passed on to our analysis pipeline.

Families were asked to record these audios in a relatively quiet situation, for instance reading a book, singing to the child, or playing with the child. The team originally considered including recommendations to record during mealtime and bathtime, since these can provide "quality time" between caregivers and children, but the idea was abandoned given that such audio contexts will likely contain a great deal of background noise and thus be challenging to analyze. Families were asked to record these short interactions ideally when other people were not around, so that we came to expect just the key child (the focus of the intervention, typically aged 3 years or younger) with one caregiver (often the mother).

Given that it was the first time such a service was offered to these families, we had no idea about how often families would upload audios, how long these would be, and how quickly they would want feedback. Taking into account also cost, we assumed families would upload maximally one audio per week; we capped analysis time to the first 10 minutes of audio (and required the minimum duration to be 1 minute); and we batched analyses and feedback provision to be carried out once a day.

The phases of this process that are covered in this report are in Figure 1. We do not cover here how a chatbot was set up in WhatsApp, how conversations were established with hundreds of families at once, nor how audios that the family shares in a conversation with this chatbot are extracted from this conversation, since these are steps developed by our partners SimpleTech. Similarly, we do not provide costing for these steps nor technical/security analysis for them.

**Other potential use cases.** We believe this pipeline can be re-purposed for a variety of other purposes. We provide here a short list to help readers decide in which type of re-use case they are in, and the estimated additional effort it would take them to adapt our pipeline.

**Minimal effort.** Readers working on caregiver-child interaction in a similar population (e.g. same language, same kind of smartphone hardware) and who want to use the exact same feedback set-up can directly borrow our code. They will need to set up an AWS account and replicate the setup as described below.

**Additional effort to check that automated audio analyses perform well enough.** Some readers working on caregiver-child interaction may intend to collect data in a substantially different population (e.g., a language that is very different from Spanish, with children who are older than 3 years of age, interested in the interaction between the child and other children, or interactions happening during "noisy" activities, like mealtimes and daycare) and who want to use the exact same feedback set up can directly borrow our code. In this case, in addition to the adaptation work described above, we highly recommend human annotation for a random or representative sample of audios, so that performance of audio analysis routines can be evaluated.

**Additional effort to adapt the automated analyses.** If comparison against human annotation reveals performance is insufficient, retraining will be needed. Retraining is also needed for a substantially different use case, such as collecting audio samples from people with neurodegenerative disorders, to assess the progress of their symptoms. In the latter case, the metrics we employ are likely irrelevant and others are more informative. Fortunately, the audio analyses build on open source retrainable software [7, 8, 9].

**Additional effort to adapt the VM for scale.** We developed this solution assuming that maximally, we'd receive 300 10-min files in a day, and that feedback could take up to 24h to arrive. If more or longer files are expected, then the parameters of the VM need to be revised. One option is to analyze more frequently, every 6h rather than once a day, which would also allow faster feedback. Attention should be paid to the risk that space requirements are exceeded; and/or that the time needed to analyze one batch bleeds into the time for the next batch.

### 1.3. General architecture of the software

The pipeline is described including technical details below, but in a nutshell, the following phases are covered in this report (see Figure 1):

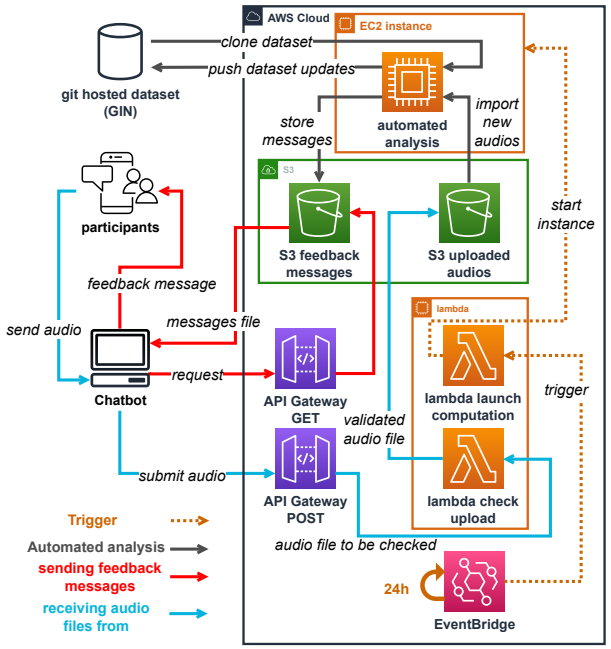


Figure 1: *Graphical representation of the pipeline phases covered in this report*

- The audio file is uploaded to our API and preliminary tests are run, to provide instant automated error messages. If no error is found, the file is kept for analyses.
- Once a day, the VM is turned on, the dataset and analysis package are installed
- Automated analyses are carried out. These include four open-source modules (VTC, ALICE, VCM, pychattr), which do (respectively) voice type classification, estimation of adult word/syllable/phone counts, classification of key child vocalization's vocal types (into more and less advanced speech-like utterances versus crying and laughing), and estimation of conversational turns. In addition, sections attributed to an adult and/or key child were submitted to acoustic analyses to extract acoustic parameters like f0 (a correlate of pitch, which is a key variable in Parentese, REF). The automated analyses resulted in file-level metrics such as the average number of words uttered by the caregiver.
- The RCT leading team decided to provide feedback based on a comparison between a previous audio and the last one by the same family (with a default message being provided for the first audio from that family). This allowed simpler messages (e.g., "you talked more" rather than "you talked 17 words per minute").

We do not cover here how a chatbot was set up in WhatsApp, how conversations were established with hundreds of families at once, nor how audios that the family shares in a conversation with this chatbot are extracted from this conversation, since these are steps developed by our partners SimpleTech. Similarly, we do not provide costing for these steps nor technical/security analysis for them.

## 2. Technical description of the pipeline

In this section, we provide a more detailed technical description of each of three key aspects in our pipeline: 1) an API through which audios can be uploaded and resulting messages retrieved; 2) managing the dataset and its structure; 3) analyzing audios, and preparing a personalized message.

To provide a constantly available upload platform and a self-running analysis, we rely on using Amazon Web Services (AWS)[1]. While it is possible to set up a similar pipeline on other cloud services, we will limit our explanation to this specific platform. To learn more about the costs of using AWS, refer to our Evaluation section.

AWS services that we use and reference in the following description of the pipeline[2]:

- S3: cloud storage service
- EC2: Virtual Machine service
- Lambda: cloud code execution service
- API Gateway: REST API service
- EventBridge: service to define trigger events

### 2.1. API

The pipeline only requires end users to perform two operations, that is: 1) uploading the audio files and 2) downloading the resulting messages to dispatch to the families. Both of those operations are conducted by an API hosted on AWS. We include checks and messages to the users in both of these phases, which we detail here as well.

#### *2.1.1. Uploading*

An overview of the upload phase (including checks and messages) is provided in Figure 2. The upload is done by dispatching a POST request to the API upload endpoint provided by Amazon when creating the API. The content of the file is sent directly in the query along with a file name. For the request to be accepted, it must include a secret API key, preventing unauthorized parties from sending unwanted audio files to the pipeline. Once the file is received by the API server, it is sent to a validation process using the AWS lambda service. This service is running code (in our case python code) to perform a number of checks before the file is accepted. We include controls on:

- Format of the file (we accept .ogg only, determined by our target use case, in which a WhatsApp chatbot receives the audios)
- Integrity of file content
- Length of audio (should be at least 60 seconds)
- Maximum size of the file (should be smaller than 5MB)

It is possible to add other tests such as checking the sample rate as long as those checks do not include information that is obtained by the pipeline analysis, e.g. rejecting a file because we did not find any child vocalization. When a submitted file does not meet one of the requirements, an error code and an explanation message are returned in response to the request. Once successfully controlled, the recording is then stored into S3, which we use in this case as a temporary storage location until the

[1]To start using AWS, you will need an account. If you don't already have one, create a new account at https://portal.aws.amazon.com/billing/signup#/start/email

[2]We do not include in this description the service for managing users and permissions, called IAM. This is used to set up permissions. For instance, when the AWS account is created, this is automatically the 'root' account, which is used to create the first Administrator users inside the IAM platform. Then administrators can sign in to their account to set up the pipeline. Incidentally, we remind readers that it is good practice not to use the 'root' account directly after that.

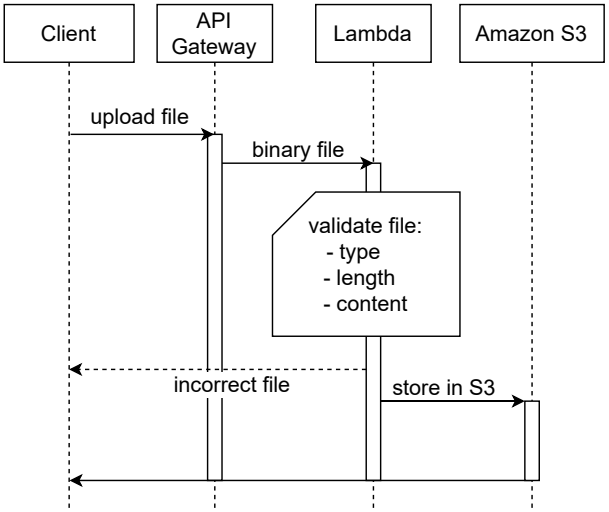


Figure 2: *uploading a new audio file*

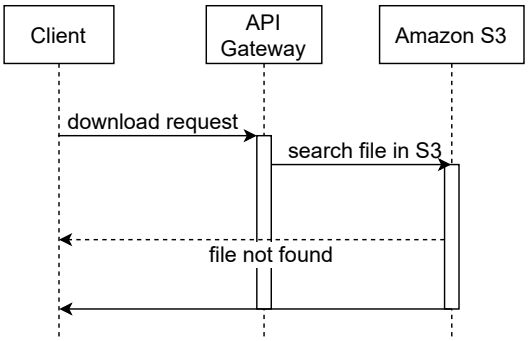


Figure 3: *download of a message file*

daily analysis is run.

#### *2.1.2. Downloading*

The pipeline analysis (which will be detailed below) outputs messages files in a csv format. Those files contain a list of messages as a series of responses to dispatch to the participants. An overview of the downloading phase is given in Figure 3. In order to get those files, in a similar process to what is done to upload, A GET request is sent to the API download endpoint, including a valid API key to avoid responding to requests from unauthorized sources. In the context of our use case, we agreed with our chatbot partners that we would name the output files "messages_AAAAMMDD.csv" where AAAA is the year, MM the month and DD the day on which we received the recordings. This allows our partners to submit a query for a file containing all feedback messages for each day. S3 is used as storage for all the output files, keeping them available even after the day has passed. If the request makes reference to a file that does not exist, a "file not found" message is returned.

### 2.2. Managing the dataset

The collected audio files as well as all the different analysis outputs are assembled into a dedicated dataset. So as to build on current routines in our team, we build on standards and a server that we have described elsewhere [10]. In a nutshell, the dataset is organized using a specific set of standards, which allows us to employ functions that we have used in other work. The database thus organized is archived in a scientific data hosting platform which ensures 1) appropriate handling of sound files, 2) versioning via git, 3) availability to authorized collaborators, and 4) a robust longer-term storing solution. Other users may prefer to spend effort to skip this step altogether, if they are only focused on providing feedback, and do not want a full copy of the dataset accessible for additional scientific analyses. Setting up the initial dataset is made easier by a script made available generating the required structure if it does not already exist. The addition of new audio files to the dataset is done by another script. We made the decision to rely only on information that came with the file itself, so as to avoid having too much personal information about participants and handling metadata files that could be inconsistent. Our solution was to require users to name files with the following pattern. "CHILD-ID_[info1_info2_..._infoX]_YYYYMMDD_HHMMSS.xxx" where:

- CHILD-ID is a unique identifier for a specific child, allowing the pipeline to track the evolution for each child.
- Info1 / info2 / infoX are relevant information to be kept along the audio file, the pipeline will not use those fields, but they will be integrated into the metadata.
- YYYYMMDD the date the audio file was recorded on, e.g. 20230131 for Jan 31st 2023
- HHMMSS the hour, in a 24h format, the audio file started at, e.g. 235712 for 23h57'12"
- .xxx the file format

Using this pattern, the script attempts to index in the metadata of the dataset the newly detected audio files in the "recordings/raw" folder.

### 2.3. Automated analysis

In this part, we describe the infrastructure behind the daily automated running of the computation as well as the actual software used to analyze and extract exploitable information from audio files.

#### *2.3.1. Daily running of the computation*

We want to provide feedback to the families within a reasonable timeframe while maintaining the overall cost low. We must also make sure that a computation has enough time to complete before the next one is called. In that context, the best compromise was to run a full computation every 24 hours. Other projects with different constraints may want to change that delay to be longer or shorter depending on their use case. We also considered triggering the computation for every upload received but this would cause significant risks to have simultaneous modification of a dataset which will cause editing conflicts that cannot be resolved automatically. The computation environment is a Virtual Machine (VM) hosted on the Amazon EC2 service. That environment is independent from our local infrastructure and as such, does not require as much work to set up and maintain, so it has an overall lower cost. It is also an easily scalable solution if more computation power is needed to handle more audio files. From one of the default VMs provided by AWS, we built a custom VM machine image (designated as AMi in AWS) integrating the necessary software and authorization keys necessary to download and modify the dataset. This way, at launch, the VM does not need to reinstall anything and is already ready

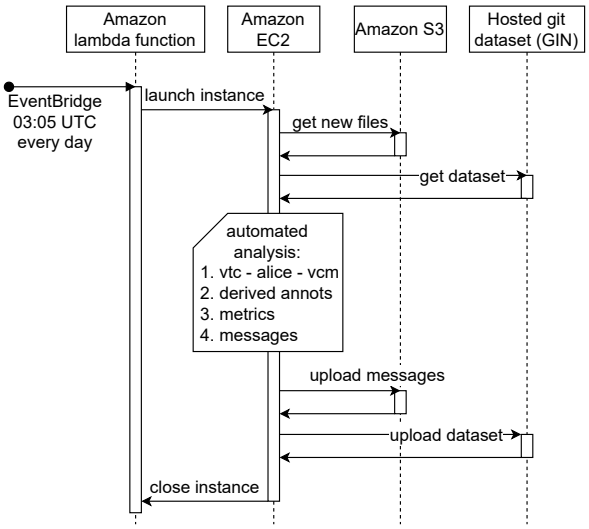


Figure 4: *sequence of a computation run*

to run the analysis. To launch the computation, the Amazon EventBridge service is issuing a trigger every day at 0:05 local time of the intervention area. The 5 minute delay is intended to leave enough time for an upload started before 0:00 to fully complete. A lambda function then starts an instance of the custom VM image that will run the actual analysis. The different components called for a computation run are described in Figure 4.

### 2.3.2. *Deriving other annotations*

From the baseline of annotations we obtain thanks to the models mentioned above, we are able to extract additional derived annotations. In our use case, we want to extract the following information:

1. Acoustic: we extract acoustic features such as the median pitch of the audio for each vocalization identified by VTC.
2. Conversations: we connect VTC vocalizations together to assert the presence of conversations in the audio. This gives us information on turn taking and number of interactions in conversations for example. The connection of vocalizations is conducted by an open-source package (REF) that is highly configurable.

These derived annotations are integrated in the dataset. Other kinds of derived annotations that one's would deem relevant to have would easily be added to this list by providing and linking to the project the python code carrying out the derivation.

### 2.3.3. *Computing metrics*

The computation of metrics is a built-in function of the package we use to organize our dataset. It allows the extraction of a list of usable values from the huge amount of information contained inside the complete annotations. We have already defined many likely useful metrics (e.g. average duration of child vocalizations), and future users would be able to adapt these routines to other calculations they would need. The pipeline extracts all of our standard metrics for VTC, ALICE and VCM (listed here REF) as well as additional metrics based on the derived annotations, for example the average pitch of each speaker category and the number of turn transitions. Those metrics are calculated

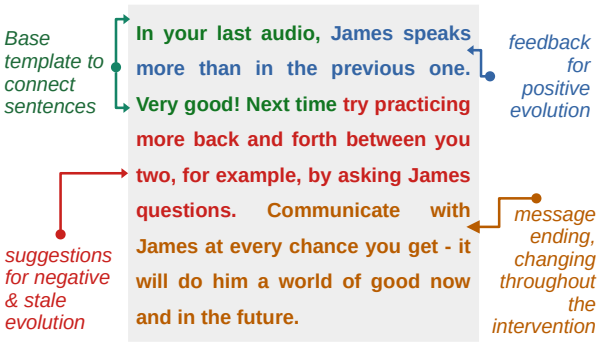


Figure 5: *Structure of messages.*

at the level of the recording, meaning that every new integrated audio file will have a specific set of metrics with values coming only from that file. This extracted metric information is saved and uploaded in the dataset.

### 2.3.4. *Generating the content of personalized messages*

Another feature of our system again reflected the use case that this solution was developed for. Given that the intervention aims to enable caregivers to change the way they talk to their children, it was decided that the messages generated would reflect the evolution measured in the different metrics for a specific child, rather than just the current metrics. This evolution measure is obtained by comparing, for each child, the metrics of the last audio received to the metrics of the previous audio. For each metric, we record a positive evolution if the value has grown and negative otherwise. In addition, since we planned on providing this as feedback in a simple message to parents, we did not want to overload them with information by telling them about all changes. Instead, only a subset of the extracted metrics that was chosen to best capture caregiver-child interaction is considered, taking also into account the aspects of such interactions that are discussed in the information provided to families as part of the intervention. The pipeline uses the evolution measure along with a file defining the content and structure to use in order to create the final messages. The messages are highly customisable by defining into the description file:

- Each metric label that should be taken into account in the generation and for each of them two sentences, one for when the metric has recorded a positive evolution and one for a negative evolution.
- Templates of message to use depending on the evolution measured on all metrics.
- Fixed sentences that can be reused in the templates without any modification.
- Variable sentences that will evolve according to what stage of the intervention we are in.

The steps of generating a message and how the description file is used are exemplified in Figure 5.

A new csv file containing all the messages for a specific day is created and saved in the dataset. This file is also stored in the Amazon S3 storage to make it downloadable with the API. The computation run is considered completed after the save of the messages file and the process can terminate.

A new CSV file containing all the messages for a specific day is created and saved in the dataset. This file is also stored in

the Amazon S3 storage to make it downloadable with the API. The computation run is considered completed after the save of the messages file and the process can terminate.

## 3. Legality, ethics and security

Families agree to share their audios in the context of the ongoing study. To set up the consent form, we consulted legislation in Uruguay [11]. Uruguay's data protection laws are extensive, and current local recommendations are even more detailed than those found in Europe. Uruguayan law gives their citizens a right to know what data are being collected, for what purposes, and who will have access to them. They similarly have rights of accessing, rectifying, and requiring the deletion of those data. However, they have more rights than Europeans in that they have the right to be told what metrics or parameters are being used for algorithmic-based decisions on anything that has consequences for their wellbeing or economic outcomes. Presumably, this regulation was put in place in the context of e.g., authorizing a loan or calculating health primes, but we believe it is also relevant in the case of the current study, since at least based on current data, these interventions can have positive impacts on the children's academic outcomes.

The team complies with obligations regarding right of information, rectification, etc., by keeping a record that relates the family's identity to the audios. Thus, if a family decides to withdraw, their data can be safely removed from the system. We reduce risk by making sure that only the Uruguayan collaborators have access to the file linking phone numbers and individual identities. We comply with the obligation of informing them about which parameters are extracted from their data by aligning the metrics we select to provide feedback on with the content of the parental training. For instance, some of the weeks focus on using Parentese, and in those weeks, our metrics provide feedback on the pitch parents used.

In addition to complying with legal requirements, we also took into account additional ethical dimensions. Although we set up our system such that we do not have access to the child's or parents' names or other textual identifying information, the audios contain voices of family members, and voice constitutes biometric data according to most definitions of biometric. For instance, if a family posted videos of themselves on Youtube or Facebook, a third party gaining access to the audios could cross the two sources in order to identify the family. They could also run analyses on the audios to find evidence of health markers (e.g., using the audio to check for potential depression). Given all of this, we treat these audios as sensitive identifying data.

The audio transfer between the family and the chatbot is encrypted end to end thanks to WhatsApp algorithms. IS THE AUDIO TRANSMITTED TO US ENCRYPTED? WHAT OTHER MEASURES HAVE WE PUT IN PLACE TO PROTECT FAMILIES?

From both the legal and ethical standpoint, it is important to use these data for the original, restricted purpose for which they were collected. Audios are archived solely with the purpose of evaluating and potentially improving our automated algorithms. We specifically intend not to use these audios for unrelated purposes, such as developing algorithms to measure depression or other dimensions that could be represented in them. We have not specified a time after which they will be deleted, as at present speech technology tools in this domain are progressing relatively slowly, and thus we cannot foresee when technology will be good enough to warrant a shorter time cycle.

## 4. Estimation of the resources necessary

In this section, we attempt to provide readers with an idea of what resources they need to replicate our system. The entire pipeline encapsulates a lot of different technologies (AWS, API REST with http, python pandas, git, datalad, linux bash, ssh keys) that interact together. A slight misconfiguration can easily lead to a failure of the entire process. For this reason, a certain level of expertise in software development and/or computer coding is required. While it is not necessary to master all of those technologies, a solid understanding of the way they work and interact together (or a good deal of patience and experience debugging) is required to both set things up for a new AWS account and fix errors in the pipeline that may emerge when setting things up de novo. We may also be overly conservative: We would be grateful if readers of this manuscript share their experience trying to replicate this system, which would allow us to provide a more accurate estimate of the level of expertise required.

The amount of time to dedicate to run the pipeline will depend on the knowledge and number of people involved. For instance, it took two people contributing an estimated XX hours total to craft the system we currently have; presumably, it should take less than that to simply replicate it. In contrast, maintaining the pipeline during the intervention will require little investment. For the most part, the pipeline will run on its own. In our case, since we moved to production, it took one person less than 10 hours to deal with unanticipated issues during the first month of the intervention.

Regarding the cost of the processing, the only paying component is Amazon AWS and more specifically the usage of the Virtual Machine taking care of the processing. There are a lot of different plans on Amazon offering different levels of processing power and processing time for varying prices. We estimate that taking a suitable plan for the amount of audio to process and using a daily computation will cost around 0.20 US$ per hour of audio analyzed.

## 5. Conclusions

In this report, we describe how we set up a system that analyzes audios of conversations and returns information based on metrics that are automatically computed. We hope this description and the open-source software on which we build will prove useful to readers.

## 6. Acknowledgements

We are grateful to the families who provided pilot data and to annotators? Anyone else?

## 7. References

[1] N. Ferjan Ramírez, S. R. Lytle, M. Fish, and P. K. Kuhl, "Parent coaching at 6 and 10 months improves language outcomes at 14 months: A randomized controlled trial," *Developmental science*, vol. 22, no. 3, p. e12762, 2019.

[2] C. Gallezot, R. Riad, H. Titeux, L. Lemoine, J. Montillot, A. Sliwinski, J. H. Bagnou, X. N. Cao, K. Youssov, E. Dupoux *et al.*, "Emotion expression through spoken language in huntington disease," *Cortex*, vol. 155, pp. 150–161, 2022.

[3] R. J. Moran, R. B. Reilly, P. de Chazal, and P. D. Lacy, "Telephony-based voice pathology assessment using automated speech analysis," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 3, pp. 468–477, 2006.

[4] G. Deeva, D. Bogdanova, E. Serral, M. Snoeck, and J. De Weerdt, "A review of automated feedback systems for learners: Classification framework, challenges and opportunities," *Computers & Education*, vol. 162, p. 104094, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S036013152030292X

[5] C. R. Greenwood, J. J. Carta, D. Walker, J. Watson-Thompson, J. Gilkerson, A. L. Larson, and A. Schnitz, "Conceptualizing a public health prevention intervention for bridging the 30 million word gap," *Clinical Child and Family Psychology Review*, vol. 20, pp. 3–24, 2017.

[6] M. Schmidt, R. Walters, B. Ault, K. Poudel, A. Mischke, S. Jones, A. Sockhecke, M. Spears, P. Clarke, R. Makram, S. Meagher, M. Sarkar, J. Wade, and N. Sarkar, "A simple web utility for automatic speech quantification in dyadic reading interactions," in *Learning and Collaboration Technologies. Ubiquitous and Virtual Environments for Learning and Collaboration*, P. Zaphiris and A. Ioannou, Eds. Cham: Springer International Publishing, 2019, pp. 122–130.

[7] M. Lavechin, R. Bousbib, H. Bredin, E. Dupoux, and A. Cristia, "An open-source voice type classifier for child-centered daylong recordings," in *Interspeech*, 2020.

[8] N. Al Futaisi, Z. Zhang, A. Cristia, A. Warlaumont, and B. Schuller, "Vcmnet: Weakly supervised learning for automatic infant vocalisation maturity analysis," in *2019 International Conference on Multimodal Interaction*, 2019, pp. 205–209.

[9] O. Räsänen, S. Seshadri, M. Lavechin, A. Cristia, and M. Casillas, "Alice: An open-source tool for automatic measurement of phoneme, syllable, and word counts from child-centered daylong recordings," *Behavior Research Methods*, vol. 53, pp. 818–835, 2021.

[10] L. Gautheron, N. Rochat, and A. Cristia, "Managing, storing, and sharing long-form recordings and their annotations," *Language Resources and Evaluation*, pp. 1–33, 2022.

[11] M. Korwin-Zmijowski and A. Cristia, "Regulation relevant to (long-form) audio recordings gathered in uruguay," Sep 2022. [Online]. Available: https://osf.io/8m3ev