An open-source voice type classifier for child-centered daylong recordings

Marvin Lavechin1,2, Ruben Bousbib1,2, Hervé Bredin3, Emmanuel Dupoux1,2, Alejandrina Cristia1

1ENS-PSL/CNRS/EHESS, Paris, France;
2INRIA Paris, France;
3LIMSI, CNRS, Univ. Paris-Sud, Université Paris-Saclay, Orsay, France.

marvinlavechin@gmail.com, alecristia@gmail.com

Abstract

Spontaneous conversations in real-world settings such as those found in child-centered recordings have been shown to be amongst the most challenging audio files to process. Nevertheless, building speech processing models handling such a wide variety of conditions would be particularly useful for language acquisition studies in which researchers are interested in the quantity and quality of the speech that children hear and produce, as well as for early diagnosis and measuring effects of remediation. In this paper, we present our approach to designing an open-source neural network to classify audio segments into vocalizations produced by the child wearing the recording device, vocalizations produced by other children, adult male speech, and adult female speech. To this end, we gathered diverse child-centered corpora which sum up to a total of 260 hours of recordings and covers 10 languages. Our model can be used as input for downstream tasks such as estimating the number of words produced by adult speakers, or the number of linguistic units produced by children. Our architecture combines SincNet filters with a stack of recurrent layers and outperforms by a large margin the state-of-the-art system, the Language Environment Analysis (LENA) that has been used in numerous child language studies.

Index Terms: Child-Centered Recordings, Voice Type Classification, SincNet, Long Short-Term Memory, Speech Processing, LENA

1. Introduction and related work

In the past, language acquisition researchers’ main material was short recordings [1] or times of in-person observations [2]. However, investigating the language phenomenon in this manner can lead to biased observations, potentially resulting in divergent conclusions [3]. More recently, technology has allowed researchers to efficiently collect and analyze recordings over a whole day. By the combined use of a small wearable device and speech processing algorithms, one can get meaningful insights of children’s daily language experiences. While daylong recordings are becoming a central tool for studying how children learn language, a relatively small effort has been made to propose robust and bias-free speech processing models to analyze such data. It may however be noticed that some collaborative works that benefit both the speech processing and the child language acquisition communities have been done. In particular, we may cite Homebank, an online repository of day-long child-centered audio recordings [4] that allow researchers to share data more easily. Some efforts have also been made to gather state-of-the-art pretrained speech processing models in DiViMe [5], a user-friendly and open-source virtual machine. Challenges and workshops using child-centered recordings [6, 7], also attracted the attention of the speech processing community. Additionally, the task of classifying audio events has often been addressed in the speech technology literature. In particular, the speech activity detection task [8] or the acoustic event detection problem [9, 10] are similar to the voice type classification task we address in this paper.

Given the lack of open-source and easy-to-use speech processing models for treating child-centered recordings, researchers have been relying, for the most part, on the Language Environment Analysis (LENA) software [11] to extract meaningful information about children’s language environment. This system will be introduced in more detail in the next section.

1.1. The LENA system

The LENA system consists of a small wearable device combined with an automated vocal analysis pipeline that can be used to study child language acquisition. The audio recorder has been designed to be worn by young children as they go through a typical day. In the current LENA system, after a full day of audio has been captured by the recorder, the audio files are transferred to a cloud and analyzed by signal processing models. These latter have been trained on 150 hours of proprietary audio collected from recorders worn by American English-speaking children. The speech processing pipeline consists of the following steps [11, 13, 14]:

1. First, the audio is segmented into mutually exclusive categories that include: key child vocalizations (i.e., vocalizations produced by the child wearing the recording device), adult male speech, adult female speech, other child speech, overlapping sounds, noise, and electronic sounds.

2. The key child vocalization segments are further categorized into speech and non-speech sounds. Speech encompasses not only words, but also babbling and pre-speech communicative sounds (such as squeals and growls). Child non-speech sounds include emotional reactions such as cries, screams, laughs and vegetative sounds.

3. A model based on a phone decoder estimates the number of words in each adult speech segment.

4. Further analyses are performed to detect conversational turns, or back and forth alternations between the key child and an adult.
The LENA system has been used in multiple studies covering a wide range of expertise including a vocal analysis of children suffering from hearing loss [15], the assessment of a parent coaching intervention [16], and a study of autism spectrum disorders [17]. An extensive effort has been made to assess the performance of the LENA speech processing pipeline [18, 19, 20].

Despite its wide use in the child language community, LENA imposes several limiting factors to scientific progress. First, as their software is closed source, there is no way to build upon their models to improve performance, and we cannot be certain about all design choices and their potential impact on performance. Moreover, since their models have been trained only on American English-speaking children recorded with one specific piece of hardware in urban settings, the model might potentially be overfit to these settings, with a loss of generalization to other languages, cultures, and recording devices.

1.2. The present work

Our work aims at proposing a viable open-source alternative to LENA for classifying audio frames into segments of key child vocalizations, adult male speech, adult female speech, other child vocalizations, and silence. The general architecture is presented in 2.1. Additionally, we gathered multiple child-centered corpora covering a wide range of conditions to train our model and compare it against LENA. This data set is described in further details in 2.2.

2. Experiments

2.1. End-to-end voice type classification

The voice type classification problem can be described as the task of identifying voice signal sources in a given audio stream. It can be tackled as a multi-label classification problem where the input is the audio stream divided into N frames \( S = \{ s_1, s_2, \ldots, s_N \} \) and the expected output is the corresponding sequence of labels \( y = \{ y_1, y_2, \ldots, y_N \} \) where each \( y_i \) is of dimension \( K \) (the number of labels) with \( y_{i,j} = 1 \) if the \( j^{th} \) class is activated, \( y_{i,j} = 0 \) otherwise. Note that, in the multi-label setup, multiple classes can be activated at the same time.

At training time, fixed-length sub-sequences made of multiple successive frames, are drawn randomly from the training set to form mini-batches of size \( M \).

As illustrated in Figure 1, these fixed-length sub-sequences are processed by a SincNet [12] that aims at learning meaningful filter banks specifically customized to solve the voice type classification task. These low-level signal representations are then fed into a stack of bi-directional long short-term memory (LSTM) layers followed by a stack of feed-forward (FF) layers. Finally, the sigmoid activation function is applied to the final output layer of dimension \( K \) so that each predicted score \( \hat{y}_{i,j} \) consists of a number ranging between 0 and 1. The network is trained to minimize the binary cross-entropy loss:

\[
    \mathcal{L} = - \frac{1}{KM} \sum_{i=1}^{M} \sum_{j=1}^{K} y_{i,j} \log \hat{y}_{i,j} + (1 - y_{i,j}) \log (1 - \hat{y}_{i,j})
\]

At test time, audio files are processed using overlapping sliding sub-sequences of the same length as the one used in training. For each time step \( t \), and each class \( j \), this results in several overlapping sequences of prediction scores, which are averaged to obtain the final score for class \( j \). Finally, time steps with prediction scores greater than a tunable threshold \( \sigma_j \) are marked as being activated for the class \( j \).

Our use case considers \( K = 5 \) different classes or sources which are:

- **KCHI**, for key-child vocalizations, i.e., vocalizations produced by the child wearing the recording device
- **OCH**, for all the vocalizations produced by other children in the environment
- **FEM**, for adult female speech
- **MAL**, for adult male speech
- **SPEECH**, for when there is speech

As the LENA voice type classification model is often used to sample audio in order to extract segments containing the most speech, it appeared to us that it was useful to consider a class for speech segments produced by any type of speaker. Moreover, in our data set, some of the segments have been annotated as **UNK** (for unknown) when the annotator was not certain of which type of speaker was speaking (See Table 1). Considering the SPEECH class allows our model to handle these cases.

One major design difference with the LENA model is that we chose to treat the problem as a multi-label classification task, hence multiple classes can be activated at the same time (e.g., in case of overlapping speech). In contrast, LENA treats the problem as a multi-class classification task where only one class can be activated at a given time step. In the case of overlapping speech, LENA model returns the **OVL** class (which is also used for overlap between speech and noise). More details about the performance obtained by LENA on this class can be found in [18].

2.2. Datasets

In order to train our model, we gathered multiple child-centered corpora data [21, 22, 23, 24, 25, 26, 27, 28, 29, 30] drawn from various child-centered sources, several of which were not day-long. Importantly, the recordings used for this work cover a
Table 1: Description of the BabyTrain data set. Child-centered corpora included cover a wide range of conditions (including different languages and recording devices). ACLEW-Random is kept as a hold-out data set on which LENA and our model are compared. DB correspond to datasets that can be found on Databrary, HB the ones that can be found on Homebank.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Access</th>
<th>LENA-recorded?</th>
<th>Language</th>
<th>Tot. Dur.</th>
<th>KCHI</th>
<th>OCH</th>
<th>MAL</th>
<th>FEM</th>
<th>UNK</th>
</tr>
</thead>
<tbody>
<tr>
<td>BabyTrain</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACLEW-Starter</td>
<td>mixture (DB)</td>
<td>mostly</td>
<td>Mixture</td>
<td>1h30m</td>
<td>10m</td>
<td>5m</td>
<td>6m</td>
<td>20m</td>
<td>0m</td>
</tr>
<tr>
<td>Lena Lyon</td>
<td>private (HB)</td>
<td>yes</td>
<td>French</td>
<td>26h51m</td>
<td>4h33m</td>
<td>1h14m</td>
<td>1h9m</td>
<td>5h02m</td>
<td>1h0m</td>
</tr>
<tr>
<td>Namibia</td>
<td>upon agreement</td>
<td>no</td>
<td>Ju’hoan</td>
<td>23h44m</td>
<td>1h56m</td>
<td>1h32m</td>
<td>41m</td>
<td>2h22m</td>
<td>1h01m</td>
</tr>
<tr>
<td>Paido</td>
<td>public (HB)</td>
<td>no</td>
<td>Greek, Eng., Jap.</td>
<td>40h08m</td>
<td>10h5m</td>
<td>0m</td>
<td>0m</td>
<td>0m</td>
<td>0m</td>
</tr>
<tr>
<td>Tsay</td>
<td>public (HB)</td>
<td>no</td>
<td>Mandarin</td>
<td>132h02m</td>
<td>34h07m</td>
<td>2h08m</td>
<td>10m</td>
<td>57h31m</td>
<td>28m</td>
</tr>
<tr>
<td>Tsimane</td>
<td>upon agreement</td>
<td>mostly</td>
<td>Tsimane</td>
<td>9h30m</td>
<td>37m</td>
<td>23m</td>
<td>11m</td>
<td>28m</td>
<td>0m</td>
</tr>
<tr>
<td>Vanuatu</td>
<td>upon agreement</td>
<td>no</td>
<td>Mixture</td>
<td>2h29m</td>
<td>12m</td>
<td>5m</td>
<td>5m</td>
<td>9m</td>
<td>1m</td>
</tr>
<tr>
<td>WAR2</td>
<td>public (DB)</td>
<td>yes</td>
<td>English (US)</td>
<td>50m</td>
<td>14m</td>
<td>0m</td>
<td>0m</td>
<td>0m</td>
<td>9m</td>
</tr>
</tbody>
</table>

Hold-out set

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ACLEW-Random</td>
<td>private (DB)</td>
<td>yes</td>
<td>Mixture</td>
<td>20h</td>
<td>1h39m</td>
<td>45m</td>
<td>43m</td>
<td>2h48m</td>
<td>0m</td>
</tr>
</tbody>
</table>

2.3. Evaluation metric

For each class, we use the F-measure between precision and recall, such as implemented in pyannote.metrics [31] to evaluate our systems:

\[
F\text{-measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
\]

where precision = \( tp/(tp + fp) \) and recall = \( tp/(tp + fn) \) with:

- \( tp \) the duration of true positives
- \( fp \) the duration of false positives
- \( fn \) the duration of false negatives

We select our models by averaging the F-measure across the 5 classes. Note that these 5 metrics have been computed in a binary fashion, where the predictions of our model for a given class were compared to all reference speaker turns such as provided by the human annotations (no matter if the latter were overlapping or not). In diarization studies, the choice of a collar around every reference speaker turns is often made to account for inaccuracies in the reference labels. We chose not to do so, consequently all numbers reported in this paper can be considered as having a collar equal to 0.

2.4. Implementation details

Figure 1 illustrates the broad architecture used in all experiments. For SincNet, we use the configuration proposed by the authors of the original paper [12]. All LSTMs and inner feed-forward layers have a size of 128 and use \( \text{tanh} \) activations. The last feed-forward layer uses a sigmoid activation function.

Data augmentation is applied directly on the waveform using additive noise extracted from the MUSAN database [32] with a random target signal-to-noise ratio ranging from 5 to 20 dB. The learning rate is set up by a cyclical scheduler [33], each cycle lasting for 1.5 epoch.

Since we address the problem in a multi-label classification fashion, multiple classes can be activated at the same time. For the reference turns, the SPEECH class was considered to be activated whenever one (or more) of the KCHI, CHI, FEM, MAL, or UNK class was activated. The UNK class (see Table 1) corresponds to cases when the human annotator could hear that the audio contained speech or vocalizations, without being able to identify the voice source. This class does contribute in activating the SPEECH class, but our model does not return a score for it.

2.5. Evaluation protocol

For all experiments, the neural network is trained for 10 epochs (approximately 2400 hours of audio) on the training set. The development set is used to choose the actual epoch and thresholds \( \{\sigma_j\}_{j=1}^K \) that maximizes the average F-measure between precision and recall across classes.

We report both the in-domain performance (computed on the test set of BabyTrain) and the out-of-domain performance (computed on the hold-out set, ACLEW-Random). We compare our model with the LENA system on the hold-out set.

3. Results

We evaluate two different approaches, one consisting of 5 models trained separately for each of the class (referred as binary), and one consisting of a single model trained jointly on all the classes (referred as multitask). At first, both in the binary and the multitask scenario, architectures shared the same set of hyper-parameters. Only the dimension of the output layer differed. Results indicated that multitask approaches were significantly better than binary ones, which seems to show that sharing weights during training helps better learn the boundaries between the different classes.

To further improve the performance of our model, we tried

wide range of environments, conditions and languages and have been collected and annotated by numerous field researchers.

We will refer to this data set as BabyTrain, of which a broad description is given in Table 1.

We split the BabyTrain data set into a training, development and test sets, containing approximately 60%, 20% and 20% of the audio duration respectively. We applied this split such that files associated to a given key child were included in only one of the three sets, splitting children up within each of the 8 corpora of BabyTrain. The only exception was WAR2, too small to be divided, and therefore put in the training set in its entirety.

In order to ensure that our models generalize well enough to unseen data, and to compare the performance with the LENA system, we kept the ACLEW-Random as a hold-out data set.
multiple sets of hyper-parameters (varying the number of filters, the number of LSTM and FF layers, and their size). However, no significant differences have been observed among the different architectures. The retained architecture consists of 256 filters of length $L=251$ samples, 3 LSTM layers of size 128, and 2 FF layers of size 128.

Finally, removing Paido from the training and development set led to improvements on the other test domains, as well as the hold-out set, while the performance on the Paido domain remained high. Indeed, we observed a F-measure of 99 on the KCHI class for the model trained with Paido as compared to 89 for the model trained without it. This difference can be explained by a higher amount of false alarms returned by the model trained without it. The Paido domain is quite far from our target domain since it consists of laboratory recordings of words in isolation spoken by children, and thus it is reasonable to think that removing it leads to better models.

3.1. In-domain performance
Since LENA can only be evaluated in data collected exclusively with the LENA recording device and BabyTrain contains a mixture of devices, we do not report on LENA in-domain performance. Additionally, comparing performance on a domain that would have been seen during the training by our model but not by LENA would have unfairly advantaged us.

Table 2 shows results in terms of F-measure between precision and recall on the test set for each of the 5 classes. The best performance is obtained for the KCHI, FEM, and SPEECH classes, which correspond to the 3 classes that are the most present in BabyTrain (Table 1). Performance is lower for the OCH class and MAL classes, with an F-measure of 25.6 and 22.5 on the KCHI and MAL classes, respectively.

Table 3: Performance on the hold-out data set in terms of F-measure between precision and recall. "Ave." column represents the F-measure averaged across the 5 classes. The hold-out data set has never been seen during the training, neither by LENA, nor by our model.

Turning to the comparison with LENA, both the LENA model and our model show lower performance for the rarer OCH and MAL classes. Our model outperforms the LENA model by a large margin. We observe an absolute improvement in terms of F-measure of 13.8 on the KCHI class, 4.6 on the OCH class, and 5.6 on the MAL class, 20.8 on the FEM class, and 8.1 on the SPEECH class. This leads to an absolute improvement of 10.6 in terms of F-measure averaged across the 5 classes.

4. Reproducible research
All the code has been implemented using pyannote.audio [34], a python open-source toolkit for speaker diarization. Our own code, easy-to-use scripts to apply the pretrained model can be found on our GitHub repository 1, which also includes confusion matrices and a more extensive comparison with LENA. As soon as required agreements will be obtained, we plan to facilitate access to the data by hosting them on Homebank.

5. Conclusion
In this paper, we gathered recordings drawn from diverse child-centered corpora that are known to be amongst the most challenging audio files to process, and proposed an open-source speech processing model that classifies audio segments into key child vocalizations, other children vocalizations, adult male speech, and adult female speech. We compared our approach with a homologous system, the LENA software, which has been used in numerous child language studies. Our model outperforms LENA by a large margin and will, we hope, lead to more accurate observations of early linguistic environments. Our work is part of an effort to strengthen collaborations between the speech processing and the child language acquisition communities. The latter have provided data as that used here, as well as interesting challenges [6, 7]. Our paper is an example of the speech processing community returning the favor by providing robust models that can handle spontaneous conversations in real-world settings.

1https://github.com/MarvinLvn/voice-type-classifier
6. References


