Property-Aware Multi-Speaker Data Simulation: A Probabilistic Modelling Technique for Synthetic Data Generation

Tae Jin Park, He Huang, Coleman Hooper, Nithin Koluguri, Kunal Dhawan, Ante Jukić, Jagadeesh Balam and Boris Ginsburg

NVIDIA, Santa Clara, USA

{taejin, heh, chooper, nkoluguri, kdhawan, ajukic, jbalam, bginsburg}@nvidia.com

Abstract

We introduce a sophisticated multi-speaker speech data simulator, specifically engineered to generate multi-speaker speech recordings. A notable feature of this simulator is its capacity to modulate the distribution of silence and overlap via the adjustment of statistical parameters. This capability offers a tailored training environment for developing neural models suited for speaker diarization and voice activity detection. The acquisition of substantial datasets for speaker diarization often presents a significant challenge, particularly in multi-speaker scenarios. Furthermore, the precise time stamp annotation of speech data is a critical factor for training both speaker diarization and voice activity detection. Our proposed multi-speaker simulator tackles these problems by generating large-scale audio mixtures that maintain statistical properties closely aligned with the input parameters. We demonstrate that the proposed multi-speaker simulator generates audio mixtures with statistical properties that closely align with the input parameters derived from real-world statistics. Additionally, we present the effectiveness of speaker diarization and voice activity detection models, which have been trained exclusively on the generated simulated datasets.

Index Terms: speaker diarization, data simulator, multi-speaker data simulation

1. Introduction

The evolution of deep neural network models within the realm of speech signal processing has significantly enhanced the performance and precision of the machine learning systems [1]. These advances have facilitated an end-to-end training approach, allowing the entire model to be optimized, transforming raw audio input into meaningful labels. However, achieving competitive accuracies with these neural models depends on the procurement of a substantial amount of data. This data is integral to ensuring generalizability and improving accuracy.

Obtaining sufficient training data in certain domains poses a significant challenge due to an array of factors. In speech signal processing, the challenges are concentrated on privacy concerns, data-imbalance issues, limited availability and the financial cost of data collection. The task becomes even more demanding when it involves speaker diarization. This increased difficulty is primarily because speaker diarization requires a complex dataset with multiple speakers, embodying a broad range of variabilities. These variabilities encompass aspects such as gender, acoustic conditions, and conversation types. Hence, the development and optimization of effective deep neural network models for speech signal processing, particularly speaker diarization, hinges on overcoming these challenges related to data collection.

In response to the challenge of data scarcity in specific fields, the machine learning community has adopted synthetic data, which mitigates the aforementioned issues to a certain degree. In order for synthetic data to be effective, the generated data should capture the characteristics and patterns of real-world data (realism) while maintaining a broad range of variabilities (diversity). Also, accurate and consistent labeling of the synthetic dataset is essential. Additionally, in speaker diarization or Voice Activity Detection (VAD), the diversity of speakers, sentence length, and frequency of speaker turns in conversations should be well balanced, mirroring real-world data.

Until now, in the fields of speech recognition and speaker diarization, most published articles have focused on data augmentation techniques, such as the widely used SpecAugment [2] or the data augmentation speech recognition toolkit [3, 4]. There exist simulation tools (e.g., one featured in [5]) initially developed for source separation but often utilized in training speaker diarization systems [5, 6, 7]. Recently, a multi-speaker data simulator for speaker end-to-end speaker diarization also appeared in [8, 9], which tries to create mixtures that resemble the pauses and overlaps of the real-world audio recordings. While the data simulation techniques introduced in [6, 9] serve their purpose very well, these data simulation techniques tend to employ a range of parameters which do not explicitly correlate with specific properties such as pauses and overlaps within the resulting simulated speech recordings. Consequently, even though the previously proposed simulation systems accept numerous parameters, their lack of control over the generated signal could lead to unpredictability in the amount of silence and overlap.

In this work, we introduce a dynamic sampling technique that constantly reflects the discrepancy between the generated data and the targeted amount of overlap speech and silence employing probabilistic models for precision and control. We refer to such feature as “Property-aware simulation”. As illustrated in Fig. 1, the proposed multi-speaker data simulator takes speech signal and its alignment (time stamps) and blends these signals to simulate multi-speaker audio recordings. Herein, we elaborate on the guiding principles for our data simulation systems:

- The simulated sessions are designed to incorporate the required amount of silence, overlap, and sentence length based on statistical analysis.
• The speech signal generated by the simulation system exhibits a significant level of variability across sessions, including overlap ratio, silence ratio, and average sentence length.
• The simulation system employs parallel processing techniques, leveraging multiple graphics processing units (GPUs), enabling large-scale data generation at higher speed.
• The implementation of the data simulator is open-source and publicly available online.¹

2. System description

2.1. Major parameters

A flow diagram of the proposed system is shown in Fig. 2. In the following sections we describe the main parameters and implementation details. Note that the following parameters are the most crucial subset of parameters that are determined before starting data simulation:

• Session length $L_S$: A floating point number that determines the total duration of the created session in seconds.
• Number of sessions $N_S$: An integer to determine the number of session to be simulated, so that the total duration of the generated data is $L_S \cdot N_S$ seconds.
• Number of speakers $N_{spks}$: An integer number that determines how many speakers in a session.
• Turn Probability $p_{turn}$: A floating point number that determines how many speakers in a session.
• Overlap ratio mean $\mu_o$ and variance $\sigma_o^2$: Parameters that determine the distribution of overlap.
• Silence ratio mean $\mu_s$ and variance $\sigma_s^2$: Parameters that determine the distribution of silence.

The following are random variables that are created at each session:

• Sentence length $s_i$: determines how many words are included in a newly added utterance (also referred to as a sentence).
• Silence length $\bar{m}_s$: determines the duration between sentences.
• Overlap length $m_o$: determines how much portion of speech is overlapped with the following speech segment.

2.2. Session Parameter Sampling

The following variables are sampled during the very first step named “Sample Session Parameters” in Fig. 2.

- Session random seed sampling: Set a random seed value which would be used to create a reproducible simulation environment.
- Setting Session Silence and Overlap Mean: To control the amount of silence and overlap in a session, we can set the mean values for these parameters using the following equation, which describes the method of moment estimates [10] for a Beta distribution:²

\[
\alpha_{(o,s)} = \frac{\mu_{(o,s)}^2 \cdot (1 - \mu_{(o,s)})}{\sigma_{(o,s)}^2} - \mu_{(o,s)} \quad (1)
\]

\[
\beta_{(o,s)} = \frac{\mu_{(o,s)} \cdot (1 - \mu_{(o,s)})^2}{\sigma_{(o,s)}^2} - (1 - \mu_{(o,s)}) \quad (2)
\]

Here, $\mu$ represents the mean ratio of silence or overlap and $\sigma$ represents its variance. These parameters are fed by the user to control the overall silence and overlap ratio. To ensure that $\alpha_{(o,s)}>0$ and $\beta_{(o,s)}>0$, the input mean and variance values should be within the following range:

\[
\begin{cases}
0 < \mu_{(o,s)} < 1 \\
0 < \sigma_{(o,s)}^2 \leq \mu_{(o,s)} \cdot (1 - \mu_{(o,s)})
\end{cases} \quad (3)
\]

We can then sample the session silence mean $X_{\mu_s}$, and session overlap mean $X_{\mu_o}$ from the Beta distribution, as follows:

\[
X_{\mu_s} \sim \text{Beta}(\alpha_s, \beta_s), \quad (4)
\]

\[
X_{\mu_o} \sim \text{Beta}(\alpha_o, \beta_o). \quad (5)
\]

Here, $\alpha_s$ and $\beta_s$ are based on the mean $\mu_s$ and variance $\sigma_s$ for the silence ratio in a session, while $\alpha_o$ and $\beta_o$ are based on the mean $\mu_o$, and variance $\sigma_o$, for the overlap speech ratio in a session. By setting the session silence and overlap mean values in this way, we can control the amount of silence and overlap in a session, which follows Beta distribution.

2.3. Sampling Routine for Data simulation

The following provides a description of each step involved in generating a simulated multi-speaker audio recording. In this section: $n_s$ denotes the current sample count; $s_{spks}$ the speaker index, and $L_S$ the running length of the audio signal thus far.

2.3.1. Data synthesis loop

As described in the Algorithm 1, the running length of the current session $L_S$ is monitored at every loop and while the condition $L_S < L_S$ is held, the sampling process is continued until the running length $L_S$ exceeds the desired length $L_S$.

2.3.2. Sample Speaker ID

The turn probability $p_{turn}$ is compared with a value drawn from a uniform distribution.

\[
U(0, 1) < p_{turn} \quad (6)
\]

If the sampled value is less than the $p_{turn}$ value, a randomly chosen speaker is selected from the pre-determined speaker group, for example $S_{spks} = \{s_1, s_2, \ldots , s_{N_{spks}}\}$.

¹https://github.com/NVIDIA/NeMo/main/tools/speech_data_simulator

²Beta distribution is employed due to its compatibility with the range of overlap and silence ratios, which fall within its support of $[0, 1]$, and its capacity to model skewed distributions[11].
2.3.3. Build Sentence
The parameter $s_l$, which represents sentence length, is assumed to follow a negative binomial distribution. This approach is based on the probabilistic model for word-level sentence length (measured in words) of human language as detailed in [12].

$$s_l \sim NB(k_w, p_w)$$

$$P_{NB}(X = k_w) = \left(\frac{X + k_w - 1}{k_w - 1}\right) p_w^k (1 - p_w)^X \tag{7}$$

Based on the sentence length $s_l$ and speaker (also referred as speaker turn) $s_{spk}$, we randomly select the given number of words from the forced-alignment data. This process is denoted as $\text{BUILD\ SENTENCE()}$ function in the Algorithm 1.

$$\hat{L}_{\text{spch}}, \hat{L}_{\text{sil}} = \text{BUILD\ SENTENCE}(s_l, s_{spk}). \tag{9}$$

2.3.4. Overlap-Silence Selector
In this step, the data-simulator system compares the current silence ratio to the current overlap ratio. Thus, at each utterance loop, it switches to either silence or overlap mode according to the amount of the gap between current ratio and session mean in configurations.

$$\Delta S = \frac{L_{\text{sil}} - \mu_s}{L_S} \tag{10}$$

$$\Delta O = \frac{\hat{O}_{\text{spch}} - \mu_o}{\hat{L}_{\text{spch}}} \tag{11}$$

We employ two different quantities: silence discrepancy $\Delta S$ represents the gap between desired silence time and the current silence time and overlap discrepancy $\Delta O$ represents which is the gap between desired overlap speech and the current overlap speech time. We choose whichever is smaller than other.

2.3.5. Estimating the Required Overlap Amount
Overlap $\bar{m}_o$ is calculated so that the newly added amount of overlap matches the expected amount of $X_{\mu_o}$.

$$X_{\mu_o} = \bar{m}_o + \frac{\hat{O}_{\text{spch}}}{\hat{L}_{\text{spch}}} - \bar{m}_o \tag{12}$$

Afterwards, we solve for $\bar{m}_o$ and assign it the value derived from the following equation:

$$\bar{m}_o \leftarrow \frac{X_{\mu_o} \hat{L}_{\text{spch}} - \hat{O}_{\text{spch}}}{X_{\mu_o} + 1} \tag{13}$$

2.3.6. Estimating the Required Silence Amount
We set up an equation that matches the expected amount of silence after adding the silence (denoted by $\bar{m}_s$) with the sampled mean $X_{\mu_s}$, as follows:

$$X_{\mu_s} = \frac{\bar{m}_s + \hat{L}_{\text{sil}}}{\bar{m}_s + \hat{L}_S} \tag{14}$$

Solve for $\bar{m}_s$, then we assign the following value:

$$\bar{m}_s \leftarrow \frac{\hat{L}_{\text{sil}} - X_{\mu_s} \hat{L}_S}{X_{\mu_s} - 1} \tag{15}$$

Algorithm 1 Dialogue Simulation

**Require:** $L_S, \sigma_s^2, \mu_s, \mu_s, \delta_s, \sigma_d^2, \sigma_o^2 \in \mathbb{R}$ and $N_{spk} \in \mathbb{N}$

$p \in \{0, 1\}$ \hspace{1cm} $\mu_d, \mu_o, \mu_s, \sigma_d^2, \sigma_o^2, \sigma_s^2 \in [0, 1]$

$(\alpha_s, \beta_s) \leftarrow (\frac{1}{\sigma_s^2}, \frac{1}{\sigma_s^2}) - (1 - \mu_s)$

$(\alpha_o, \beta_o) \leftarrow (\frac{1}{\sigma_o^2}, \frac{1}{\sigma_o^2}) - (1 - \mu_o)$

$X_{\mu_o} \sim \text{Beta}(\alpha_o, \beta_o)$ \hspace{1cm} Sample session silence rate mean

$X_{\mu_o} \sim \text{Beta}(\alpha_s, \beta_s)$ \hspace{1cm} Sample session overlap rate mean

**while** $L_S < L_S$ **do**

**if** $\hat{L}_{\text{sil}} < L_S$ **then**

$s_{spk} = \text{GET\ NEXT\ SPEAKER}(\hat{S}_{spk}, s_{spk})$

**end if**

**if** $s_l \sim NB(k_w, p_w)$

$\hat{L}_{\text{spch}}, \hat{L}_{\text{sil}} \leftarrow \text{BUILD\ SENTENCE}(s_l, s_{spk})$

$\Delta S = \frac{L_{\text{sil}} - \mu_s}{L_S}$ \hspace{1cm} Silence deficiency

$\Delta O = \frac{\hat{O}_{\text{spch}} - \mu_o}{\hat{L}_{\text{spch}}}$ \hspace{1cm} Overlap deficiency

**if** $\Delta S \leq \Delta O$ **then**

$$\bar{m}_o \leftarrow \frac{\hat{m}_o \hat{L}_{\text{spch}} - \hat{L}_{\text{sil}}}{\hat{m}_o \hat{L}_{\text{spch}}} \tag{16}$$

$$\bar{m}_s \leftarrow \frac{\hat{m}_s \hat{L}_{\text{sil}} + \hat{O}_{\text{spch}}}{\hat{m}_s \hat{L}_{\text{sil}}} \tag{17}$$

**ADD\ SENTENCE(\hat{s}_{\text{sil}}, 0)**

**end if**

**else if** $\Delta S > \Delta O$ **then**

$$\bar{m}_o \leftarrow \frac{\hat{m}_o \hat{L}_{\text{spch}} + \hat{O}_{\text{spch}}}{\hat{m}_o \hat{L}_{\text{spch}}} \tag{18}$$

$$\bar{m}_s \leftarrow \frac{\hat{m}_s \hat{L}_{\text{sil}} - \hat{L}_{\text{sil}}}{\hat{m}_s \hat{L}_{\text{sil}}}$$

**ADD\ SENTENCE(0, \hat{s}_{\text{sil}})**

**end if**

**end while**

2.3.7. Sampling overlap and silence amount
For both silence and overlap cases, we employ gamma distribution since gamma distribution is continuous version of negative binomial distribution that is used to model sentence length in [12] Thus, we model the distribution of the two continuous quantity, silence and overlap length, as following equations:

$$X = \frac{\bar{m}_s^2}{\sigma_s^2} \tag{16}$$

$$\bar{m}_o \sim \text{Gamma}(k, \theta) \tag{17}$$

where $X_{\Delta_t}$ is the sampled silence amount $s_{\Delta_t}$ in silence case and overlap amount $s_{\Delta_t}$ in overlap case.

3. Experimental Results

3.1. Data simulation test
In this section, we test whether the proposed data simulator can simulate the multi speaker data with the given parameters. To check whether the simulator generates data which has the distribution we intended to create, we compare the simulated data with the statistics extracted from real-world datasets. The overlap mean, overlap mean variance, silence mean and silence mean var values are collected from the real-world datasets and fed to the simulator.

Table 1 presents a quantitative comparison of observed values derived from both the simulated and real-world datasets for train split of AMI(MixHeadSet) [13] and CallHome American English Speech (CHAES) [14], highlighting key metrics such
Table 1: Simulated vs. Real-world Dataset Statistics

<table>
<thead>
<tr>
<th>Dataset Type</th>
<th>Mean</th>
<th>Var.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH. Simul. observed sil. ratio</td>
<td>0.1409</td>
<td>0.0045</td>
</tr>
<tr>
<td>CH. Simul. observed ovl. ratio</td>
<td>0.0759</td>
<td>0.0019</td>
</tr>
<tr>
<td>CHAES real-world sil. ratio</td>
<td>0.1473</td>
<td>0.0061</td>
</tr>
<tr>
<td>CHAES real-world ovl. ratio</td>
<td>0.0754</td>
<td>0.0020</td>
</tr>
<tr>
<td>AMI Simul. observed sil. ratio</td>
<td>0.1804</td>
<td>0.0077</td>
</tr>
<tr>
<td>AMI real-world sil. ratio</td>
<td>0.1814</td>
<td>0.0081</td>
</tr>
<tr>
<td>AMI Simul. observed ovl. ratio</td>
<td>0.1711</td>
<td>0.0092</td>
</tr>
<tr>
<td>AMI real-world ovl. ratio</td>
<td>0.1473</td>
<td>0.0047</td>
</tr>
</tbody>
</table>

Table 2: Evaluation of models on different parameters

(a) AUROC for VAD task

<table>
<thead>
<tr>
<th>Training Data</th>
<th>Dataset Split</th>
<th>DH3</th>
<th>VoxConv</th>
<th>AMI</th>
<th>CH109</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>dev</td>
<td>dev</td>
<td>eval</td>
<td>-</td>
</tr>
<tr>
<td>D1, μs=0.5</td>
<td>dev</td>
<td>87.71</td>
<td>96.15</td>
<td>95.7</td>
<td>88.07</td>
</tr>
<tr>
<td>D1, μs=0.3</td>
<td>test</td>
<td>89.83</td>
<td>96.19</td>
<td>94.69</td>
<td>91.04</td>
</tr>
<tr>
<td>+ Gain. Aug.</td>
<td>eval</td>
<td>93.76</td>
<td>96.02</td>
<td>96.55</td>
<td>88.73</td>
</tr>
<tr>
<td>+ D2 + Noise. Aug.</td>
<td></td>
<td>93.96</td>
<td>97.42</td>
<td>96.04</td>
<td>92.43</td>
</tr>
</tbody>
</table>

(b) DER(%) on Diarization Datasets

<table>
<thead>
<tr>
<th>Training Data</th>
<th>Dataset Split</th>
<th>DH3</th>
<th>VoxConv</th>
<th>AMI</th>
<th>CH109</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>eval</td>
<td>test</td>
<td>eval</td>
<td>-</td>
</tr>
<tr>
<td>LibriVox-3Kh μs=0.07</td>
<td></td>
<td>14.49</td>
<td>6.01</td>
<td>15.96</td>
<td>9.94</td>
</tr>
<tr>
<td>LibriVox-3Kh μs=0.15</td>
<td></td>
<td>14.38</td>
<td>5.72</td>
<td>15.89</td>
<td>10.03</td>
</tr>
</tbody>
</table>

(c) DER(%) on CHiME7 Datasets

<table>
<thead>
<tr>
<th>Training Data</th>
<th>Dataset Split</th>
<th>Chime6</th>
<th>Dico</th>
<th>Mixer6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>dev</td>
<td>dev</td>
<td>dev</td>
</tr>
<tr>
<td>LibriVox-3Kh μs=0.07</td>
<td></td>
<td>45.01</td>
<td>32.50</td>
<td>17.35</td>
</tr>
<tr>
<td>LibriVox-3Kh μs=0.15</td>
<td></td>
<td>44.37</td>
<td>31.07</td>
<td>17.13</td>
</tr>
</tbody>
</table>

Figure 3: Histograms: Real-World (Magenta) vs. Simulated (Blue) Data; Overlaps in Purple.

3 https://catalog.ngc.nvidia.com/orgs/nvidia/teams/nemo/models/vad_multilingual_frame_marblenet

and end of each segment, can markedly degrade the performance of VAD. This issue is exacerbated by data augmentation, as the model is then trained with the added noise at the boundaries of each segment. Secondly, gain perturbation is a necessary consideration as the VAD model frequently overlooks low-volume speech signals. To mitigate this, the model should be trained with substantial variation in gain during the creation of audio mixtures. Lastly, the addition of overlapping speech is also essential for enhancing performance, as overlapping speech can lead to an increase in missed detections.

3.3. Speaker Diarization Model

As in the previous section, we train a modified speaker diarization model, based on [24], alongside the speaker embedding model from [25]. The experiment utilizes 1k hours of LibriSpeech and 2k hours of VoxCeleb 1 and 2 [26]. We use the same type of time stamps and random word-level alignment as in the Fisher dataset for Voice Activity Detection (VAD). For diarization evaluation, we use the VAD model from Table 2a. Diarization error rate (DER) is calculated using a 0.25 sec collar, with overlap considered. DER is assessed on DIHARD3, VoxConverse-3 [27], AMI eval(test)-sets, and 2-speaker CHAES subset, CH109. Tables 2b and 2c show performance variations with different synthetic dataset settings.

4. Conclusions

In this paper, we introduce a property-aware data simulator capable of reflecting statistics provided by the user or extracted from real-world data. The proposed data simulator utilizes an online sampling technique, allowing the system to generate a predetermined quantity of silence and overlap speech while adhering to the given probability distributions. Consequently, the generated dataset can be leveraged to train VAD models and speaker diarization models, providing highly accurate ground-truth timestamps, which is a critical element for both speech activity detection and speaker diarization. Potential future research could involve adapting this system for online generation, whereby users could supply a source dataset and generate the training dataset on-the-fly. We anticipate that the proposed data simulator will be adopted by the speech signal processing community for training neural models related to speech signals, which necessitate accurate ground truth timestamps and highly customizable speech training data.
5. References


