Modelling Turn-taking in Multispeaker Parties for Realistic Data Simulation

Jack Deadman, Jon Barker
Department of Computer Science, University of Sheffield, UK
jdeadman1@sheffield.ac.uk, j.p.barker@sheffield.ac.uk

Abstract
Simulation plays a crucial role in developing components of automatic speech recognition systems such as enhancement and diarization. In source separation and target-speaker extraction, datasets with high degrees of temporal overlap are used both in training and evaluation. However, this contrasts with the fact that people tend to avoid such overlap in real conversations. It is well known that artifacts introduced from pre-processing with no overlapping speech can be detrimental to recognition performance. This work proposes a finite-state based generative method trained on timing information in speech corpora, which leads to two main contributions. First, a method for generating arbitrary large datasets which follow desired statistics of real parties. Second, features extracted from the models are shown to have a correlation with speaker extraction performance. This leads to the contribution of quantifying how much difficulty in a mixture is due to turn-taking, factoring out other complexities in the signal. Models which treat speakers as independent produce poor generation and representation results. We improve upon this by proposing models which have states conditioned on whether another person is speaking.

Index Terms: turn-taking, representations, target speaker extraction

1. Introduction
Automatic speech recognition (ASR) is a challenging task with numerous factors contributing to its difficulty, with recognising multi-party conversations in real, noisy environments being one of the most difficult scenarios. Recent ASR challenges have shown that much work is left to be done [1, 2]. It is therefore vital to break down the factors that are contributing to the difficulty. In recent work, we explored the impact of the angular separation between speakers [3] and the microphone distance [4] through analysing videos in the CHiME-5 dataset [1]. In this work, we extend this analysis by looking at speakers' temporal behaviour, i.e., their turn-taking behaviour.

Generating realistically overlapped speech data is crucial for the development of better conversational ASR systems. There has been much recent work in speech separation (a crucial ASR component) but it has mainly focused on highly-overlapped mixtures [5, 6, 7]. This data poorly models the real challenges. For example, applying separation techniques can be detrimental to ASR if the models attempt to extract more sources than those that are present [8], hence good estimates of the number of active speakers are needed. However, this is trivial in fully-overlapped cases. More recently, attention has turned to sparse versions of commonly used datasets [9] that use a parameter to govern the amount of overlap when creating mixtures. This is a step towards creating more realistic simulations but still fails to model the complexity of real conversations.

The simulated dataset most closely capturing real conversation dynamics is LibriCSS [10], which creates long-form parties containing many utterances from several speakers. It can be processed into segmented mixtures, directly used for tasks such as diarization, or used in experiments with enhancement systems requiring long context windows [11]. However, LibriCSS contains just 10 hours of data, which are recordings of audio played back in a room. Therefore, the dataset is only appropriate for evaluation rather than training. The current work greatly extends this, through analysis of the 50 hours of conversations recorded from 20 parties in the CHiME-5 dataset [1].

The use of the generative overlapped speech simulation techniques presented in this paper extend beyond ASR to the task of diarization. The recent trends in diarization has been towards jointly optimising voice activity detection and segmentation [12], and towards end-to-end systems [13]. If such systems are to be trained using simulated datasets, realistic turn-taking behaviour is essential.

In this work, we establish a framework for analysing the turn-taking behaviour of people in real life recordings and establish a method for extracting how much of the difficulty of a recording can be attributed to the turn-taking behaviour. In Section 2, we discuss the complexity of human turn-taking. In Section 3, we introduce a framework for modelling the turn-taking behaviour of multi-person “parties” using a simple finite-state representation. Section 4 shows that representations can be created that can be used to characterise recordings in real datasets with interpretable meanings. These representations are then evaluated in Section 5 in the context of target-speaker extraction, where the representations can predict the difficulty of a mixture purely based on turn-taking behaviour. Finally, Section 6 concludes our findings.

2. Background
Modelling the turn-taking behaviour of humans is a well-established research field [14, 15]. The behaviour of people changes depending on numerous factors such as their environment, who they are talking to and whether the conversation is physical or virtual. Modelling turn-taking is also a multi-modal activity, where cues are not always verbal. Gazes are often used to select the next speaker, or head nods are used to indicate confirmation and encourage the speaker to continue talking. Much work has been done towards predicting how the turns will develop, allowing for conversational agents to naturally take their turn and add backchannels without the perception of interruption [16].

Instead of predicting this behaviour, we aim to observe real turn-taking and generate more data following the observed distribution. Whilst prediction will require observations of the speaker (e.g., a video recording), a generator will not require a recording. The generator can model characteristics of this behaviour by looking at the resulting turns that were taken even if that turn occurs due to non-verbal behaviour.

When generating parties for larger mixtures of speech, simplistic approaches are used, such as in [13], where speakers are
treated independently. This completely neglects how speakers interact with each other. Agent-based models for generating has been explored [17], where participants are parameterised by engineered features such as talkativeness (desire to talk), confidence (persistence to talk when others are talking), verbosity (desire to continue talking). Motivated by this, we aim to build a structure that can learn these parameters from the data.

3. Framework for Modelling Turns

In this section, we introduce a framework for modelling turn-taking solely based on utterance timings, i.e., looking at the turn-taking behaviour of speakers while ignoring any linguistic and acoustic cues that will come with these signals. This simple approach allows for easily computed models that can be used across a wide array of datasets, i.e., not all datasets provide fully transcribed text but most provide endpointing.

With a transcript providing start and endpoints of utterances in a party, a discrete representation of the turn-taking behaviour is achieved through Markov models. The state duration distributions are estimated by, e.g., Maximum Likelihood Estimation. The models can be trained by observing real parties and counting the transitions. The transitions weights are estimated by, e.g., the number of speakers. The value \( y^i_l = 1 \) if speaker \( k \) is speaking at time frame \( l \) and \( y^i_l = 0 \) otherwise.

#### 3.1. Finite-state Model Formulation

We want to build a vector representation \( \phi \) to summarise the speaker activity matrix, \( \mathbf{Y}' \). We propose to achieve this through training a generative model \( M_p \) and computing features from the learnt parameters \( \Theta \) i.e., \( f(\Theta) = \phi \). Finite-state models have been ubiquitous in speech recognition history [18] with their relatively simple design and well-defined mathematical properties. They have also been shown to be useful for modelling and predicting turn-taking [19]. Using the parameters of hidden-Markov models has been well established in many tasks requiring speaker representations e.g., speaker recognition [20], speaker verification [21] and adaptation for robust speech recognition [22, 23].

We propose to model the behaviour of speakers as a series of observations \( y_l \) being generated by a model depending only on \( y_{l-1} \). This can be achieved through Markov models where states indicate the active speakers at that point in time. First, fully-connected Markov model, where every possible combination of speakers has a state, \( \mathbf{S}^{\text{full}} = \mathcal{P}(\mathcal{P}) \) and an \( |\mathcal{S}| \times |\mathcal{S}| \) transition matrix \( \mathbf{T}^{\text{full}} \) where \( T_{ij} \) is the probability of transitioning from state \( i \) to state \( j \). The full model requires many parameters \( O(2^K) \). We therefore also describe two further models which require fewer parameters \( O(K) \) by treating speakers as independent generators with their own Markov models. First, an independent model with a set of independent models \( \mathbf{S}^{\text{ind}} = \{s_1^{\text{ind}}, ..., s_K^{\text{ind}}\} \) where \( s_k^{\text{ind}} = \{\emptyset, \emptyset\} \), with the corresponding transition matrices for each of the sub-models \( \mathbf{T}^{\text{ind}} = \{\mathbf{T}_1^{\text{ind}}, ..., \mathbf{T}_K^{\text{ind}}\} \). Next, a competing speaker model where the sub-models have dependencies on other speakers, \( \mathbf{S}^{\text{comp}} = \mathcal{P}(\mathcal{P}_b, \emptyset) \) where \( \emptyset \) symbolises some other person speaking, the model also has independent transition matrices \( \mathbf{T}^{\text{comp}} = \{\mathbf{T}_1^{\text{comp}}, ..., \mathbf{T}_K^{\text{comp}}\} \).

The modelling power of all models can be further extended through fitting a time distribution \( P_s(D = d; \Theta_s) \) on each of the states \( s \) for a time \( d \), making the model semi-Markovian [24]. The time spent in the state is drawn from this distribution instead of being purely based on the previous state. A comparison of the different models is depicted in Figure 1 showing how the topology changes for modelling parties where \( K = 2 \).

#### 3.2. Training Models

The models can be trained by observing real parties and counting the transitions. The transitions weights are estimated by,

\[
T_{ij} = \frac{\eta_{ij}}{\sum_k \eta_{ik}}
\]

where \( \eta_{ij} \) is the count of the transitions from \( s_i \) to \( s_j \). This is the same process for all the models. The state duration distributions \( P_s(D = d; \Theta_s) \) are fit through counting the number of consecutive samples that have the same state, i.e., computing a duration. These data are then used to fit the duration models inside each state, e.g., using Maximum Likelihood Estimation.

#### 3.3. Sampling Models

To generate an activity matrix \( Y \), the model is initialised at a starting state (e.g., silent state), sampling the duration from the state distribution \( P_s(D; \Theta_s) \) and then sampling the next state based on the transition weight. The model then transitions to this state, and a new duration is sampled. The process is repeated until a desired number of samples has been generated. For the independent model, this process can be done in parallel across the sub-models.

---

1Python Package: https://github.com/jackdeadman/turn-taking

---

Figure 1: Depiction of the three model types. The states represent which speakers are actively talking when in that state. The speakers are given the labels ‘a’ and ‘b’ respectively. In the competing speaker model, \( \xi \) in the state set represents any other person is speaking. Only transitions between states which yield a single onset or offset change are shown.
Table 1: Comparison of the fully-connected model with the independent model. Computed using transcript, no voice activity detection. Monte Carlo estimation using 500 samples to estimate mean overlap for the models. $E[X]$ is the expected number of people talking at one time.

<table>
<thead>
<tr>
<th>Model</th>
<th># People Speaking</th>
<th>$E[X]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHiME-5</td>
<td>0% 22% 52% 20% 5%</td>
<td>1.15</td>
</tr>
<tr>
<td>Fully-connected</td>
<td>19% 42% 30% 8% 1%</td>
<td>1.30</td>
</tr>
<tr>
<td>Independent</td>
<td>6% 25% 37% 25% 6%</td>
<td>1.98</td>
</tr>
</tbody>
</table>

3.4. Model Comparison

We will briefly compare the independent speaker model and the fully-connected model with respect to generation power. Generating from the competing speaker model is not as straightforward. This is due to the competing model using sub-models which are sampled independently, but the data it generates does not have an independent meaning. The results in Table 1 show the distribution of the CHiME-5 overlap according to the transcript.\footnote{No voice activity detection used in this calculation.} For each session in the dataset, a $Y$ is computed using $f_s = 100$ and a Wald distribution [25] is used for the time distribution ($P_x$) in the states. A model is fit on the session, and then data is generated matching the original session length. The overlap statistics are then computed. The process is repeated 500 times in order to estimate the mean statistics of the overlap. The numbers reported in the table are the average of this across all the parties. From the table, we can see that the independent model produces a larger expected number of speakers than the original dataset. It is also mismatched with respect to each of the number of people speaking at one time. The fully-connected model still produces more overlapped data, but it is far closer to the training data. Although the model is not directly trained on the overlap distribution, it successfully approximates the degree of overlap observed in CHiME-5.

4. Party representations

Using the Markov-models we can compute features from the parameters learnt from the data. For the independent generators, features are computed for each model and then concatenated together, creating a larger representation.

4.1. Extracting Features from Models

Given a transition matrix for a Markov-model, a steady-state distribution can be computed to give the probability of being in each of the states [26],

$$\phi^{\text{trans}} = [Q_T Q^{-1}]^T \phi^{\text{state}}$$

where $Q = [T - I, 1]$ and $1$ is a column vector of ones with length $|S|$. From the steady-state, we can compute a distribution over the transitions to give the probability of a transition occurring,

$$\Phi^{\text{trans}}_{ij} = \phi^{\text{trans}}_i \cdot \phi^{\text{trans}}_j$$

which is flattened into a vector $\phi^{\text{trans}} \in \mathbb{R}^{|S|^2}$. Finally, using the state time distributions, an expected duration in the state can be computed,

4.2. Visualisation

To illustrate the embeddings, we will train models on a real-world dataset, CHiME-5. For visualisation, we train the fully-connected model on 40-minute chunks with an overlap of 5 minutes using a sample rate $f_s = 100$ and Wald for state distribution $P_x$. The speaker IDs are assigned based on activity, i.e., $\mathbb{P}_x$ is always the most active, and $\mathbb{P}_K$ is the least. This is done to mitigate the effect of speaker ordering being arbitrary but affecting the feature order. The plot in Figure 2 shows a t-SNE representation of $[\phi^{\text{state}}; \phi^{\text{trans}}; \phi^{\text{trans}}]^T$. The plot illustrates how sessions in CHiME-5 form clusters, i.e., the statistics of the parties remain fairly consistent. The size of the points in the plot are derived from their position in time in the party i.e., showcasing the movements of the points around the space over time. The plot illustrates some interesting behaviour amongst parties. S07 and S21 exhibit homogeneous behaviour throughout the parties, whilst other parties show a more varied distribution.

5. Evaluation

Now that a representation has been described, we will evaluate the representation with respect to the task of speaker extraction. That is, we aim to investigate how SI-SDR [27] changes with respect to the location of parties in the embedding space. Speaker extraction involves extracting the speech signal from the desired speaker in a mixture of zero or more other speakers and noise. The target speaker is indicated to a model through an enrollment. For this work, an utterance from a speaker is used that is not present in the mixture.

5.1. Target-speaker Extraction Model

For evaluation, time-domain Speakerbeam model [28] will be used. The model is based on Conv-Tasnet [29] with an additional component to learn a speaker embedding encoder jointly. The model is trained on Libri2Mix with WHAM noise [30]. The advantage of target-speaker extraction as an evaluation task is that the model works with mixtures with more than two speakers, which will be the case with the mixtures generated from the models.
Figure 3: t-SNE representation of learnt space against target-speaker extraction performance.

5.2. Data Generation

In total, 506 chunks are created using CHiME-5, and AMI [31] transcript data. Chunk lengths are again 40-minutes with a 5-minute overlap. The AMI data is reduced to only sessions containing four speakers and at least 40 minutes in duration. The fully-connected model is then fit on the chunks to generate the data, again a Wald distribution is used for states \(\mathbf{P}_s\). To generate the mixture data, first, the activity matrix \(\mathbf{Y}\) is generated from the fully-connected model, starting the model at the silent state \(\mathbf{ϕ}_0\). For this evaluation, \(\mathbf{Y}\) contains 2.5 hours of activity for each chunk being evaluated.

The large \(\mathbf{Y}\) matrix is broken into chunks 1 minute in length. Utterances from LibriSpeech (dev-clean) are then placed onto these segments such that they minimise the difference between LibriSpeech utterance duration and turn-generated duration. This process is repeated for all the chunks, and care is taken such that mixtures do not contain speakers overlapping themselves. If the error between the segments and LibriSpeech utterances is too large, it is discarded.\(^3\)

The 1-minute chunks are then broken into segmented mixtures. Each of the utterances in the chunk becomes a mixture where the duration is the length of the target signal.

This allows for the steady-state of this learnt turn-taking behaviour to be evaluated, i.e., the fully-connected model has been shown to accurately produce turn-taking behaviour with approximate statistics to the learnt data. Therefore generating from this model allows for statistically stationary turn-taking data to be generated.

5.3. Results

Table 2: Predicting SI-SDR using the model embeddings. Where \(\rho\) is Pearson Correlation and RMSE is the root mean square error in dB. \([\text{Si-SDR}]\) is using the mean of the training data as the prediction for the test samples.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Fully-connected (\rho)</th>
<th>Competing (\rho)</th>
<th>Independent (\rho)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSE</td>
<td>RMSE</td>
<td>RMSE</td>
<td>RMSE</td>
</tr>
<tr>
<td>(\text{[SI-SDR]})</td>
<td>-</td>
<td>2.17</td>
<td>2.17</td>
</tr>
<tr>
<td>(\mathbf{ϕ}<em>{\text{trans}} + \mathbf{ϕ}</em>{\text{dur}})</td>
<td>0.62</td>
<td>1.72</td>
<td>0.66</td>
</tr>
<tr>
<td>(\mathbf{ϕ}_{\text{trans}})</td>
<td>0.63</td>
<td>1.70</td>
<td>0.67</td>
</tr>
<tr>
<td>(\mathbf{ϕ}_{\text{dur}})</td>
<td>0.60</td>
<td>1.76</td>
<td>0.63</td>
</tr>
<tr>
<td>(\mathbf{ϕ}<em>{\text{trans}} + \mathbf{ϕ}</em>{\text{dur}})</td>
<td>0.63</td>
<td>1.70</td>
<td>0.66</td>
</tr>
<tr>
<td>(\mathbf{ϕ}_{\text{dur}})</td>
<td>0.60</td>
<td>1.76</td>
<td>0.63</td>
</tr>
<tr>
<td>(\mathbf{ϕ}_{\text{trans}})</td>
<td>0.60</td>
<td>1.76</td>
<td>0.62</td>
</tr>
<tr>
<td>(\mathbf{ϕ}_{\text{dur}})</td>
<td>0.53</td>
<td>1.85</td>
<td>0.40</td>
</tr>
</tbody>
</table>

\(^3\)If the average difference between utterance and segment is greater than 0.5 seconds.

Next, to evaluate the efficacy of the features, we use the application of predicting the SI-SDR scores. A Support Vector Regressor [32] is trained with a radial-basis function kernel using combinations of features presented in Section 4. The 506 samples are evaluated using a bootstrap cross-fold validation where 80% sample of sessions is used as training data, and the other 20% is used for the test. Where in each fold all the chunks in a session are evaluated after being trained on all other chunks, we then report Pearson correlation and root mean squared statistics in Table 2; this is the average across 100 repeats of this sampling procedure. The results show all the features provide information in predicting SI-SDR with a fairly strong correlation when using all the features together. The result indicates that the embeddings can provide a method of extracting how difficult a mixture can be when the generator is known. In addition to this, the table also shows the performance of the independent generator models at predicting the SI-SDR. The features of the fully-connected model are able to predict the performance of SI-SDR well. The competing model is able to surpass the performance of the fully-connecting model. This may be attributed to the fact that the full model is over parameterised for the embedding task but well suited for the generation task. The independent model lacks representation power. Each of the Markov-models inside of the independent models contain transitions which all have a weight \(T_{ij} = 1\), therefore \(\mathbf{ϕ}_{\text{trans}}\) and \(\mathbf{ϕ}_{\text{dur}}\) contain no information.

6. Conclusions

In this work, we have presented a simple Markov-model approach for generating arbitrary large datasets with statistical behaviour approximating the behaviour of people in real parties in terms of overlap. From these models, we showed features that can be computed to give a vector representation of a party. Using the CHiME-5 dataset, we illustrated that sessions within the dataset provided homogeneous behaviour and clustering.

These embeddings were then evaluated within the task of target-speaker extraction, where they were shown to have an interpretable meaning with respect to the performance of SI-SDR improvement. This was evaluated with respect to a prediction task based solely on the embeddings. We have shown that treating people as independent generators does not provide a realistic way of creating turn-taking behaviour, and it does not provide an adequate way of building a representation to predict difficulty. The fully-connected model was shown to be the best approach for generating turn-taking. The competing model was shown to surpass the performance of the fully-connected model as a representation.

269
7. References


