INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Discovering Language in Marmoset Vocalization

Sakshi Verma[1], *K L Prateek*[1], *Karthik Pandia*[1], *Nauman Dawalatabad*[1], *Rogier Landman*[2], *Jitendra Sharma*[2], *Mriganka Sur*[2], *Hema A Murthy*[1]

[1]Indian Institute of Technology Madras, India
[2]Massachusetts Institute of Technology, Cambridge, USA

[1]{sakshiv, prateekk, pandia, nauman, hema}@cse.iitm.ac.in,
[2]{landman, jeetu, msur}@mit.edu

Abstract

Various studies suggest that marmosets (*Callithrix jacchus*) show behavior similar to that of humans in many aspects. Analyzing their calls would not only enable us to better understand these species but would also give insights into the evolution of human languages and vocal tract. This paper describes a technique to discover the patterns in marmoset vocalization in an unsupervised fashion. The proposed unsupervised clustering approach operates in two stages. Initially, voice activity detection (VAD) is applied to remove silences and non-voiced regions from the audio. This is followed by a group-delay based segmentation on the voiced regions to obtain smaller segments. In the second stage, a two-tier clustering is performed on the segments obtained. Individual hidden Markov models (HMMs) are built for each of the segments using a *multiple frame size* and *multiple frame rate*. The HMMs are then clustered until each cluster is made up of a large number of segments. Once all the clusters get enough number of segments, one Gaussian mixture model (GMM) is built for each of the clusters. These clusters are then merged using Kullback-Leibler (KL) divergence. The algorithm converges to the total number of distinct sounds in the audio, as evidenced by listening tests.

Index Terms: clustering, group delay, segmentation, marmoset vocalization.

1. Introduction

The common marmoset (*Callithrix jacchus*) is a species of monkeys found in the Northeastern coast of Brazil. Marmosets have shown behavior close to that of humans in various aspects [1] and are commonly used for different neuroscience-related researches [2, 3]. They have a large repertoire of vocal behaviors. Also, the lifespan of this species is around 11.7 years, and they have good reproducibility. All these factors make marmosets an excellent model for studying vocal production and cognition [4].

A study showed that marmosets learn the language (calls) as they grow-up [5]. This study also shows how the type of calls change as marmosets grow up from infant to adult. In addition to learning language, the authors in [6] observed that the marmoset turn-taking skill, while they communicate, is a vocal behavior learned under the guidance of their parents. Marmosets use different kinds of calls to express anger, fear, aggressiveness, submissiveness and to alert other group members during threats [7]. Analyzing the calls made by marmosets would not only enable us to understand these species better but would also give insights into how human vocal tract and languages have evolved over time.

To understand their language the first step is to identify and classify different calls made by them. There have been some attempts made to classify the type of calls made by marmosets [4,7–9]. Most of the techniques use hand picked features for representation and labeled data to train classifiers. Authors in [7] have proposed a framework wherein features are chosen automatically. And yet, for the classification task, they have used different supervised classifiers like naive Bayes, support vector machine (SVM), decision trees, etc. All the approaches assume knowledge about the type of calls in the audio file.

Labeling the audio of marmosets vocalization requires skill and is a time-consuming task. Also, the recorded audio is usually noisy due to background noise, cage rattling, marmoset scratching the microphone collar, etc. Different marmosets produce different variants of the same sound. For example, the spectrograms of the same call from an infant and an adult look different [5]. Moreover, there are also some infant specific calls such as cry, compound cry, and call strings or babbling. There has been no attempt to segment and label the audio of marmoset vocalization automatically. Thus, in this work, apart from classifying different calls, we attempt to identify all the distinct calls present in the audio file in an unsupervised fashion. First, voice activity detection (VAD) is performed to remove silences and non-voice regions. Group delay based segmentation [10] is then applied on the output to obtain syllable-like segments. Individual hidden Markov models (HMMs) are built for each of the segments using a *multiple frame size* and *multiple frame rate*. This idea is borrowed from [11] where the objective was to discover the sounds of a language spoken by the human. The HMMs are then merged iteratively into clusters until each cluster is made up of a large number of segments. After the HMM-based clustering, one Gaussian Mixture Model (GMM) is trained for each of the clusters. These clusters are further merged using Kullback-Leibler (KL) divergence between GMMs. The algorithm converges to the total number of distinct sounds in the audio, as evidenced by listening tests.

The rest of the paper is organized as follows: Section 2 discusses the data collection and pre-processing details. Section 3 describes the proposed approach. Section 4 presents experimental results. Finally, Section 5 concludes the paper.

2. Data collection and pre-proccessing

Marmoset pairs in this study had lived together for at least six months at the time of experimentation. Two pairs of marmoset, namely, Enid and cricket, and Johnny and Baby Beans are used for experimentation. Vocalizations were recorded using commercially available, light weight voice recorders (Penictech, available on Amazon) mounted on a neck collar or a backpack. The voice recorder dimensions are 45x17x5 mm and the entire assembly (with backpack or collar) weighs ∼9 grams. The voice recorders have an omnidirectional microphone and a sam-

http://dx.doi.org/10.21437/Interspeech.2017-842

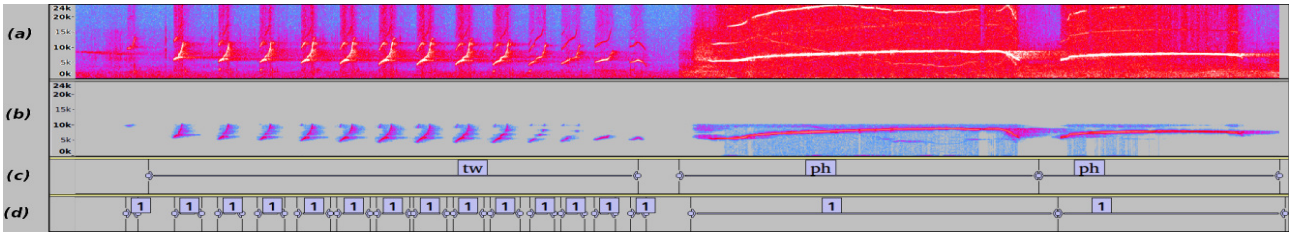

Figure 1: *A sample illustrating the pre-processing and segmentation of audio. 1(a) Spectrogram of original audio while 1(b) Spectrogram after pre-processing the audio. 1(c) Labeled ground-truth for the audio. Figure 1(d) output obtained from GD based segmentation algorithm.*

pling rate of 48 kHz. The data is stored on-board memory with an 8 GB capacity which allows for several hours of recording. The data is downloaded after that via an USB interface. The recordings were performed when the marmosets were habituated to wearing the collar/backpack. At the time of data collection, the recorders were placed on a selected pair, after gently holding the animals and were given treats to minimize stress. Following recording conditions were performed: both animals together; one animal (male or female) alternately taken out of the cage and placed in a transfer booth in the front of the home cage, where both animals had visual access; finally, both animals were together. Each epoch or condition lasted for 5 minutes, and there was a rest period of 10 minutes between each epoch. After completion of a recording session, the recorders were taken off, and the animals were again rewarded with bits of fruit/marshmallows or similar palatable treats. The .wav files of voice data were downloaded for post-processing in Audacity (http://www.audacityteam.org). The constant background noise present in the signal is removed by using the noise profile in the signal. Sounds are heavily clipped at some regions leading to artifacts in the spectrum as shown in the *ph* region in Figure 1(a). These artifacts are removed by applying a low-pass filter with a cut-off frequency of 15 kHz as shown in Figure 1(b). Spectrograms were aligned and were manually annotated to extract call start, end and type by experts for later comparison with automated classification and confirming the ground truth. The spectrograms of the dominant calls used for experiments are shown in Figure 3.

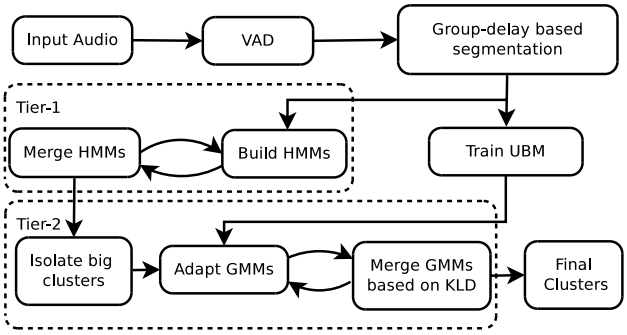

Figure 2: *Block diagram of complete clustering algorithm.*

3. Proposed approach

The flow chart of the proposed approach is shown in Figure 2. The input marmoset vocalization audio is first processed to obtain the vocalized region using a VAD algorithm. The obtained voiced segments are further segmented into finer segments using group delay based segmentation. These finer segments are

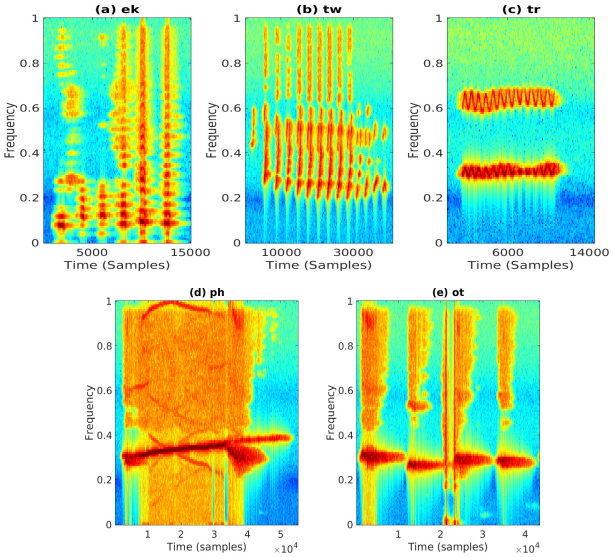

Figure 3: *Different calls made by Marmoset*

merged in *Tier-1* in an iterative manner by training and merging HMMs. This step yields clusters of finer segments. These clusters are then further merged in *Tier-2*, again in an iterative manner, to get larger clusters of distinct sounds. Group delay based unsupervised segmentation and a two-tier clustering algorithm are detailed in the subsequent subsections.

3.1. Unsupervised segmentation

As the task is to cluster similar sounds, first the sounds in the audio file must be segmented appropriately. Segmentation of sounds under noisy circumstances is a challenging task. Usually, marmoset vocalization is segmented based on intensity and duration criteria [4,8,9]. Using a single static threshold may not suffice to segment a long audio. Also, this type of thresholding is not adequate for the unsupervised clustering approach that we pursue in this paper.

To segment the audio, a bottom-up approach is followed right from VAD to segmentation. First, each frame under consideration is classified as either vocalized or non-vocalized using short-time energy (STE) and short-time zero crossing rate (SZCR). Then a duration constraint is applied to combine consecutive frames to obtain one segment (VAD segment) which is either vocalized or non-vocalized. Another duration constraint is set on the length of the VAD segments. On the vocalized regions (VAD segments), the finer segments to be clustered are obtained. This segmentation is performed using a group delay based processing on cepstrum obtained from the short-time energy. This algorithm has been used for segmenting human

speech into syllable-like units [10]. The high-resolution property of the group-delay helps in resolving the closely placed poles in a signal [12]. The group delay segmentation algorithm is as follows.

1. For the considered VAD segment, compute the STE function.
2. The STE function is then symmetrized to look like an arbitrary magnitude function.
3. Inverse Fourier transform of the assumed magnitude function is obtained which is called as root cepstrum. It has been shown that the causal portion of a root cepstrum is a minimum phase signal.
4. Group-delay of the root cepstrum is computed with appropriate window size.
5. The valleys in the obtained function correspond to the segment boundaries.

The segmentation output is illustrated in Figure 1(d). The sound *tw* gets segmented into a set of syllables. The obtained finer segments are grouped based on the similarity. This is performed by a two-tier unsupervised clustering.

3.2. Unsupervised clustering

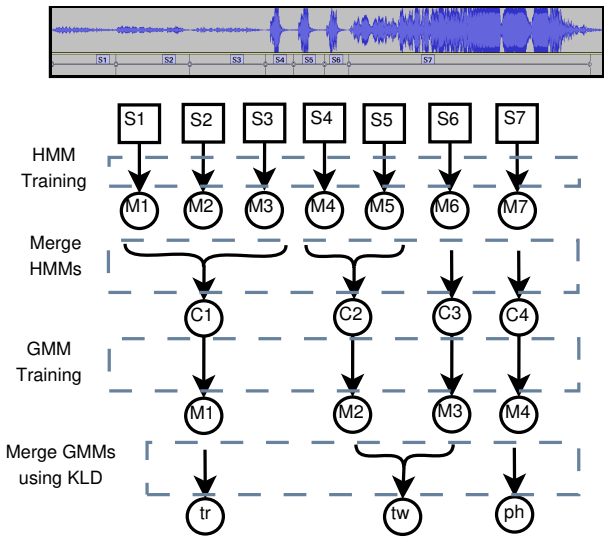

Figure 4: *A two-tier clustering algorithm*

Segments correspond to different sound units present in the audio are obtained. The objective is to group similar sound units into a cluster. For this, we propose a two-tier merging algorithm. In both the stages, bottom-up agglomerative approaches are used to cluster similar sounds. The clustering procedure is illustrated in Figure 4. The example waveform shown in the figure contains 7 segments. HMMs (M1 to M7) are trained for each of the segments (S1 to S7). Here, each HMM is trained using multiple training instances of a segment obtained by multiple frame rate and multiple frame size. The segments are merged to form clusters C1 to C4. GMMs (M1 to M4) are trained using the segments from the respective clusters. This is performed by maximum a posteriori (MAP) adaptation of the individual GMMs from a universal GMM trained using complete data. The GMMs are again merged iteratively using KL divergence score obtained between all pairs of GMMs (clusters) to from a distinct set of clusters. The algorithm is as follows:

1. Each segment obtained from GD segmentation is assumed to be one cluster.
2. Train HMMs $H_1, H_2, \cdots, H_n$ for each of the clusters using multiple frame rate and multiple frame size.
3. Calculate the log-likelihood for each segment with respect to all the trained models.
4. Based on the log-likelihood scores, get the 2-best models for each of the segments [13] [14].
5. Merge the clusters C_a and C_b only if the 2-best models for any of the two segments are $\{H_a, H_b\}$ and $\{H_b, H_a\}$, respectively. Before merging, the model pairs are sorted in descending order based on the sum of log-likelihood scores with respect to the segment of the 2-best models.
6. If the merged cluster has more than 5 segments, it does not participate in the merging process anymore.
7. Repeat steps 2 to 6 until no new cluster is obtained.
8. Train a universal GMM using complete data.
9. Train GMMs $G_1, G_2, \cdots, G_m$ for each of the clusters by MAP adaptation from the universal GMM
10. KL divergence is measured between all pairs of GMMs.
11. Let $\{r,s\}$ be a pair with the least KL divergence. Let X and Y be the set of points in the clusters r and s respectively.
12. Let G_t be the GMM to be merged using G_r and G_s. If $|P(X|G_r) - P(Y|G_s)| > |P(X|G_t) - P(Y|G_t)|$, the clusters are merged else block the cluster pair from merging in the subsequent iterations. Where, $P(X|G)$ is the average likelihood of the set of points X belonging to the GMM G.
13. Repeat steps 10 to 12 until no new cluster is merged.
14. The final clusters correspond to different sounds present in the audio file.

4. Experiments and results

For training models, cepstral coefficients of 39 dimension (13 MFCC + 13 velocity + 13 acceleration) are used as features. They are obtained by applying linear filterbank on the log magnitude spectrum, followed by Discrete cosine transform (DCT). While computing cepstral coefficients, an analysis window of $1ms$ with a shift of $0.5ms$ is used. As the frequency of marmoset calls are above 5 kHz, a window size of $1ms$ will ensure that there are at least 10 cycles per frame. The window size also ensures that there are enough number of feature vectors available for each segment to estimate reliable models. Features are extracted with 20 different configurations of frame size and frame rate so that enough number of examples are there to train HMMs for each of the segments. Frame sizes of $2ms$ to $10ms$ with a step value of $0.4ms$ with corresponding frame shifts of $0.5ms$ to $2.5ms$ with a step value of 0.1 are used. VAD algorithm gives a set of possible regions for which segmentation is to be performed. Each frame of size $1ms$ is classified as vocalized/non-vocalized frame. The thresholds for VAD, as explained in Section 3.1, are empirically chosen as 0.4 for STE and 0.01 for SZCR. A VAD region should be at least $5ms$ long to consider as vocalized/non-vocalized segment; else it is associated with the previous segment. That is when a pair of voiced regions are disconnected by an unvoiced region with a duration

Table 1: *Cluster purity for different sounds on different files*

call	Cricket			Enid			Johnny			Baby Beans		
	#seg	#clusters	purity	#seg	#clusters	purity	#seg	#clusters	purity	#seg	#clusters	purity
ph	52	6	0.75	80	5	1.00	16	0	-	11	0	-
ot	70	8	0.67	0	0	-	11	0	-	1	0	-
ek	29	3	0.85	16	2	0.75	0	0	-	4	0	-
tr	54	5	0.80	93	4	0.67	117	9	0.82	102	8	0.76
tw	36	4	0.52	23	3	0.91	150	8	0.83	63	7	0.78
chi	0	-	-	0	-	-	54	6	0.81	7	0	-
$trph$	0	-	-	0	-	-	16	1	1.00	0	0	-
$others$	14	0	-	0	0	-	0	-	-	3	0	-
Tot/Avg.	255	26	0.72	212	14	0.83	364	24	0.87	191	15	0.77

of $5ms$; they correspond to two different voiced regions. Different calls as shown in Figure 3 of the marmoset are either a concatenation of syllables (tw, ek) or a single call (tr, ot and ph). For the calls of the first kind, the segmentation algorithm segments at the syllable level. For the second kind, the full call is obtained as one segment. During the HMM clustering process, clusters size is restricted to 5 segments to ensure that the clusters are more or less pure. If an impure cluster is allowed to grow, it tends to attract pure clusters. GMM merging is used to merge the big clusters obtained from HMM clustering. By using KL divergence measure, GMMs can be directly compared. While computing KLD measure, to maintain the correspondence between the mixture of the GMMs, the individual GMMs are trained by adapting a universal GMM trained using all the segments. Ideally, if the clusters are pure, at every iteration, the pair with least KLD should merge. The difference in the average likelihoods before and after merging is used as a criterion to ensure the merging of pure clusters, as explained in section 3.2.

The overall clustering results and the quality of clusters for the 4 files of different marmosets are shown in Table 1. Each cluster is assigned to a call that has a maximum number of segments in the cluster. The purity is measured as a percentage of the true calls across the clusters. One hundred percent (1.0) purity implies that the clusters of a particular call contain segments from no other call. It can be seen from the results that the average purity of the clusters is around 0.75 with the best average cluster purity of 0.87 for Johnny. Purity is not defined for calls that are not present in the file and also for the calls that have no cluster. Not all the sounds are equally available in one recording instance. For example, 85% of the calls by Baby Beans are from tr and tw.

It can be seen from the table that the total number of final clusters is more than the number of distinct sounds present in the audio. Each call has more than one cluster. For instance, the call *ot* for the file cricket has 8 clusters. Further investigation of the clusters reveals that each of the segments in a cluster shares common characteristics. This is illustrated using the spectrograms of 3 calls from the cricket file in Figure 5. Each row in the figure corresponds to spectrograms of one call. In each row, first two columns represent segments from one cluster and the next two columns represent segments from a different cluster. Thus, two clusters for 3 kinds of calls are shown. The first, second and third row correspond to the calls *ot*, *tw* and *tr* respectively. For the call *ot*, the segments of cluster 1 show one band in the spectrogram whereas, that of the cluster 2 show two bands. Similarly, for the call *tw*, the segments in cluster 2 has a hook-like structure, which is not seen in cluster 1. For the call

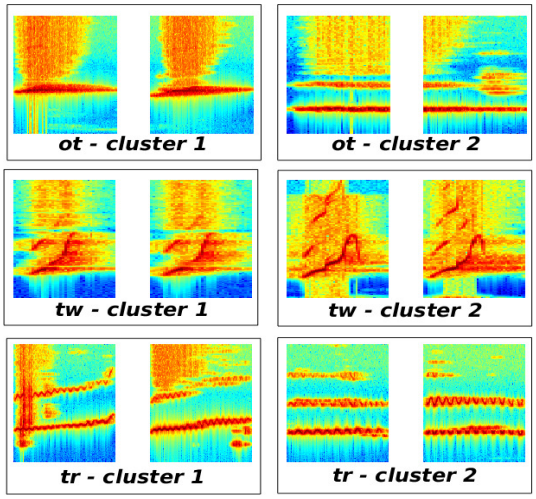

Figure 5: *Spectrograms of calls from different clusters*

tr, there are two bands in cluster 1 whereas there are three bands in cluster 2. There are many such observations on other clusters and other calls as well. These characteristics may reveal information about the conversation or the mood of the marmoset.

5. Conclusions

The objective of this work is to discover the calls in a marmoset conversation. The marmoset calls are also syllable-like, similar to human calls. The syllable-like calls are first segmented using group delay signal processing. Next, the segmented syllable-like units are individually modeled using HMMs. The obtained segments are merged using a two-tier agglomerative approach. The clustered units are found to be similar. Names are associated with the clustered units using experts' information. The specific characteristics observed among the clustered segments may reveal some interesting information about the marmoset. The discovered clusters can also be used to model the dialogue between different marmosets.

6. References

[1] A. de Castro Leãoa, A. D. D. Netob, and M. B. C. de Sousaa, "New developmental stages for common marmosets (Callithrix jacchus) using mass and age variables obtained by K-means algorithm and self-organizing maps (SOM)," *Computers in Biology and Medicine*, vol. 39, pp. 853–859, 2009.

[2] M. G. Rosa and R. Tweedale, "Visual areas in lateral and ven-

tral extrastriate cortices of the marmoset monkey," *The Journal of Comparative Neurology*, vol. 422, no. 4, pp. 621–651, 2000.

[3] R. Rajan, V. Dubaj, D. H. Reser, and M. G. P. Rosa, "Auditory cortex of the marmoset monkey complex responses to tones and vocalizations under opiate anaesthesia in core and belt areas," *European Journal of Neuroscience*, vol. 37, no. 6, pp. 924–941, 2013. [Online]. Available: http://dx.doi.org/10.1111/ejn.12092

[4] J. A. Agamaite, C.-J. Chang, M. S. Osmanski, and X. Wang, "A quantitative acoustic analysis of the vocal repertoire of the common marmoset (callithrix jacchus)," *The Journal of the Acoustical Society of America*, vol. 138, no. 5, pp. 2906–2928, 2015. [Online]. Available: http://dx.doi.org/10.1121/1.4934268

[5] A. L. Pistorio, B. Vintch, and X. Wang, "Acoustic analysis of vocal development in a New World primate, the common marmoset (Callithrix jacchus)," *The Journal of the Acoustical Society of America*, vol. 120, pp. 1655–1670, 2006.

[6] C. P. Chow, J. F. Mitchell, and C. T. Miller, "Vocal turn-taking in a non-human primate is learned during ontogeny," *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 282, no. 1807, 2015. [Online]. Available: http://rspb.royalsocietypublishing.org/content/282/1807/20150069

[7] A. Wisler, L. J. Brattain, R. Landman, and T. F. Quatieri, "A framework for automated marmoset vocalization detection and classification," in *Interspeech*, 2016, pp. 2592–2596. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2016-1410

[8] C.-J. Chang, "Automated classification of marmoset vocalizations and their representations in the auditory cortex," 2014. [Online]. Available: http://jhir.library.jhu.edu/handle/1774.2/37097

[9] H. K. Turesson, S. Ribeiro, D. R. Pereira, J. P. Papa, and V. H. C. de Albuquerque, "Machine learning algorithms for automatic classification of marmoset vocalizations," *PLOS ONE*, vol. 11, no. 9, pp. 1–14, 09 2016. [Online]. Available: http://dx.doi.org/10.1371%2Fjournal.pone.0163041

[10] V. K. Prasad, T. Nagarajan, and H. A. Murthy, "Automatic segmentation of continuous speech using minimum phase group delay functions," *Speech Communication*, vol. 42, no. 34, pp. 429 – 446, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167639303001444

[11] T. Nagarajan, "Implicit systems for spoken language identification," Ph.D. dissertation, Indian Institute of Technology Madras, 2004.

[12] J. Sebastian, P. A. Manoj Kumar, and H. A. Murthy, "An analysis of the high resolution property of group delay function with applications to audio signal processing," *Speech Communication*, vol. 81, pp. 42 – 53, 2016, phase-Aware Signal Processing in Speech Communication.

[13] G. Lakshmi Sarada, A. Lakshmi, H. A. Murthy, and T. Nagarajan, "Automatic transcription of continuous speech into syllable-like units for indian languages," *Sadhana*, vol. 34, no. 2, pp. 221–233, 2009. [Online]. Available: http://dx.doi.org/10.1007/s12046-009-0006-0

[14] G. L. Sarada, N. Hemalatha, T. Nagarajan, and H. A. Murthy, "Automatic transcription of continuous speech using unsupervised and incremental training," in *Interspeech*, 2004, pp. 405–408.