|  |
| --- |
|  |
| **Unsupervised approaches for Khmer word acoustic and speaker discrimination** |
|  |

**Chanmann LimFernando Torre-Mora**

Department of Computer Science Department of Computer Science

University of Missouri University of Missouri

Columbia, Missouri, 65211 Columbia, Missouri, 65211

*cl9p8@mail.missouri.edu* *fthc8@mail.missouri.edu*

**Abstract**

Although many researches have been conducted in the field of human-language technology, hoping to bridge the gap between human-to-machine and human-to-human communication, the focus has mainly been on English. Among the neglected languages is the Khmer language, the official language of the Kingdom of Cambodia. We seek to investigate the natural grouping of utterances concealed in speech acoustic properties by digging into how purely unsupervised approaches – namely hierarchical clustering, k-medoids clustering, fuzzy c-medoids clustering and spectral clustering – are able to reveal the natural grouping in the context of word and gender clustering from raw audio (Khmer speech). We present an evaluation of the methods using normalized mutual information (NMI), accuracy by counting. We conclude that DTW and Spectral Clustering are the best methods for this task.

# Introduction

The advances in human language technology have given rises to myriad competitive advantages to the speakers of the mainstream language such as English [1]. However, more than 80% of all technological progress is done in less than 5% of the world’s languages [2, 3] despite willingness of those languages’ millions of speakers to utilize new technology in their language [4, 5]. Albeit many researches have been conducted in the field of human language technology to hopefully bridge the gap between human-to-machine and human-to-human communication, the focus has mainly been on English and most of world languages remain to be marked as low-resourced or under-resourced and continue to lack behind new innovation adaptation and advantages. We take as our case study the Khmer language, the official language of the Kingdom of Cambodia, spoken by more than 15 million speakers worldwide. In this task, we seek to investigate the natural grouping of utterances concealed in speech acoustic properties by digging into how purely unsupervised approaches namely hierarchical clustering, k-medoids clustering, fuzzy c-medoids clustering and spectral clustering be able to reveal the natural grouping in the context of word and gender clustering from raw audio speech input in Khmer language. Finally, we will present the evaluation of each clustering method in term of normalized mutual information (NMI), accuracy by counting.

We wish to have this work as one of the kick-off project to establish an understanding of various unsupervised approaches in dealing with real-world speech data to better prepare for our goal towards building automatic speech recognition (ASR) for Khmer language. We believe this task not only strengthen what we have learned in the “Unsupervised Learning” class but also remove one of the barriers between our goal.

# Related Work

Masood Masoodian and Bill Rogers and Saturnino Luz. “Improving automatic speech transcription for multimedia content”

# Dataset

The “Khmer keywords” database [6] created by Institute of Technology of Cambodia to be used in Interactive Voice Response (IVR) system originally consists of 193 commonly used words including provinces’ names, numerical counting, months’ names, weekdays, yes/no answer, common diseases’ names and essential commodities. Recorded by 14 university students (8 males and 6 females) aged between 19 and 23 in a low to semi-noisy environment with mobile phones to mimic the voice responses during IVR transaction. In our experiments we selected a subset from the aforementioned database to include only the first 103 unique words from 5 male and 5 female speakers such that data imbalance in either spoken words or gender are filtered out.

# Method / Approach

As preprocessing, we performed spectral analysis to extract 13-dimension Mel-Frequency Cepstral Coefficients (MFCCs) (MFCC) and their corresponding first and second derivatives for each sound sample. Dynamic Time Warping (DTW) distance between each pair of samples is computed to be used as proximity measure for our clustering algorithms. We then attempt to visualize word clustering of each talker via hierarchical clustering and dendrogram to see if meaningful structures can be formed. To continue we relax our problem by excluding samples belong to each gender to take away the variation contributed by gender difference for word clustering and vice versa for gender clustering before applying our clustering methods. We then move on to run our experiments on the entire dataset and evaluate the validity of the clustering results by comparing to either the ground truth or the output of corresponding transcription clustering using normalized mutual information and accuracy by counting.

## Feature Extraction

Instead of working directly with raw audio signals, we make use of speech signal processing technique commonly used in automatic speech recognition system to extract feature vectors, which might contain better information-packing properties for speech analysis.

The Basic mechanisms in speech signal processing is to divide the time signals into a series of successive frames, each consists of a finite of N samples with overlapping neighboring frames (Figure 1(a)), that are assumed to be quasistationary such that spectral analysis can be carried out (Figure 1(b)).

To simplify the process we employ Hidden Markov Models Toolkit (HTK) for extracting 13-dimension MFCCs, first and second order derivatives with the window size is set to 25ms, the step size is 10ms resulting in a vector of 39 elements representing each frame. Then the first 13 elements of each vector corresponding to the static part of MFCCs are standardized with *zscore* function and the rest are normalized to have unit-variance such that the contribution of each element is noticeable.

## Dynamic Time Warping

Compare two sequences with different length (see Figure 3) then normalize the distance by it warping path

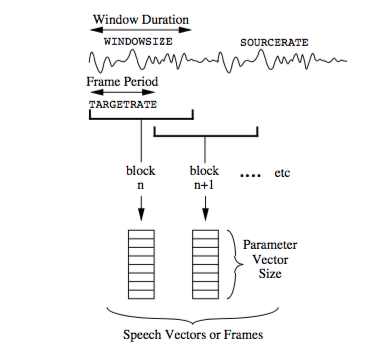
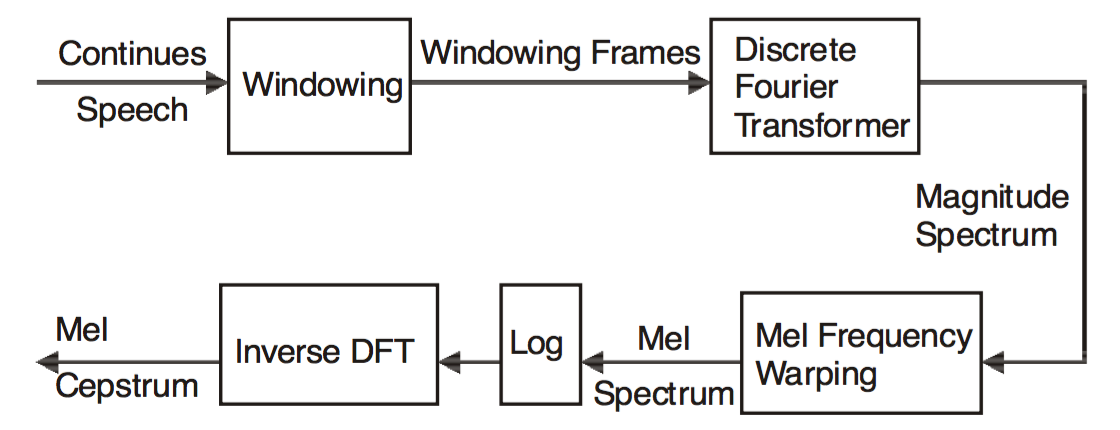
 

Figure : Process for (a) Speech Encoding (HTK) and (b) the entire MFCC computation [7]

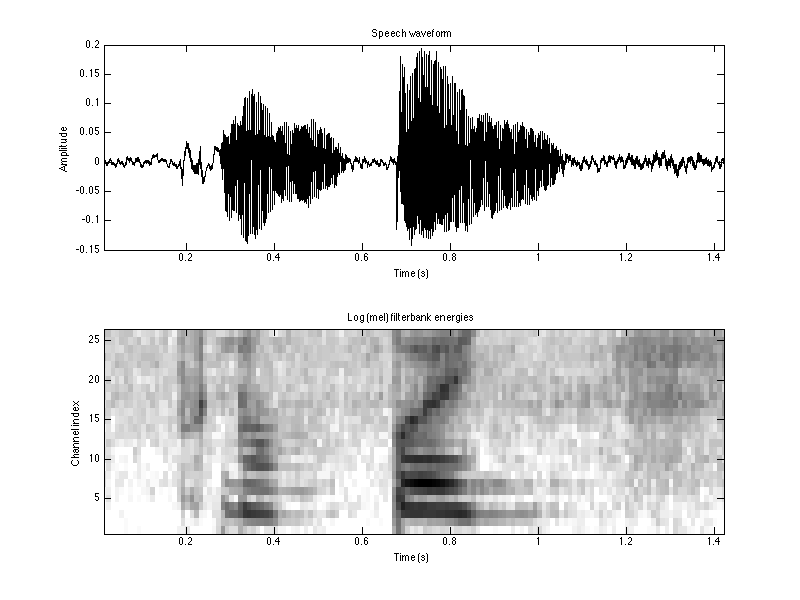
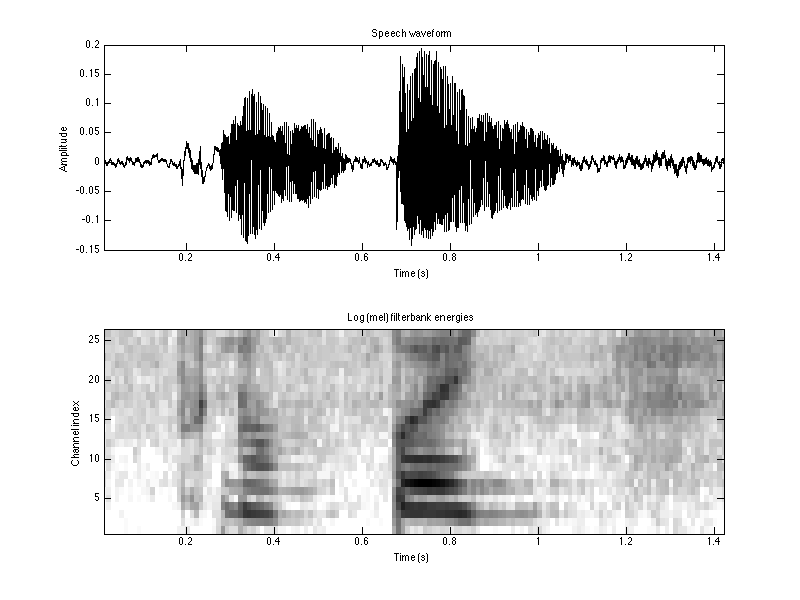
 

Figure : Waveform (a) and Spectrogram (b) the word ភ្នំពេញ (IPA: pnum.peːɲ)

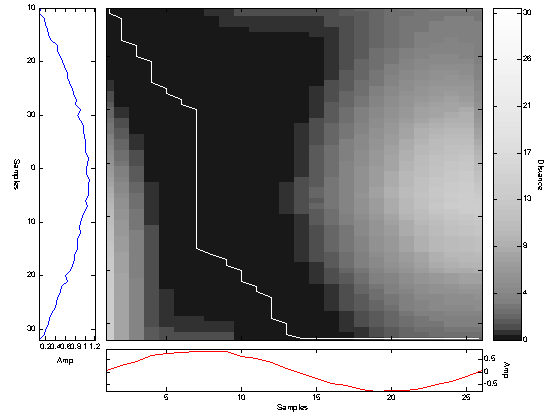
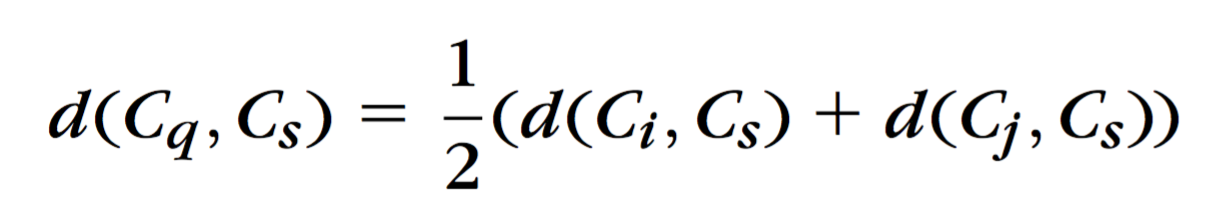


Figure :

## Clustering methods

### Hierarchical Clustering

WPGMA (weighted pair group method with averaging)

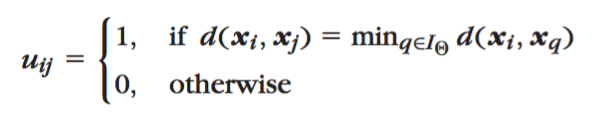
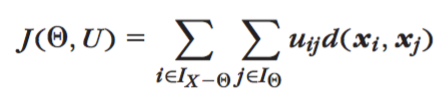


### k-Medoids Clustering

For this task which each sample is a sequence of different length, k-Medoids is to be considered since the mean vector of the dataset could not be deterministically defined. The quality of the solution of k-Medoids given a set of medoids is to minimize the cost function

Randomly initialize k medoids

Iteratively swap the medoid to minimize the cost function



### Fuzzy c-Medoids

In Fuzzy c-Medoids algorithm we followed the procedure from [ref]

1. Fix the number of cluster c
2. Pick initial medoids randomly from entire dataset *X*
3. **Repeat**
   1. Compute membership *uij* for *i = 1, 2, …, n* and *j = 1, 2, …, c*
   2. Store current modoids: ;
   3. Compute new medoids *vj* for *j = 1, 2, …, c*: ;  
      *vi = xk;*
4. **Until** ()

where the membership function is defined as and is chosen to reflect the size the clusters and the fuzzifier *q* is set to 1*.8* since higher *q* will make all memberships become very small except the centroid itself.

### Spectral clustering

Spectral clustering, the result from the field of spectral graph theory, is to minimize the normalized cut criterion, which can be formulated as generalized eigenvalue problem [8, 9]. Here we adapted the algorithms from [8] and [9] to first compute normalized graph Laplacian matrix of fully connected graph where and *D* is defined to be diagonal matrix whose *(i,i)* element is the sum of *W*’s *i*-th row, then zscore normalization is performed on *k* eigenvectors corresponding to *k* largest eigenvalues to form Y matrix and finally applying k-mean clustering by treating each row of Y as a point in . The parameter in computing *W* is chosen to maximize the NMI of the clustering result.

## Evaluation metrics

We performed three evaluations on our clustering results to measure the usefulness of the clustering: an accuracy measure, mutual information, and (in the case of hierarchical clustering) cophenetic correlation. For the first two measures, which require a ground truth to compare against, we generated one for each level of clustering.

As an initial exploration, we trained each classifier with just one speaker. Unfortunately, due to space considerations, we cannot report those results here. Our focus will be the evaluation of the clusterings generated when using all speakers.

We performed and evaluated two clusterings for each classifier: one using all 39 MFCCs, and the other using just the first 13 MFCCs.

### Ground Truth generation

The Khmer keywords dataset contains, for each audio file, a transcription of the word being recorded in ASCII characters. The transcriptions are standardized using Arpabet; thus, whenever the syllable Macintosh HD:Users:fthc8:Desktop:Screen Shot 2015-12-13 at 3.32.53 PM.png (IPA: kaːɛt) occurs, the ASCII transcription is always “KH AE T”. This allows us to verify if the clusters match the words.

To be able to verify the word clustering at all levels of clustering, we performed an additional clustering using the set of all transcriptions. Clustering algorithms received the same number of text data points as of speech data points. In this way, we were able to compare the clustering of the audio files against the text regardless of how many clusters were desired. For comparability, we performed each clustering on the text using the same clustering method and parameters, changing only the distance measure to Levenshtein, since it is a more natural distance measure for strings. Specifically, we used the Vagner-Fisher algorithm to compute the distance. [10]

We reason that if the clustering methods can behave close enough for both the text representation of the recording and the audio representation, they are correctly identifying the words.

For Gender clustering, we labelled each recording with the speaker’s gender using the metadata. We expected to see all the individual recordings by female speakers in the same cluster, and all those by male speakers in the other cluster.

### Naïve Accuracy measure

Cluster labels are arbitrary. To perform the comparison, we first perform a mapping from the clusters found by the speech clustering and the clusters found by the text clustering . The mapping for each *i*th cluster of is determined by seeing which cluster *j* of is most common among the points . We then assume that cluster maps to , and consider the points as accurately clustered, while considering the points as inaccurately clustered. We call this naïve accuracy, because it doesn’t enforce a 1:1 mapping. This computation is summarized in Figure 4. More formally,

This means that, for a number of clusters *m*, the accuracy is computed by performing a clustering method on the MFCC representation of each recording’s audio , and then performing the same clustering method on the ASCII representation of the same recordings’ transcriptions . The results, and are then compared by trying to map each cluster in to a cluster in and counting the number of points that follow this mapping.

**Function** naïve\_accuracy(, : linear vectors of cluster labels of length )

**returns**

**Let** :a set; :a vector of sets; :a vector of multisets; :a map; :an integer

remove\_duplicates()

**For** **each**

{: }

**For each**

**End For**

mode()

**End For**

**For** **each**

**If**

**End If**

**End For**

**Return**

**End Function**

Figure 4: Pseudocode for the Naïve accuracy measure

### Normalized Mutual Information score

As a less naïve method of comparison between our text clustering and our speech clustering, we used normalized mutual information (NMI): a statistical method comparing the distribution of the two labelings ( and ) [11, p. 884]. Specifically, we use the normalization proposed by Strehl and Ghosh [12]: , where is the unnormalized mutual information and is the entropy. Defining as ; as ; and as ; we have: ; ; and

The computation of ,, and can be seen in Figure 5 and Figure 6

**Function** P( : a linear vector of cluster labels of length )

**returns**

**Let** :a set; :a vector of sets; :a vector

remove\_duplicates()

**For** **each**

{: }

**End For**

**For each**

**End For**

**Return**

**End Function**

Figure 5: Pseudocode for the computation of and

**Function** Pij(, : a linear vector of cluster labels of length )

**returns**

**Let** ,:sets; ,:vectors of sets; :a matrix

remove\_duplicates()

remove\_duplicates()

**For** **each**

{: }

**End For**

**For** **each**

{: }

**End For**

**For** **each**

**For** **each**

**End For**

**End For**

**Return**

**End Function**

Figure 6: Pseudocode for the computation of

### Cophenetic Correlation coefficient

The cophenetic correlation coefficient (CPCC) is a metric that measures to what degree the dissimilarities between a dataset’s points are preserved by a clustering. It is specifically designed for agglomerative clustering, as it uses the point at which the clusters merge to compute the cophenetic distance [11, p. 873]

where *C* is the clustering, *Z* is the cophenetic distance, *μ* is the arithmetic mean, and [13]

The CPCC requires the assumption of monotonicity for every clustering with *m* clusters with respect to a clustering with *m*+1 clusters [11, p. 679], however, proving this for K-medioids, C-medioids, and Spectral Clustering + K-means, which by their very nature are non-deterministic, is impossible. Further, CPCC computations are built on the assumption that the point at which two clusters merge can be easily pointed out [11, p. 680, 13] and finding a suitable way to compute the cophenetic distance at all levels for these methods, escapes the scope of this paper. For these reasons we cannot assert it is a valid metric in these cases. We therefore only evaluate Hierarchical Clustering with CPCC.

# Results

We will first discuss our study of the hierarchical clustering results, and then perform a comparison of all the methods evaluated. We dedicate particular attention to the hierarchical clustering results, since it, in fact, gives as many results as the clusters it is asked to create.

## Hierarchical Clustering

First we present the dendograms (Figure 7) which will prove instrumental in understanding all other results. We’ll ignore superficial differences (such as cluster ordering or sizes) and instead point out that the number of clusters at each successive level is quite close between the text dendrogram (Figure 7(a)) and the 39 MFCC dendrogram (Figure 7(b)), suggesting that the underlying structure found by the hierarchical classifier in the speech is reflective of the underlying structure of the transcriptions. However, of note is the differences between the y-axes (1–18 for Figure 7(a) and 13–16 for Figure 7(b)). This suggests that it may be difficult to obtain a perfect score, since the cophenetic distances of two identical clusters will always be mismatched. The number of clusters in the 13 MFCC dendrogram (Figure 7(c)) maintains closeness with the other two except in the bottommost levels; however, its y-axis range of 2-9 make it a much better candidate for scoring.

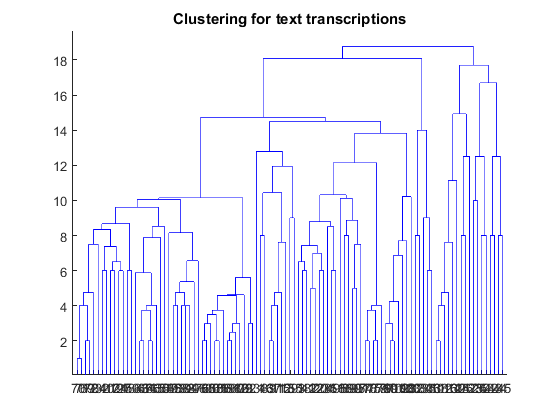
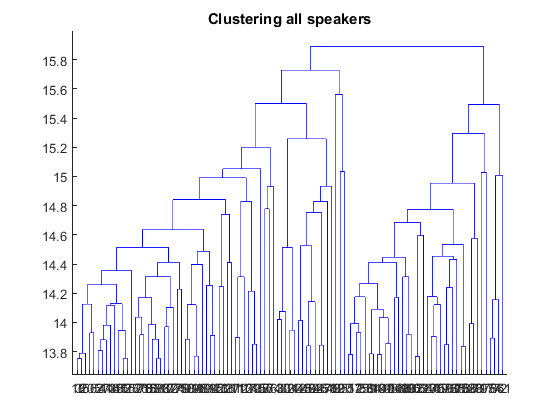
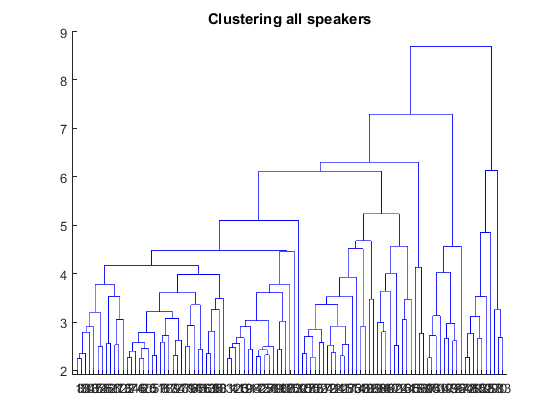
  

Figure : Dendrograms showing the clusters each of the 103 words get put in for (a) a classifier trained on the text representation of the file, (b) a classifier trained on the 39 MFCC representation of all 10 speaker’s files, and (c) a classifier trained on the 13 MFCC representation of all 10 speaker’s files

We then studied the accuracy for every possible value of where *m* is the number of clusters. In effect, we moved up the dendrograms and compared the similarity of the clusters. (See Figure 8.) We can easily observe that the naïve accuracy starts at 1 for one cluster, but since all data points are in the same cluster, this provides no information at all (NMI=0). NMI steadily increases as *m* grows, while naïve accuracy steadily decreases. It is of note that naïve accuracy crosses 0.4, especially considering that, if each datapoint were to be assigned a word cluster with uniform probability, less than one percent would be clustered as indicated by the text clustering .

We have highlighted the intersection in Figure 8(a) because our initial explorations showed that this is the point with the optimal performance; after that, the classifier does not generalize as well, as can be seen by NMI tapering off and the training set accuracy beginning to rise. Note that in Figure 8(b), the intersection would occur after 103 (the number of words).

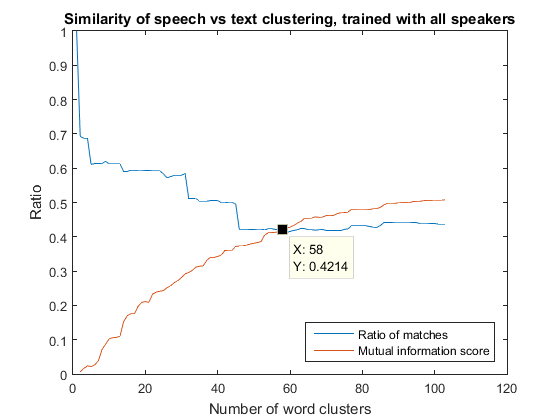
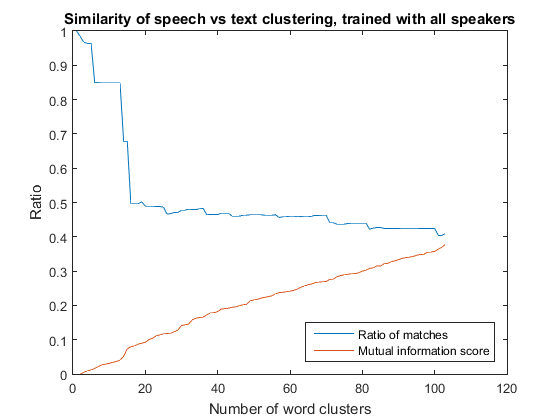
 

Figure : Comparison of NMI score (red) with naïve accuracy (blue) for (a) classifier trained with all 10 speakers using 39 MFCCs and (b) classifier trained with all 10 speakers using 13 MFCCs

Finally, we would like to point out the sudden drop in accuracy at 13 clusters in Figure 8(b), which gives us a more exact indication of where the dendrograms no longer resemble one another.

## Comparison

### Word identification

When using all 39 MFCCs, spectral clustering – as the only clustering method to obtain an NMI>0.6 – is by far the best clustering method we evaluated. Fuzzy C-medioids and K-medioids and are close seconds with NMI≈0.55 and Hierarchical clustering comes in last with an NMI of 0.51. Naïve accuracy results were in line with the NMI results, ranking the methods in the same order; however, naïve accuracy results were only of 0.21 at the lowest and 0.29 at the highest. The CPCC score was similarly poor, obtaining only 0.39.

When using 13 MFCCs, Fuzzy C-medioids takes the lead with an NMI of 0.49 and a naïve accuracy of 0.15. K-medioids gets the same naïve accuracy, but falls short on its NMI with 0.47. Spectral clustering comes third with an NMI of 0.42 and an accuracy of 0.13. Hierarchical clustering sees the greatest disimprovement in its NMI with 0.38; however, its naïve accuracy is in line with the other methods at 0.14

The results are shown in Table 1. Note that all metrics except CPCC improve when using all 39 MFCCs. This makes sense due to the dissimilarity indices shown in the dendrogram (see discussion for Figure 7). However, it seems to suggest as well that 13 MFCCs gives a better hierarchy, which Figure 8 reveals is not the case.

Table : Comparison of the experimented methods by metric for Word identification

| **Row Labels** | **Naïve** | **NMI** | **CPCC** |
| --- | --- | --- | --- |
| **39 MFCCs** |  |  |  |
| Hierarchical | 0.2126 | 0.5109 | 0.3926 |
| K-Mediods | 0.2464 | 0.5530 |  |
| Fuzzy C-Med | 0.2581 | 0.5501 |  |
| Spectral | 0.2850 | 0.6100 |  |
| **13 MFCCs** |  |  |  |
| Hierarchical | 0.1379 | 0.3782 | 0.6659 |
| K-Mediods | 0.1544 | 0.4734 |  |
| Fuzzy C-Med | 0.1544 | 0.4896 |  |
| Spectral | 0.1318 | 0.4277 |  |

### Gender identification

The clustering methods fared much better for the gender identification experiment, with the highest naïve accuracy score being 0.8 (hierarchical clustering, 13 MFCCs). This suggests that the clusterings are, in fact, influenced by the gender of the speaker for about 8 of the 10 speakers. However, the same method with 39 MFCCs obtained a much lower naïve accuracy: 0.51. This suggests that with more features the clustering becomes less gender-dependent. The fact that the number is also not a multiple of one-tenth suggests that certain words may have a higher probability of being influenced by the speaker’s gender when being clustered. K-medioids and Fuzzy C-means both obtained around 0.5 naïve accuracy in both cases, suggesting a lesser variability of these methods given the number of features.

The hierarchical clustering was also the only one that scored appreciably when measured with NMI. It obtained 0.39 with 13 MFCCs while all other methods obtained less than 0.01.

Finally, the CPCC numbers were again in stark opposition with the accuracy numbers: for 13 MFCCs the CPCC was only 0.36 while for 39 MFCCs, the CPCC was 0.67.

The details of these results can be observed in Table 2.

Table : Comparison of the experimented methods by metric for Gender identification

| **Row Labels** | **Naïve** | **NMI** | **CPCC** |
| --- | --- | --- | --- |
| **39 MFCCs** |  |  |  |
| Hierarchical | 0.5068 | 0.0061 | 0.6659 |
| K-Mediods | 0.5233 | 0.0019 |  |
| Fuzzy C-Med | 0.5000 | negligible |  |
| Spectral |  |  |  |
| **13 MFCCs** |  |  |  |
| Hierarchical | 0.8000 | 0.3970 | 0.3553 |
| K-Mediods | 0.5427 | 0.0110 |  |
| Fuzzy C-Med | 0.5000 | negligible |  |
| Spectral |  |  |  |

# Conclusions

Normalized DTW offers a means for clustering experimentation for speech data

Gender, speaker characteristics and emotion, words similarity and recording condition play major role in speech variabilities

Using 39 MFCCs instead of 13 MFCCs gives an average improvement of 50.5 percent in word clustering, but gives an average disimprovement of 55.3 percent in gender clustering (Table 1).

It’s hard to generalize acoustic discovery using purely unsupervised methods. We were able to cluster by word in less than one-third of all data points, indicating that a system based on these methods would be able to make out less than one third of the words it records. This is not suitable for commercial purposes. However, given the number of clusters we were trying to identify, we point out that our results are 29.4 times better than selecting words purely at random. This suggest that, even with this limited performance, the clustering methods were able to find at least some underlying structure.

## Future work

The high degree of similarity among the words as opposed to how they’re spoken (see Figure 7) poses a great challenge for word clustering.

The hierarchical clustering apparent optimal (see discussion on Figure 8(a)) suggests that looking for less clusters might offer substantial steps towards making the system commercially viable, but creates the challenge of finding a meaning for these clusterizations. If the results could be labelled using the word’s linguistic properties (for example, trying to find a cluster grouping words by their root, affix, or part of speech) and later disambiguated (in effect, multi-step classification), a more precise word identification may result. An intermediary study using the clusterings for all values of *m* for all the clustering methods (see sections 4.4.4 and the introduction to section 5.1) might aid in defining the nature and size of these groupings.

However, it is of note that Khmer words have an especial tendency to share affixes. For this reason we suggest, rather than using a dataset split by word, using a dataset where each recording contains a smaller acoustic unit (e.g.: a phoneme or a syllable). Clustering on such a dataset could yield more practical results, and (if successful) would transfer much of the weight of the problem to the already well-developed field of splitting speech into acoustic units [14, 15]. Context could then be analyzed using, for example, a context-dependent Hidden-Markov Model.

# References

|  |  |
| --- | --- |
| [1] | M. Masoodian, B. Rogers and S. Luz, "Improving automatic speech transcription for multimedia content," in *Proceedings of WWW/Internet*, 2007. |
| [2] | C. Dewey, "How the Internet is killing the world’s languages," Washington Post, 2013. |
| [3] | M. Orcutt, "The Online Language Barrier," MIT Technology Review. |
| [4] | N. Kelly, "Speak to Global Customers in Their Own Language," Harvard Business Review, 2012. |
| [5] | E. Zuckerman, "English is no longer the language of the web," Quartz, 2013. |
| [6] | C. Lim, "Khmer Keywords Transcription," Institute of Technology of Cambodia, 2015. [Online]. Available: http://khmerasr.github.io/blog/2015/05/29/khmer-keywords-transliteration/. |
| [7] | V. Tiwari, "MFCC and its applications in speaker recognition," *International Journal on Emerging Technologies,* vol. 1, no. 1, pp. 19-22, 2010. |
| [8] | J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, p. 8, 2000. |
| [9] | A. Y. Ng, M. I. Jordan and Y. Weiss, "On Spectral Clustering: Analysis and an algorithm," *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS,* pp. 849--856, 2001. |
| [10] | E. Polityko, "Calculation of distance between strings," The MathWorks, Inc., 2007. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/17585-calculation-of-distance-between-strings/content/strdist.m. |
| [11] | S. Theodoris and K. Koutroumbas, Pattern Recognition, Fourth ed., Burlington, MA: Academic Press, 2009. |
| [12] | A. Strehl and J. Ghosh, "Cluster Ensembles – A Knowledge Reuse Framework for," *Journal of Machine Learning Research,* vol. 3, pp. 583-617, 2002. |
| [13] | "cophenet," The Mathworks, Inc., 2005. [Online]. Available: http://www.mathworks.com/help/stats/cophenet.html. |
| [14] | A. Sethy and S. Narayanan, "Split-Lexicon Based Hierarchical Recognition Of Speech Using Syllable And Word Level Acoustic Units," in *International Conference on Acoustics, Speech and Signal Processing*, Hong Kong, 2003. |
| [15] | R. Bachu, S. Kopparthi, B. Adapa and B. Barkana, "Voiced/Unvoiced Decision for Speech Signals Based on Zero-Crossing Rate and Energy," *Advanced Techniques in Computing Sciences and Software Engineering,* pp. 279-282, 2009. |