Unsupervised Phoneme Segmentation in Continuous Speech

Stephanie Antetomaso
Wheaton College
Norton, MA USA
antetomaso_stephanie@wheatoncollege.edu

Abstract

A phonemic representation of speech is necessary for many real world applications, but the algorithms for deriving these representations are generally either language specific, or require heavy amounts of manual preprocessing. We use a developmental approach to the problem to arrive at an unsupervised algorithm for discretizing continuous speech into a sequence of phonemes which is inspired by algorithms used for text segmentation. In this paper we outline the algorithm and demonstrate its use on multi-speaker continuous speech.

1 Problem and Motivation

Many real-world problems in Computer Science – such as speaker modeling, speech recognition, and text-to-speech – require a representation of human speech at the phonemic level. However, the abundance of natural speech data means that manually annotating all such existent data would be unfeasible. Rather than training an algorithm on a specific language, we hope to develop a process that is language independent, allowing users to work with underrepresented languages and novel speaker data without requiring large amounts of manual preprocessing. While focusing on a developmental approach to the problem, we propose a solution based in unsupervised learning. Our work centers around the task of taking an algorithm initially developed for use in the domain of text-segmentation and modifying it to discover phoneme boundaries in multi-speaker continuous speech.

2 Background and Related Work

In the past, unsupervised phoneme discovery in speech has centered around algorithms focused purely on acoustic features (frequency, pitch, and zero-crossings) as well as signal processing techniques [3, 10]. However, these methods generally require the maximum number of phonemes in a sentence to be passed to the algorithm, limiting the possibilities for unsupervised learning. Our approach draws inspiration from word and text segmentation literature based on a text processing inspired algorithm, Voting Experts (VE), developed by Cohen, Heeringa, and Adams [6]. VE utilizes both frequency and entropy experts which vote on segment boundaries: the frequency expert votes to place segmentation points in order to maximize segment counts, while the entropy expert votes for locations where the next character is difficult to predict, taking advantage of the internal cohesion inherent in language segments. In short, this entropy expert would vote to place a segmentation point anywhere the text had a relatively high entropy.

Figure 1 shows how a passage of text is read as input and formed into a trie based on a fixed window. Frequency and entropy are calculated for each node. After the trie is formed, the text is iterated over and votes are made based on these node values. If a potential boundary has a

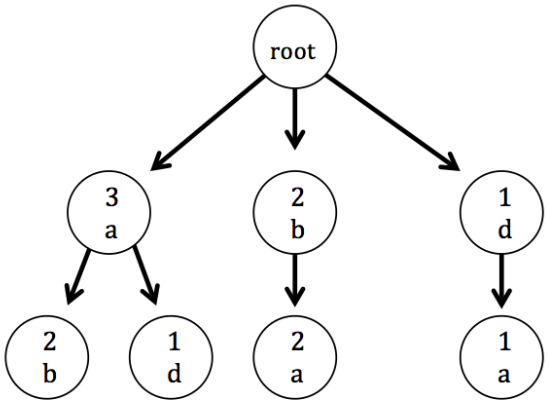

Figure 1: A Trie with Frequency Counts Formed from the Sequence *abadaba*; Window Size 2

locally maximum number of votes, a segmentation point is placed at that boundary. As output, VE creates a lexicon out of the chunked corpus. Cohen and his colleagues suggest that the use of this algorithm is not limited to just text, but can be used with any sort of information "chunks" or bits of information with a similar high-low-high entropy signature [5]. Previous applications of VE to speech have found success at discovering word boundaries from speech data building on inducing hierarchies in time series data through multiple iterations of VE [9, 1, 2]. As a continuation of this research, we believe that the VE algorithm would be helpful for solving the problem of how to segment phonemes from raw natural speech in an unsupervised manner.

3 Approach and Uniqueness

Our approach takes raw speech data as input and outputs a segmentation of this data based on the discovery of phoneme boundaries. Before we are able to use VE in a speech domain, however, the continuous stream of audio must be discretized so that there are potential boundaries on which the experts may vote. We begin with WAV files of adult speech and process them through Praat [4], a speech analysis software program which allows us to gather sequences of MelFrequency Cepstral Coefficients for each audio file using a fixed window size of 15ms and a step size of 5ms [8]. These vectors are then used as input into Vector Quantization (VQ) – a discretization method which is used to build a codebook of subphonemic prototypes labeled with unique, random string labels. The size of the codebook is an input parameter, based loosely on the number of phonemes in the relevant language – generally fewer than 50 phonemes. Using the codebook created by VQ, we replace each feature vector, obtained from the original audio files, with the closest codebook label. Similar MFCC vectors, and therefore similar speech segments, are given the same label, allowing the speech to be discretized. Once a speech string is discretized it becomes input into the VE algorithm, which forms a trie based on frequency and entropy. With our experiments we ran VE with a window size of 3 and an entropy threshold of 4, which we determined based on experimentation. Finally, the votes from the frequency and entropy experts are mapped back onto the original speech data and potential segmentation points are proposed.

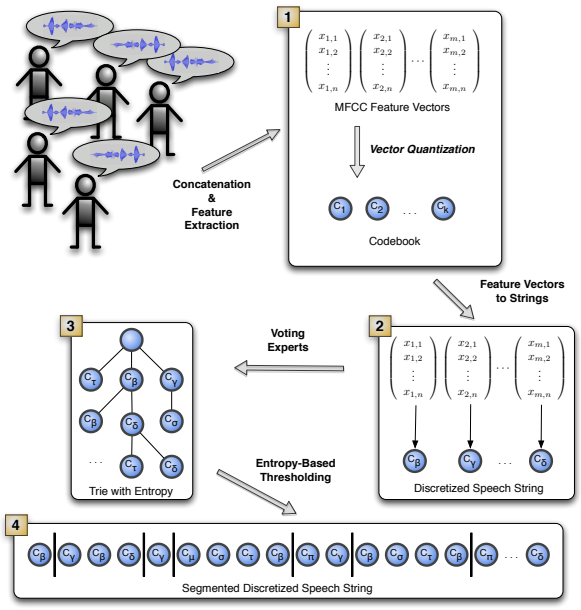

Figure 2: Algorithm Diagram

4 Results and Contributions

As input to this algorithm we used the TIMIT data set [7], a set of American English phonemically balanced sentences spoken by adults of different genders, ages, etc., and split up into eight different dialect regions specified and labeled by DARPA. Ten sentences were recorded by each speaker, with both some overlap and some sentence variation between speakers. Along with the audio files, the data set provided gold standard established phoneme boundaries for the speech corpora – these were manually annotated text files corresponding to each audio file; each listed all the phonemes in the sentence, as well as beginning and ending timestamps in frames. We used these gold standards to judge the accuracy of the phoneme segmentations proposed by the VE algorithm. Proposed boundary positions were evaluated using a windowed approach. If a selected boundary position fell within 20ms of a target position, it was marked as a true positive (a correct segmentation location) [8]. Our results were compared against a random baseline which chose n random segmentation locations with no duplicate locations, where n was the target number of boundary locations given by the gold standard included in the TIMIT data set. Precision is defined as the number of true positives divided by the total number of segmentation points *proposed by the algorithm*, while recall is defined as the number of true positives divided by the total number of segmentation points that actually *exist in the data*. F-score is a function of precision and recall, where an F-score of 1 would indicate that segmentation was performed perfectly.

	Algorithm			Baseline
k	Precision	Recall	F_1	Random F_1
122	**0.6030**	0.8250	0.6968	0.4494
155	0.6004	0.8433	**0.7014**	**0.4547**
244	0.5895	**0.8580**	0.6988	0.4502

Table 1: Results from all New England Speakers – Maximum Values in Bold

	Seg. Points		Algorithm			Baseline
Speaker	Target	Total	Precision	Recall	F_1	Random F_1
DR1/FDML0	346	5264	0.6366	0.8848	0.7404	0.4788
DR1/FECD0	409	7259	0.5989	**0.9122**	0.7230	0.4347
DR1/FETB0	372	5929	0.6104	0.8984	0.7269	0.4317
DR1/MDPK0	371	5531	0.6595	0.9074	0.7638	0.4435
DR1/MPSW0	349	4829	0.6378	0.9069	0.7489	0.4587
DR1/MTJS0	377	7294	0.5099	0.8951	0.6497	0.3794
DR1/FCJF0	349	4809	**0.6926**	0.9023	**0.7837**	0.4883
DR4/FDKN0	414	6988	0.6121	0.8901	0.7254	0.4263
DR4/FCAG0	366	5441	0.6000	0.8953	0.7185	0.4325
DR4/FSSB0	387	6685	0.5546	0.9045	0.6876	0.4248
DR4/MSTF0	397	6464	0.5637	0.8754	0.6858	0.4330
DR4/MNET0	359	5486	0.6111	0.8817	0.7219	0.4619
DR4/MLEL0	396	6965	0.5331	0.9001	0.6696	0.4009
DR4/MTAS0	347	4847	0.6794	0.9055	0.7763	**0.5069**
DR4/MTQC0	396	8345	**0.4451**	**0.8221**	**0.5775**	**0.3658**

Table 2: Experimental Results from 15 TIMIT Speakers – Max and Min Values in Bold

In our first experiment, we focused on the impact of the VQ codebook size on phoneme boundary detection. Using a single dialect region (all the New England speaker data concatenated into a single audio file), we changed the input parameter of VQ to analyze the effect on precision and recall. The 15ms window used when first splitting the audio into feature vectors is subphonemic. This ensures that each codebook entry is subphonemic as well, allowing us to minimize unwanted overlap and capture coarticulation effects between consecutive phonemes. Since each codebook entry is subphonemic, the correct input parameter to VQ should be around $2-5$ times the number of phonemes in the language. The results from Table 1 show this to be true, although the difference in F-score is not statistically significant as long as k (the input parameter) is not excessively high or low. This means that the exact number of phonemes in a language do not have to be known in order for this algorithm to still be effective.

The next experiment contained speech by 15 individuals from 2 different dialect regions where all the sentences from a single speaker were concatenated into a single audio file. When the F-score of the algorithm is compared to that of the random baseline, it is clear that this algorithm provides a vast improvement. The last speaker listed in Table 2 has a relatively low F-score: 0.58 compared to around 0.7. It is noteworthy that the total number of possible segmentation points for this speaker is significantly higher (over 8000), indicating that he spoke significantly slower than the others, hindering a completely accurate calculation of precision and recall. Even in this situation, however, the algorithm produces significantly better results than the random baseline.

In the final experiment, we concatenated all sentences from all speakers to create a single input file for each dialect region. The results in Table 3 indicate that our approach outperforms the baseline yet again and is robust with respect to noisy speech data from multiple speakers and genders. These are results shown for all the dialect regions given by the data set. Results from this test are only slightly lower than the trials run with individual speakers, demonstrating that this approach works well with multi-speaker speech. As a whole, at around 0.7, the results from these experiments are only slightly below the F-scores given by the algorithm when used on text [6] (the medium for which it was created) and they significantly outperform the baseline.

Dialect Region		Seg. Locations		Algorithm			Baseline
Region ID	Speakers	Target	Possible	Precision	Recall	F_1	Random F_1
New England	38	14399	230380	0.5993	0.8342	0.6975	0.4443
Northern	76	29158	459015	0.6051	0.7927	0.6863	0.4597
North Midland	76	28869	458720	0.6117	0.7942	0.6911	0.4541
South Midland	68	26093	425841	0.5926	0.7993	0.6806	0.4438
Southern	70	27117	453515	0.5871	0.8115	0.6813	0.4326
New York City	35	13395	219075	0.5945	0.8541	0.7010	0.4341
Western	77	29707	464298	0.6074	0.8075	0.6933	0.4588
Army Brat	22	8342	127728	0.6309	0.8661	0.7301	0.4722

Table 3: Results from 8 TIMIT Dialect Regions

This approach, then, allows us to utilize a text segmentation unsupervised algorithm on discretized speech in order to discover phoneme boundaries. The algorithm is language independent, and requires little manual preprocessing, while still producing results comparable to those from the text domain. Future work includes testing the algorithm on larger numbers of speakers and different languages, clustering the output of VQ to build up atomic units of speech, and tailoring the Voting Experts algorithm to our needs by adding experts (such as a prosody expert) that take advantage of acoustic research.

References

[1] Tom Armstrong and Tim Oates. Riptide: Segmenting data using multiple resolutions. In *Proceedings of the 6th IEEE International Conference on Development and Learning*, 2007.

[2] Tom Armstrong and Tim Oates. Undertow: Multi-level segmentation of real-valued time series. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI)*, pages 1842–1843, 2007.

[3] Guido Aversano, Anna Esposito, Antonietta Esposito, and Maria Marinaro. A new text-independent method for phoneme segmentation. In *Midwest Symposium on Circuits and Systems*, volume 2, pages 516–519. IEEE, 2001.

[4] PPG Boersma. Praat, a system for doing phonetics by computer. *Glot international*, 5(9/10):341–345, 2002.

[5] Paul Cohen, Niall Adams, and Brent Heeringa. Voting experts: An unsupervised algorithm for segmenting sequences. *Intell. Data Anal.*, 11(6):607–625, December 2007.

[6] Paul R. Cohen, Brent Heeringa, and Niall M. Adams. Unsupervised segmentation of categorical time series into episodes. In *ICDM*, pages 99–106, 2002.

[7] John Garofolo, Lori Lamel, William Fisher, Jonathan Fiscus, David Pallett, Nancy Dahlgren, and Victor Zue. DARPA TIMIT acoustic phonetic continuous speech corpus cdrom. *NTIS order number PB91-100354*, 1993.

[8] T. Kinnunen, I. Kärkkäinen, and P. Fränti. Is speech data clustered?-statistical analysis of cepstral features. In *Seventh European Conference on Speech Communication and Technology*. Citeseer, 2001.

[9] Matthew Miller and Alexander Stoytchev. An unsupervised model of infant acoustic speech segmentation. In *Proceedings of the International Conference on Epigenetic Robotics*, 2009.

[10] Odette Scharenborg, Mirjam Ernestus, and Vincent Wan. Segmentation of speech: Child's play? In *INTERSPEECH*, pages 1953–1956, 2007.