Unsupervised Discovery of Phoneme Boundaries in Multi-Speaker Continuous Speech

Tom Armstrong and Stephanie Antetomaso
Wheaton College
Norton, Massachusetts 02766
{tarmstro|antetomaso_stephanie}@wheatoncollege.edu

***Abstract*—Children rapidly learn the inventory of phonemes used in their native tongues. Computational approaches to learning phoneme boundaries from speech data do not yet reach the level of human performance. We present an algorithm that operates on, qualitatively, similar data to those children receive: natural language utterances from multiple speakers. Our algorithm is unsupervised and discovers phoneme boundary positions in speech. The approach draws inspiration from the word and text segmentation literature. To demonstrate the efficacy of our algorithm on speech data, we present empirical results of our method using the TIMIT data set. Our method achieves F-measure scores in the $0.68 - 0.73$ range for locating phoneme boundary positions.**

I. Introduction

Speech data are abundant and language technologies provide a natural interface between humans and machines. Representing speech at the phonemic level is useful for applications in speaker modeling, speech recognition, and text-to-speech. Manually annotating speech at the phonemic level is an onerous task and not feasible for the amount of speech data available. Yet, this is one skill where children demonstrate surprising abilities in the first year or so of life [1], [2]. We are interested in approaching the problem of acquiring language from a developmental perspective (i.e., unsupervised learning) for use on a mobile robot. Our ultimate goal is to endow a robot with a suite of algorithms that enable the acquisition of language to match human performance with similar inputs. Our approach is amenable to underrepresented languages or for use on novel speaker data without the need for annotation. Toward this goal, we present an unsupervised algorithm for the automatic segmentation of speech into a sequence of phonemes.

The remainder of this paper is organized as follows. Section 2 describes feature extraction from speech and contrasting approaches to linguistic unit (e.g., words, phonemes) discovery from speech data. Next, Section 3 details our algorithm and the parameters selected to improve learning. Then, Section 4 contains three experiments to evaluate our algorithm on an established dataset. Finally, we conclude and point to future work.

II. Background & Related Work

Our approach ultimately employs a text processing-inspired algorithm that is applied to speech data, and we need a discretization method. Our approach uses Vector Quantization (VQ) to discretize the input speech, and employs Voting Experts (VE) to segment the resulting discrete speech representation. VQ is, traditionally, a compression method which we can use to take our speech signal and find k prototype vectors, or codebook entries, where k is an input parameter. To discretize the input, each data point is replaced with its corresponding codebook entry. VE is designed for the segmentation of categorical data based on votes made by one or more experts [3]. VE finds frequent segments, or episodes, that are internally coherent. That is, within segments the predictability of information is high and predictability at segment boundaries is low. A frequency expert votes for positions in the data that maximize segment counts, and an entropy expert votes for positions where the next character in the sequence is hard to predict.

The output of VE is the original corpus segmented into chunks (e.g., words) that results in an implicit lexicon. Other approaches to segmenting or chunking strings take a compression approach, and also focus on natural language texts [4], [5], [6], [7].

A. Phoneme & Word Discovery in Speech

Traditionally, unsupervised phoneme discovery in speech has focused on transition points for acoustic features to make change point decisions [8], [9]. One approach considers a bottom-up, agglomerative algorithm that merges speech frames into larger chunks [10]. But, to improve the performance of their measures, they pass the maximum number of phonemes in a sentence to the algorithm. They report accuracy scores below 80% on biphone segmentation with a 20ms window. Supervised approaches fare better with accuracies reported in the mid-90% on similar tasks using HMMs and a 70ms window [11].

Applications of VE to speech have found success at discovering word boundaries from speech data [12] building on inducing hierarchies in time series data through multiple iterations of VE [13], [14]. For direct comparisons of algorithms, however, these approaches tend to evaluate their approaches on small amount of data (e.g., few sentences or speech with a single segment) [10] or single speakers [15], or used synthesized speech [12].

978-1-61284-990-4/11/$26.00 ©2011 IEEE

B. Word Discovery in Text

The inspiration for this work is drawn from the space of text-based chunking algorithms. MK10E was the first in a series of "learning as compression" contributions introduced by Wolff over the past few decades [4]. MK10E builds a tree by chunking frequently occurring bigrams and iterating over the data multiple times, thus compressing the data and forming a hierarchy. The algorithms does not, however, generalize. MK10E has been shown to compress, effectively, natural language data to discover word boundaries.

SEQUITUR extends the MK10E model by making more judicious decisions on where and what to chunk (i.e., how to compress) in an input string [5]. SEQUITUR also lacks generalization and only operates on a single string at a time. Two constraints drive the algorithm: bigram uniqueness and rule utility. Bigram uniqueness compresses the data by replacing two or more occurrences of a bigram with a non-terminal. Rule utility compresses the representation by enforcing that each non-terminal be used more than once. The algorithm is amenable to long strings with repeating substrings. Compressing the data using these two constraints finds frequently occurring subsequences and hierarchical structures of subsequences. SEQUITUR, however, does not generalize rules and therefore can only produce the input string from the induced rules. The approach's primary objective is compression, therefore a segment with one occurrence or few occurrences is of little interest as it cannot reduce the number of bits in the representation through a new rule or replacement with an old rule. Unfortunately, in many domains (e.g., natural language), single utterances may be too heterogeneous to compress.

BOOTLEX explicitly builds and rebuilds a lexicon in terms of itself by reparsing the input strings [6]. BOOTLEX requires an optimal word-length parameter and does not generalize or produce a hierarchy for the input strings. Other comparable algorithms look at the problem from a child language acquisition perspective [7], [16].

Our approach uses VOTING-EXPERTS which takes a less restrictive approach to finding segment boundaries [3]. Two experts, entropy and frequency, vote for segment boundaries based on predictability of the next substring and occurrences of a substring, respectively. The algorithm uses windowed data to find frequent segments, or episodes, that are internally coherent (i.e., predictability of information inside the segment is high and predictability at segment boundaries is low). A frequency expert votes for positions in the data that maximize segment counts and an entropy expert votes for positions where the next character in the sequence is hard to predict. They report the results of running VOTING-EXPERTS on the text of *1984* by George Orwell attempting to discover word boundaries that results in an F-measure of 0.76. The approach discovers meaningful segments using what they qualify as *domain independent features*.

III. ALGORITHM

Our approach begins with an input of raw speech data and discovers phoneme boundaries in speech – what we will consider atomic elements of natural language (see Figure 1 for an algorithm diagram). As presented below, we detail our approach and evaluate it on gold standard, established phoneme boundaries in large speech corpora rather than on synthesized data sets with only spot checked evaluation. To begin (Step 1 in Figure 1), we use Praat [17] to preprocess the speech into sequences of Mel-frequency cepstral coefficients (MFCC) – vectors of coefficients [18]. To create the sequence of vectors, we select a fixed-window size of 15ms and a step size of 5ms. This representation choice follows established practices in speech processing [19].

After preprocessing the speech data, VQ builds a codebook of subphonemic (i.e., speech segments shorter in length than a phoneme) prototypes all labeled with unique, random string labels. The size of the codebook is a function of the number of phonemes in the language with most languages having fewer than 50 phonemes. Empirical results indicate using a codebook of size $2-5$ times the number of phonemes in the language. Next (Step 2), using the VQ codebook, each vector is replaced with its closest codebook label generating the VE input. Then (Step 3), we run VE with a window size of 3 and entropy threshold of 4 using the frequency and entropy experts. Finally (Step 4), mapping back to the original speech data, VE outputs phoneme segmentation locations. At this point we can compare the discovered phoneme boundary positions against the gold standard.

IV. EXPERIMENTS & RESULTS

To evaluate the efficacy of our approach at discovering phoneme boundaries in continuous speech with an unrestricted vocabulary, we developed a series of three experiments. First, through empirical trials, we determined a range of codebook sizes to use for speech discretization in the subsequent experiments. Second, we evaluated our approach on speech data from single speakers, and we report results on 15 individual speakers. Third, we evaluated our approach on speech data from a collection of multiple speakers, and we report results on 463 speakers.

The speaker data are drawn from the TIMIT corpus of speech data [20]. TIMIT consists of 16kHz recordings of native English speakers (both male and female speakers). Each speaker directory contains recordings of ten sentences (selected for their broad coverage of phonemes in English). All ten sentences are distinct for an individual speaker, but overlap exists between speakers. That is, some sentences are repeated in different speaker directories.

The corpus is split, first, into training and testing sets. Then, each of the training and testing sets are further divided into dialect regions (total of 8) and individual speakers ($22-77$ speakers per region). The dialect regions specified are referred to as: 1) New England; 2) Northern; 3) North Midland; 4) South Midland; 5) Southern; 6) New York City; 7) Western; and 8) Army Brat (see Table III for the number of speakers used from each dialect region).

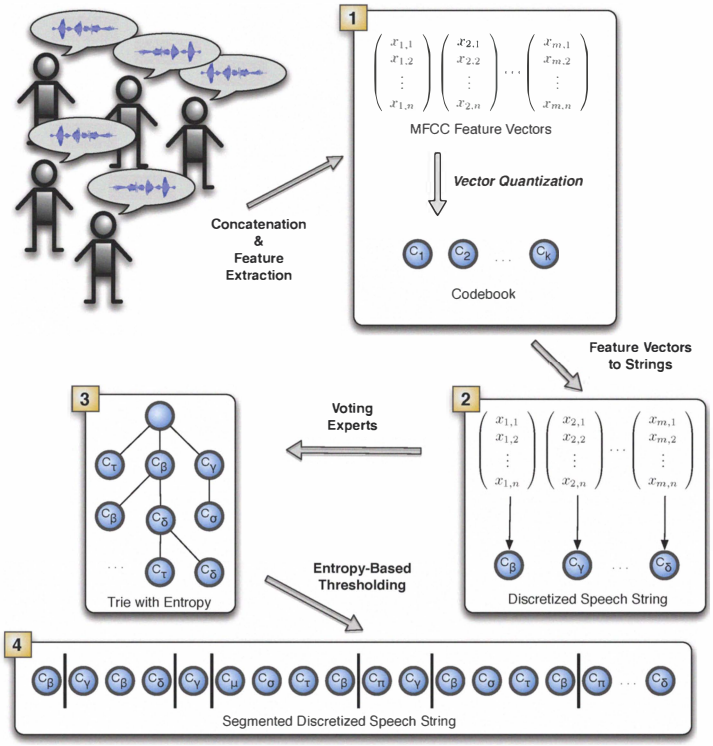

Fig. 1. Algorithm Diagram

A. Baseline & Evaluation

Each TIMIT speech file is paired with corresponding data detailing phoneme boundary positions. We use the included timestamps for phoneme boundaries as a gold standard – these locations have been manually annotated. The data also include labels for each phoneme, timestamps for word boundaries, and word transcriptions.

As a sanity check and baseline for our algorithm, we compare against a random strategy of selecting phoneme boundaries. That is, using the target (correct) number of boundary locations from the TIMIT data, the random approach selects that number of boundary positions in the data. This strategy is done without replacement.

Evaluation of Proposed Boundary Positions: To evaluate whether or not a detected phoneme boundary position is correct, we use windowed approach. If a boundary position falls within 20ms of a target position, then it is a true positive boundary position. The 20ms window is the standard window size in evaluating correctness of proposed phoneme boundary positions [19]. On data consisting of 38 speakers from the New England dialect region, there are 230380 possible locations for phoneme boundaries. The total number of correct boundaries is 14399. The F-score on this data for the baseline algorithm is 0.45 (N.B.: many phoneme segmentation algorithms present only *recall* results. However, recall may be maximized by returning all possible segmentation locations.).

	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 I
PHONEME SEGMENTATION RESULTS FOR ALL NEW ENGLAND SPEAKER DATA (MAXIMUM VALUES ARE IN BOLD)

B. Comparison of VQ Codebook Size on Phoneme Boundary Detection

The number of distinct phonemes does not exceed 50 in most natural languages. The average length of a phoneme in time ($>$ 50ms) exceeds the window used to process the original speech into feature vectors (15ms). The shortened window length seeks to minimize the overlap between consecutive phonemes and capture coarticulation effects between consecutive phonemes. The number of codebook entries should not be below 50 and should not be too large as to result in a unique entry for each feature vector. Using the speech from all New England speaker's training data, we evaluated our algorithm with three different values for codebook size ($k = 122, 155, 244$). The results for each of the three trials appears in Table I. Maximum values for precision, recall, and F-measure are highlighted in bold for our algorithm and the baseline.

	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 II
EXPERIMENTAL RESULTS FROM 15 TIMIT SPEAKERS

In our experience, for smaller and larger values of k, the results are no better than those for the three values tested. Given the fairly similar results in Table I, we selected a codebook size of 155 for all other experiments. The baseline algorithm also performed consistent with itself for the three values of k and equivalently worse than our approach for the three values of k.

C. Detecting Phoneme Boundaries in Single-Speaker Continuous Speech

For this experiment and the multi-speaker experiment, we use the empirically determined codebook size of 155 in all trials. In this experiment, we restrict the data to those of single speakers. We selected 7 speakers from the first dialect region (DR1), or New England, and 8 speakers from the fourth dialect region (DR4), or South Midland. For each speaker, we concatenated each audio file together (10 sentences per speaker from the training data folders) into a continuous stream and adjusted the gold standard timestamps accordingly.

Table II contains the results for these 15 speakers. Each row contains the TIMIT speaker ID prepended with the dialect region directory name. For each large speech data file, the number of possible segmentation positions lies in the range $4809-8345$ with target positions numbering in the range $346-414$. Maximum and minimum values for precision, recall, and F-measure are highlighted in bold for our algorithm and the baseline.

D. Detecting Phoneme Boundaries in Multiple-Speaker Continuous Speech

Analogous to the single-speaker continuous speech experiment, in this experiment we use the empirically determined codebook size of 155 in all trials. In the single-speaker experiment, each trial consisted of ten sentences for each speaker. In the multiple-speaker experiment, each trial encompasses an entire dialect region's training data. For each dialect region, we concatenated each of the ten sentence audio files together per speaker. Then, we concatenated each speakers' data together into a continuous stream and adjusted the gold standard timestamps accordingly. Not all dialect regions have the same number of speakers.

Table III contains the results for these 8 dialect regions. Each row contains the TIMIT dialect region name and includes the speaker count (ranging from $22-77$, or $220-770$ sentences). For each large speech data file, the number of possible segmentation positions lies in the range $127728-464298$ with target positions numbering in the range $8342-29158$. We report precision, recall, and F-measure for our algorithm and the baseline.

V. DISCUSSION & CONCLUSION

The experimental results presented in the previous section's tables demonstrate the utility of a text-based segmentation approach on discretized speech. The F-measure for the 15 speakers for our algorithm consistently perform around the 0.7 level. In the multi-speaker context, F-measures for the 8 dialect regions ranged from $0.68-0.73$ compared to a random baseline of $0.43-0.47$. All but one of the individual speakers in the New England dialect region outperformed the multi-speaker dialect region data set. We propose that this is to be expected given the multiple speakers, multiple genders, and two orders of magnitude increase in potential segmentation locations.

In sum, we presented an approach to using text-based segmentation algorithms on discretized speech to discover phonemes. The approach is unsupervised and language agnostic. Unlike prior approaches, we consider a phoneme-level segmentation. Discovering phonemes, in contrast to word boundary finding, in continuous speech has specific applications in speaker modeling (e.g., building speaker-specific models for language production and understanding). We presented empirical results on a standard data set with established evaluation criteria. Our results demonstrate an efficient approach to discovering phonemes in speech data.

Future work will proceed in several directions. First, the TIMIT data also contain phoneme labels for all speakers and

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 III
RESULTS FROM 8 TIMIT DIALECT REGIONS

dialect regions. We will evaluate phoneme clustering algorithms and apply them to our discovered phonemes. Second, the VE framework is designed for the addition of domain-specific or more general *experts*. We will explore the space of possible experts that are amenable to the speech domain. Finally, we will consider alternative discretization techniques and compare those results with our approach.

REFERENCES

[1] P. W. Jusczyk, "Investigations of the word segmentation abilities of infants," in *Proceedings of the Fourth International Conference on Spoken Language Processing ICSLP*, vol. 3, Philadelphia, PA, 1996, pp. 1561–1564.

[2] E. V. Clark, *First Language Acquisition*. New York: Cambridge University Press, 2009. [Online]. Available: get-book.cfm?BookID=39320

[3] P. R. Cohen, B. Heeringa, and N. M. Adams, "Unsupervised segmentation of categorical time series into episodes." in *ICDM*, 2002, pp. 99–106.

[4] J. G. Wolff, "An algorithm for the segmentation of an artificial language analogue," *British Journal of Psychology*, vol. 66, pp. 79–90, 1975.

[5] C. G. Nevill-Manning and I. H. Witten, "Identifying hierarchical structure in sequences: A linear-time algorithm," *Journal of Artificial Intelligence Research*, vol. 7, p. 67, 1997. [Online]. Available: http://www.citebase.org/cgi-bin/citations?id=oai:arXiv.org:cs/9709102

[6] E. O. Batchelder, "Bootstrapping the lexicon: A computational model of infant speech segmentation," *Cognition*, vol. 83, no. 2, pp. 167–202, 2002.

[7] M. Brent, "An efficient, probabilistically sound algorithm for segmentation and word discovery," *Machine Learning*, vol. 34, pp. 71–105, 1999.

[8] G. Aversano, A. Esposito, A. Esposito, and M. Marinaro, "A new text-independent method for phoneme segmentation," in *Midwest Symposium on Circuits and Systems*, vol. 2. IEEE, 2001, pp. 516–519.

[9] O. Scharenborg, M. Ernestus, and V. Wan, "Segmentation of speech: Childs play?" in *INTERSPEECH*, 2007, pp. 1953–1956.

[10] Y. Qiao, N. Shimomura, and N. Minematsu, "Unsupervised optimal phoneme segmentation: Objectives, algorithm and comparisons," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2008, pp. 3989–3992.

[11] X. R. Pierre Lanchantin, Andrew C. Morris and C. Veaux, "Automatic phoneme segmentation with relaxed textual constraints," in *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*. Marrakech, Morocco: European Language Resources Association (ELRA), may 2008, http://www.lrec-conf.org/proceedings/lrec2008/.

[12] M. Miller and A. Stoytchev, "An unsupervised model of infant acoustic speech segmentation," in *Proceedings of the International Conference on Epigenetic Robotics*, 2009.

[13] T. Armstrong and T. Oates, "Riptide: Segmenting data using multiple resolutions," in *Proceedings of the 6th IEEE International Conference on Development and Learning*, 2007.

[14] ——, "Undertow: Multi-level segmentation of real-valued time series," in *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI)*, 2007, pp. 1842–1843.

[15] K. Gold and B. Scassellati, "Audio speech segmentation without language-specific knowledge," in *Proceedings of the 28th Annual Conference of the Cognitive Science Society*, 2006. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=?doi=10.1.1.122.1995

[16] J. Hammerton, "Learning to segment speech with self-organising maps," *Language and Computers*, vol. Computational Linguistics in the Netherlands, no. 14, pp. 51–64, 2002.

[17] P. Boersma, "Praat, a system for doing phonetics by computer," *Glot international*, vol. 5, no. 9/10, pp. 341–345, 2002.

[18] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 4, pp. 357–366, 2003.

[19] 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.

[20] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, N. Dahlgren, and V. Zue, "DARPA TIMIT acoustic phonetic continuous speech corpus cdrom," *NTIS order number PB91-100354*, 1993.