A PARALLELIZED DYNAMIC PROGRAMMING APPROACH TO ZERO RESOURCE SPOKEN TERM DISCOVERY

Bradley Oosterveld[1]*, Richard Veale*[2]*, Matthias Scheutz*[1]

[1] Human-Robot Interaction Laboratory, Tufts University, USA
[2]Department of Neurophysiology, Kyoto University, Kyoto, Japan

bradley.oosterveld@tufts.edu, veale.richard.62c@st.kyoto-u.ac.jp, matthias.scheutz@tufts.edu

ABSTRACT

Zero resource spoken term discovery in continuous speech is the discovery of repeated patterns in acoustic signals without any higher level linguistic information. These patterns are then combined to define the compositional units of that speech. We describe and implement an algorithm that tags similar subsequences among sequences of acoustic features. We then discuss the use of this algorithm as part of a complete spoken term discovery system. Our implementation leverages parallelization via modern GPUs, allowing many independent comparisons to be executed concurrently. This parallelization enables the described system to analyze large data sets in tractable time frames. The accuracy and performance of our approach are compared to existing approaches as well as human transcriptions on two corpora of continuous natural speech. Our system improved on published results for multiple metrics.

Index Terms: spoken term discovery, zero resource speech segmentation, similarity measures, GPU computing

1. INTRODUCTION

Traditionally, the goal of speech recognition is to classify segments of speech into preexisting categories (words, phrases, etc.). Thus, traditional speech recognizers must be trained using a combination of speech examples and the corresponding linguistic categories. In contrast, in the zero resource setting, the raw speech training data is not accompanied by any linguistic information. Thus, the linguistic categories themselves must be estimated during training [1]. Methods for discovering linguistic categories are valuable both for their potential application, and their theoretical insight. In extreme cases such as aboriginal languages where there is a limited amount of transcribed data for a language, spoken term discovery mechanisms can even lay the foundation for the development of further speech processing technologies [2, 3, 4]. Additionally, humans are faced with the task of zero resource spoken term discovery in their infancy, and insights about effective artificial methods may aid in the development of models of human cognition [5].

The task of spoken term discovery in continuous speech is described in [6], which proposed a method called segmental Dynamic Time Warping (S-DTW). The high level structure of the S-DTW approach has been retained by subsequent methods [7, 8, 9]. Repeated subsequences in the acoustic feature space are discovered in pairs. Similar sets of these pairs are then grouped into larger clusters. These clusters are used to define linguistic units and create a transcription of the data from which they were derived.

This work was funded in part by US Office of Naval Research grant #N00014-14-1-0751. We thank Roger K Moore for the helpful discussion.

Discovering repeated patterns in continuous natural speech requires the ability to recognize which parts of a pair of speech sequence are similiar and which are different. *Segmental-Dynamic Time Warping* (S-DTW) [6] is used in many spoken term discovery approaches because it can find alignments of common subsequences in pairs of feature sequences [7, 8, 9]. The *Acoustic DP-Ngram algorithm* (DP-Ngrams) is another method for finding common subsequences [10]. Both are based on the dynamic programming algorithm *Dynamic Time Warping* [11]. While S-DTW determines predefined start and end points for comparison and then finds the most similar areas along those predefined paths [6, 12], DP-Ngrams calculates the similarity of all possible alignments and then determines alignment boundaries based on changes in similarity among neighboring elements [10]. Previously DP-Ngrams has been used to discover sub-word units, focusing on fine differences in data sets that contain many examples of a small set of words [10, 13]. In our approach it is modified to focus on coarser differences, discovering word/phrase level units in data sets which have large numbers of categories with few members.

We present an extension of the DP-Ngrams designed for use in place of S-DTW in zero resource spoken term discovery systems. Additionally, we propose a novel implementation that improves performance, enabling use in real-time settings. Naively finding all repeated subsequences in a stream of continuous speech requires comparing all subsequences to all subsequences, requiring $O(N^2)$ comparisons. This complexity becomes prohibitive as more and more subsequences are detected. Our approach does not aim to reduce this complexity, as was done in [7], but rather increases the efficiency of comparisons by exploiting the inherent parallelism present in the task. At high level, we leverage the assumption that no discovered linguistic unit will contain silence of more than a certain duration [6]. Because of this we can treat continuous speech as a sequence of smaller segments, delimited by silence. Pairs of these segments can be compared independently and thus in parallel. At the implementation level, our approach parallelizes previously serial calculations within DP-Ngrams. These uses of parallelism greatly reduce the overall runtime compared to a serial approach, reducing the quadratic runtime by a factor proportional to the number of GPU threads. To avoid confounding the impact of our algorithm, we opted not to implement the complexity reductions presented in [7]. The focus of this work is to evaluate the effectiveness of our extension of DP-Ngrams in the zero resource spoken term discovery context.

2. METHODS

The proposed system segments raw input data multiple times, starting with cheap heuristics and then increasing computational com-

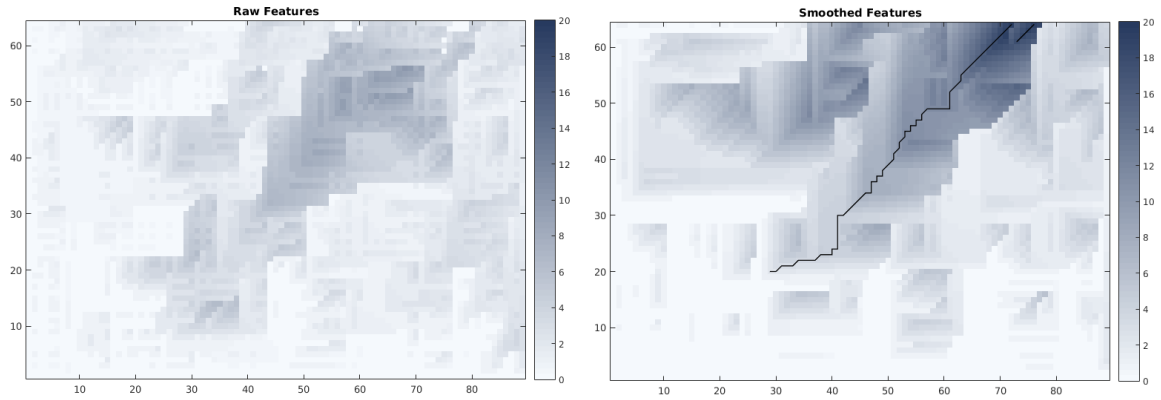

Fig. 1. *A comparison of quality matrices for two feature sequecnes with raw and smoothed features. Higher quality values represent more similar sequences. Discovered local alignment paths shown in black.*

plexity. The first segmentation is done on the raw audio data, defining segment boundaries based on regions of silence. Next, we extract acoustic feature vectors from the audio stream, and then perform subsequence discovery on these acoustic feature vectors. Sets of similar subsequence feature vectors are grouped together to form clusters which represent discovered linguistic units. The boundaries of these discovered units are combined with the boundaries from the initial segmentation to form a complete transcription of the input data. We next present an overview of the system and the most important parameters.

2.1. Initial Segmentation

The goal of the initial segmentation is to cheaply split the raw data into a set of smaller segments that can be used for parallelized comparisons. To ensure the independence of these parallel comparisons the boundaries defined by this segmentation must not split any linguistic units. We operate under the assumption that regions of silence do not contain or span any linguistic units [6].

We use an amplitude envelope filter to cheaply detect regions of silence in the raw PCM audio data. The signal is full wave rectified and averaged using a sliding window function. If the running average drops below a threshold value for more than a specified duration, a boundary is drawn. The next segment start boundary is created when the average amplitude next crosses the threshold. To ensure no data is unintentionally lost, the discovered segment start and end are padded with a constant number of frames. This implementation is simpler than the envelop filtering done in other approaches such as [9], but at this stage we favor speed over precision. This segmentation cheaply splits the raw data into smaller units for processing in parallel and removes redundant computations that would result from comparing two sequences of silence.

2.2. Feature Extraction

From the segments generated by the initial segmentation we extract MFCCs with first and second differentials using the standard HTK implementation [14]. After the features are extracted, the feature vectors are smoothed using a sliding average with a window size of 40 ms, where each feature represents 10 ms of the original signal. We found that this smoothing greatly increased accuracy of the subsequence comparison phase of the segmentation. The effect of minor dissimilarities between feature sequences is reduced, allowing for more robust performance. A visualization of the effects of this smoothing can be seen in *Figure 1*.

2.3. Subsequence Discovery

Common subsequences in pairs of feature sequences are discovered using a modified version of the subsequence alignment mechanism of DP-Ngrams which is defined in its entirety in [10]. This section contains an overview of the algorithm with emphasis on our performance modifications and our novel implementation.

The algorithm compares two feature vectors U_m, V_n by first creating a *distance matrix*, $D_{m,n}$, composed of the *cosine similarity* between every pair of elements in U and V. Cosine similarity is used to normalize the range of scores to $[-1, 1]$.

$$D_{i,j} = \frac{U_i \cdot V_j}{\| U_i \| \ \| V_j \|}$$

Using the values of D a *quality matrix*, $Q_{m-1,n-1}$, is calculated. The elements of Q hold quality scores which represent the similarity of two elements form U and V weighted by the similarity of their neighbors. The relation between Q and D is defined:

$$Q_{i,j} = max \begin{cases} Q_{i-1,j-1} + b \cdot D_{i,j} \\ Q_{i,j-1} + (p \cdot D_{i,j-1}) \cdot q_{i,j-1} \\ Q_{i-1,j} + (p \cdot D_{i-1,j}) \cdot q_{i-1,j} \\ 0 \end{cases} \quad (1)$$

where b is a positive bonus weight and p is a negative penalty weight. These weights constrain the amount which the alignment can be distorted. Our system uses values of $b = 1$ and $p = -.1$ (note that the penalty value is less than the value in [10], minimizing the effect of the insertion or deletion of frames). This parametrization allows alignments to account for more variation in the sequences of the data.

As Q is calculated a third matrix, the *backtracking matrix* $B_{m-1,n-1}$, is also calculated. $B_{i,j}$ stores the case that had the max value in the calculation of $Q_{i,j}$.

Local alignments are discovered in a quality matrix if $max(Q) > qThreshold$. Alignment discovery starts at $max(Q)_{i,j}$ where $max(Q)_i$ represents the endpoint of the common subsequence in U, and $max(Q)_j$ represents its end point in V. Elements are appended to the alignment by backtracking through Q using the indexes stored in B until $Q_{B_i,B_j} < minThresh$ (we use implementation $minThresh = 0.05 \cdot max(Q)$ and $qThreshold = 19$).

When the minimum threshold has been reached the alignment is stored. The elements in Q that the alignment spanned are set to low quality values so they are excluded from future searches. Thus, when alignment spans $[Q_{i,j}, Q_{i+u,j+v}]$ (where u and v represent the subsequence length in number of rows and number of columns, respectively), all matrix entries in Q within that region are set to -1.

The removal of these elements prevents the discovery of overlapping and sub-optimal alignments. Our implementation removes more elements per alignment than the original [10] with the intent of reducing the total number of computations and eliminating the occurrence of redundant alignments. This change was motivated by our more generous warping parameters and use of smoothed features, which caused an increase in the number of redundant alignments. New alignments are sought while $max(Q) \leq qThreshold$. Alignments shorter than 300 ms are excluded from final results to reduce the effects of feature averaging on short duration sounds.

2.4. Parallelization

The three matrices created in the subsequence discovery process are composed of elements derived through many independent operations. In the calculation of *D* the values of all of the elements can be computed in parallel. In *Q* the calculation of an element $Q_{i,j}$ depends on the values of three previously calculated elements ($Q_{i-1,j}$ $Q_{i,j-1}$ $Q_{i-1,j-1}$), so as elements in *Q* are calculated subsequent groups of elements can be calculated in parallel. A depiction of the topography of this relationship is shown in *Figure 2*. An element in *B*, $B_{i,j}$ depends on $Q_{i,j}$ from the same comparison, so the calculation of *B* follows the same topography as *Q*.

Since the segments being compared are small because of the initial segmentation, and we know that discovered units will span more than one segment, we can efficiently fill GPU thread blocks, limiting each sequence comparison to a single block. With blocks performing roughly the same number of comparisons we minimize situations where low percentages of GPU computation capacity are being utilized. By containing the entirety of a given comparison within a single block we minimize memory transfer.

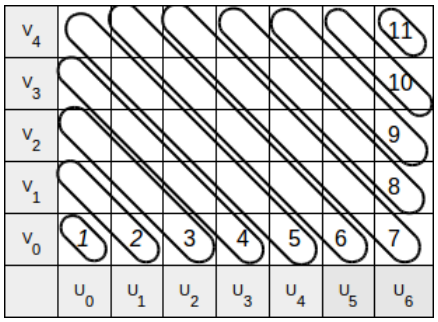

Fig. 2. *Topography of Parallel Calculations: Elements in the same group may be calculated in parallel, but element groups must be calculated sequentially.*

2.5. Clustering and Boundary Refinement

We consider groups pairs of subsequences with a common member using the clustering algorithms described in [12, 6, 7]. In cases where multiple discovered alignments in one cluster come from the same region of the original data, their start and end indices are averaged. Each alignment is only placed into a single cluster. If there is a conflict the cluster with higher similarity is chosen.

Subsequence discovery and alignment clustering define the set of linguistic units that are repeated in the data. The boundaries of the units in this set are combined with the boundaries generated by the initial segmentation. Removing their intersection leaves the set of segments which represent sound that is not repeated elsewhere in the data. Each of these previously uncategorized segments is assigned to its own category in the final transcription.

3. EVALUATION

[15] defines a combination of evaluation metrics and data sets geared toward the development of spoken term discovery systems. The existence of these standardized metrics and data sets allows for specific components of term discovery systems to be evaluated, and provides reference points for their performance.

3.1. Data

[15] provides two corpora. One is in American English, a language that has large amounts of transcribed data. The other is in the African language Tsonga, for which there exists very little transcribed data. The English corpus is composed of casual conversations selected from [16]. There are 12 speakers (6 male, 6 female; 6 young, 6 old), for a total of 634 minutes of recordings. The Tsonga corpus is composed of read speech from 24 (12 male, 12 female) speakers, totaling 444 minutes, selected from [17].

3.2. Metrics

The toolkit described in [18] contains metrics for the evaluation of various parts of term discovery systems. In general, *discovered* linguistic unit boundaries are compared with a *gold* set of units generated by human transcription.

The toolkit uses the following metrics: *Matching* measures how well the system can find similar sequences of speech across the whole corpus. *Normalized Edit Distance (NED)* is a more generalized measure of similarity, it equals 0 when all of the elements in a sequence are the same, and 1 when they are completely different. *Coverage* represents the percentage of the complete set of matching pairs that is present in the discovered set. *Grouping* measures the homogeneity of the discovered clusters. *Type* also measures cluster homogeneity, but only includes clusters that are completely present in the gold set. *Token* scores compare discovered segment boundaries to gold set boundaries. *Boundary* measures the amount of discovered gold unit boundaries. Each metric (excluding *NED* and *Coverage*) is composed of three scores: *Precision*, *Recall* and *F-Score*.

$$Precision(discovered, gold) = \frac{\mid discovered \cap gold \mid}{\mid discovered \mid}$$

$$Recall(discovered, gold) = \frac{\mid discovered \cap gold \mid}{\mid gold \mid}$$

$$F - Score(discovered, gold) = \frac{2 \times discovered \times gold}{discovered + gold}$$

3.3. Results

Table 2.5 contains the results of the toolkit evaluation described in 3 conducted on our system, baseline [15, 7], and reported scores [8, 9]. Topline scores from human transcribed data from [19] are included as a reference for the best possible performance in a given category.

	NED	Cov	Matching			Grouping			Type			Token			Boundary		
			P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
	English																
T	0.0	100	98.3	18.5	31.1	99.5	100	99.7	50.3	56.2	53.1	68.2	60.8	64.3	88.4	86.7	87.5
B	**21.9**	16.3	39.4	1.6	3.1	21.4	84.6	33.3	6.2	1.9	2.9	5.5	0.4	0.8	44.1	4.7	8.6
R	70.8	42.4				13.4	15.7	14.2	**14.1**	**12.9**	**13.5**	**22.6**	**6.1**	**9.6**	**75.7**	33.7	**46.7**
L	61.2	80.2	6.5	**3.5**	**4.6**				3.1	9.2	4.6	2.4	3.5	2.8	35.4	**38.5**	36.9
O	39.4	**92.1**	**51.8**	0.0	0.0	**76.2**	**100**	**82.7**	5.6	5.1	5.3	10.2	1.9	3.2	71.1	22.5	34.2
	Tsonga																
T	0.0	100	100	6.8	12.7	100	100	100	15.1	18.1	16.5	34.1	49.7	40.4	66.6	91.9	77.2
B	**12.0**	16.2	**69.1**	0.3	0.5	**52.1**	77.4	**62.2**	3.2	1.4	2.0	**2.6**	0.5	0.8	22.3	5.6	8.9
R	63.1	94.7				10.7	3.3	5.0	2.2	6.2	3.3	2.3	3.4	**2.7**	29.2	39.4	33.5
L	43.2	89.4	21.2	**3.8**	**6.5**				**4.9**	**18.8**	**7.8**	2.2	**12.6**	0.8	18.8	**64.0**	29.0
O	39.6	**95.5**	35.7	0.0	0.0	19.1	**100**	31.7	1.6	2.2	1.9	1.5	0.5	0.8	**49.9**	27.6	**35.5**

Table 1. *Selected results from spoken term discovery systems: topline (T) from human transcription, baseline (B) [15, 7], [8](L), [9](R), the system described in this paper (O). Notable scores in bold.*

To save space, only selected results are shown from [8, 9]. Specifically, the *oscillator* configuration from [9], and the *ConnComp-FDPLS* configuration from [8] were chosen because they contained the most improvement on the baseline scores.

For the *Grouping* metric our system shows improvements for both corpora. High recall scores here suggest that members of a given cluster are only present in that cluster, while the precision scores reflect how similar the clustered elements are. We also see improved results in the *Boundary* category for the Tsonga corpus, and relatively high results on the English corpus. Higher scores here suggest that many of the unit boundaries discovered by the system are also found in the human transcription.

More notable are the results from the matching related metrics (*NED*, *Coverage* and *Matching*) and the relationship between them. They highlight trade-offs between being able to discover a large number of units with limited accuracy and a small number of units very accurately. Our system produces the highest *Coverage* scores of all reported systems, meaning it discovers the largest portion of the set of repeated patterns. For both corpora our system has the second highest *NED*. Our system has the highest *Matching* precision on the English corpus, and second highest on the Tsonga corpus, with zero Recall and F-Scores in both cases. These results suggest that the discovered matches have high similarity, but do not directly correspond to the gold transcription. The baseline scores which have lower *NED* on both corpora and higher *Matching* precision on Tsonga, have slightly higher Recall and F-Scores and much lower *Coverage*. The other systems have lower *Coverage* and *Matching* scores, in addition higher *NED*. In comparison to these other reported results our system is able to discover large numbers of units while maintaining a relatively high level of precision.

4. DISCUSSION

Our system achieved significant improvements in performance over existing algorithms for several of the measures defined by the evaluation toolkit. However, as with all standardized corpus-based evaluations the risk of overfitting is ever-present. The algorithms discussed in this paper have a large number of user-defined parameters that determine the type and quality of the results they produce. The parameter values in our system were modified for performance on this specific task. The most relevant of these parameters were: feature smoothing window size, DP-Ngram quality threshold, and DP-Ngram bonus and penalty weights. We determined the effects of these parameters through the evaluation of a small number individual DP-Ngram comparisons where the ground truth was known. The results of our evaluation suggest that our parameter values are robust and apply across discovery domains, as evidenced by similar performance for English and Tsonga, two languages with different phonologies). Future work will extend the current evaluation to include additional languages and investigate the effects of various parameterizations on the overall characteristics of the results.

In terms of performance, our system took 346 minutes to generate the complete transcription of the English corpus which is approximately 634 minutes long and 67 minutes for 244 minutes of Tsonga data. These runtimes were recorded on a computer using 16 GB RAM, Intel Core i7-3820 CPU with 8 cores at 3.60GHz, and NVidia Titan X GPU with 3072 CUDA cores at 1.0GHz and 12 GB VRAM. The parallelized version of our implementation produced a $3200\times$ speedup over the serial CPU implementation.

Other studies have attempted to achieve speedups by improving other aspects of the computation. [8] sought improvements through graph clustering, and [9] through acoustic preprocessing before sequence comparison. [7] reduced the total number of sequence comparisons required during discovery. An ideal system would utilize improvements in all of these areas. Notably, our approach is the first to focus on improving the accuracy and speed of the comparisons themselves. The effectiveness of the algorithms presented in this paper hints at its potential in other settings where subsequence discovery is required. Possible applications include key word spotting and recognition of repeated out-of-vocabulary utterances.

5. CONCLUSION

We described an extension of DP-Ngrams using a novel implementation which allowed it to be applied to the new domain of zero resource spoken term discovery. It was thoroughly evaluated on two spoken language corpora using a standardized set of predefined metrics. The proposed algorithm performed better on several metrics than various state-of-the-art systems. The proposed algorithm and its GPU implementation are important steps towards enabling real-time term discovery which is important in many future agent-based applications (e.g., autonomous agents that have to learn new words on the fly). Future work will attempt to further improve the efficiency of the proposed algorithm (e.g., by including various approximation methods) without sacrificing its high level of accuracy.

6. REFERENCES

[1] J. Glass, "Towards unsupervised speech processing," in *Information Science, Signal Processing and their Applications (ISSPA), 2012 11th International Conference on*, July 2012, pp. 1–4.

[2] M.-h. Siu, H. Gish, A. Chan, W. Belfield, and S. Lowe, "Unsupervised training of an HMM-based self-organizing unit recognizer with applications to topic classification and keyword discovery," *Computer Speech & Language*, vol. 28, no. 1, pp. 210–223, 2014.

[3] R. Flamary, X. Anguera, and N. Oliver, "Spoken WordCloud: Clustering recurrent patterns in speech," in *Content-Based Multimedia Indexing (CBMI), 2011 9th International Workshop on*. IEEE, 2011, pp. 133–138.

[4] M. Dredze, A. Jansen, G. Coppersmith, and K. Church, "NLP on spoken documents without ASR," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2010, pp. 460–470.

[5] A. Jansen, E. Dupoux, S. Goldwater, M. Johnson, S. Khudanpur, K. Church, N. Feldman, H. Hermansky, F. Metze, R. C. Rose *et al.*, "A summary of the 2012 JHU CLSP workshop on zero resource speech technologies and models of early language acquisition." in *ICASSP*, 2013, pp. 8111–8115.

[6] A. S. Park and J. R. Glass, "Unsupervised pattern discovery in speech," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 1, pp. 186–197, 2008.

[7] A. Jansen and B. Van Durme, "Efficient spoken term discovery using randomized algorithms," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2011, pp. 401–406.

[8] V. Lyzinski, G. Sell, and A. Jansen, "An evaluation of graph clustering methods for unsupervised term discovery," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[9] O. Räsänen, G. Doyle, and M. C. Frank, "Unsupervised word discovery from speech using automatic segmentation into syllable-like units," *Manuscript submitted for publication*, 2015.

[10] G. Aimetti, R. K. Moore, and L. ten Bosch, "Discovering an optimal set of minimally contrasting acoustic speech units: A point of focus for whole-word pattern matching," 2010.

[11] M. Müller, "Dynamic time warping," *Information retrieval for music and motion*, pp. 69–84, 2007.

[12] A. Park and J. R. Glass, "Towards unsupervised pattern discovery in speech," in *Automatic Speech Recognition and Understanding, 2005 IEEE Workshop on*, Nov 2005, pp. 53–58.

[13] L. ten Bosch, O. J. Räsänen, J. Driesen, G. Aimetti, T. Altosaar, L. Boves, and A. Corns, "Do multiple caregivers speed up language acquisition?" in *INTERSPEECH*. Citeseer, 2009, pp. 704–707.

[14] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey *et al.*, *The HTK book*. Entropic Cambridge Research Laboratory Cambridge, 1997, vol. 2.

[15] B. Ludusan and M. Versteegh, "The zero resource speech challenge INTERSPEECH 2015- results," INTERSPEECH, 2015.

[16] M. A. Pitt, L. Dilley, K. Johnson, S. Kiesling, W. Raymond, E. Hume, and E. Fosler-Lussier, "Buckeye corpus of conversational speech (2nd release)," *Columbus, OH: Department of Psychology, Ohio State University*, 2007.

[17] N. J. De Vries, M. H. Davel, J. Badenhorst, W. D. Basson, F. De Wet, E. Barnard, and A. De Waal, "A smartphone-based ASR data collection tool for under-resourced languages," *Speech communication*, vol. 56, pp. 119–131, 2014.

[18] B. Ludusan, M. Versteegh, A. Jansen, G. Gravier, X.-N. Cao, M. Johnson, and E. Dupoux, "Bridging the gap between speech technology and natural language processing: an evaluation toolbox for term discovery systems," in *Language Resources and Evaluation Conference*, 2014.

[19] M. Versteegh, R. Thiolliere, T. Schatz, X. N. Cao, X. Anguera, A. Jansen, and E. Dupoux, "The zero resource speech challenge 2015," in *Proc. of INTERSPEECH*, 2015.