A computational model for unsupervised word discovery

Louis ten Bosch, Bert Cranen

Dept. Language and Speech, Radboud University, Nijmegen, the Netherlands
{l.tenbosch, b.cranen}@let.ru.nl

Abstract

We present an unsupervised algorithm for the discovery of words and word-like fragments from the speech signal, without using an upfront defined lexicon or acoustic phone models. The algorithm is based on a combination of acoustic pattern discovery, clustering, and temporal sequence learning. It exploits the acoustic similarity between multiple acoustic tokens of the same words or word-like fragments. In its current form, the algorithm is able to discover words in speech with low perplexity (connected digits). Although its performance still falls off compared to mainstream ASR approaches, the value of the algorithm is its potential to serve as a computational model in two research directions. First, the algorithm may lead to an approach for speech recognition that is fundamentally liberated from the modelling constraints in conventional ASR. Second, the proposed algorithm can be interpreted as a computational model of language acquisition that takes actual speech as input and is able to find words as 'emergent' properties from raw input.

Index Terms: speech analysis, pattern classification, pattern clustering methods, unsupervised learning, word discovery.

1. Introduction

The methodology of discovering words from the raw speech signal is an interesting issue in two different but conceptually related research areas. First, babies and young infants detect words from continuous speech. Psycholinguistic research ([13], [8] and references therein, [6a]) shows that babies can use the statistical cooccurrence of sound sequences as a cue for word segmentation. In this way, segmentation of the speech signal is possible when speech is presented as a single modality. When the input is multimodal (e.g. speech plus vision), experiments ([12], [11]) and computational models (such as the CELL model, [7b], and references therein) indicate that the process of word learning can be improved compared to learning from unimodal input.

Also for automatic speech recognition (ASR), techniques that learn to decode speech while avoiding the necessity of an upfront specified lexicon and phone models are interesting. Conventional methods for ASR are able to decode an unknown speech signal in terms of a sequence of predefined items from a closed vocabulary. Basically, the recognition of speech in terms of items outside the vocabulary is impossible. The classical limitations for defining and modelling words and phonemes in ASR might be radically reduced by exploring alternatives for data-driven word learning, for example by episodic approaches in ASR. In 'episodic' models of speech processing, the decoding of speech is facilitated by the availability in memory of explicit traces (episodes) of previously observed speech, in which a large amount of acoustic detail is stored ([4], [18]). It is therefore of considerable interest to investigate recognition approaches that circumvent the necessity of an a priori defined lexicon. This paper addresses one of such methods.

The focus of this paper is on the discovery of words and word-like speech fragments from multimodal input without using a lexicon and without any pre-existent phone-models. The multimodal input consists of a sequence of utterances in combination with abstract representations. Each utterance is associated with an abstract representation that indicates the presence of a certain word in the utterance. This representation does not specify information which word, the acoustic realisation of the word or its position in the utterance. This abstract modality can be compared to e.g. the a high-level abstraction of the visual modality during language learning. For example, it flags the visual presence of a physical 'ball' that is visible during the realisation of the utterance 'look at this nice ball'. In section 2.3 below, this modality is discussed in more detail.

Our word discovery method exploits two types of patterning in speech. First, the statistical properties of repetitive structure within the speech modality is used to hypothesize speech fragments and a labelling of these segments. Second, cross-modal associations are used to hypothesize words and to gradually improve the word representation when more and more input has been processed. Our method will be referred to as statistical word discovery (SWD).

SWD is comparable to the word discovery method described in Park & Glass ([7]), in which the primary goal was to address the out-of-vocabulary-problem in speech decoding. The method proposed here and the method described in [7] have two steps in common: first, the similarity between speech fragments is evaluated by a dynamic time-warp (DTW) like algorithm, after which a clustering technique is applied to define symbolic representations. The important difference beween SWD on the one hand and [7] and [7b] on the other is that it does not rely on the availability of a phonetic recogniser to transcribe speech fragments in terms of phone sequences.

The method has remote links with older research carried out in the nineties by Bacchiani, Ostendorf and others, in which the focus was to automatically improve the transcription of words in the lexicon [1]. The difference between that research and the current research is the bootstrapping: here, we do not assume the availability of any word or subword model.

The remainder of this paper is organized as follows. In section 2 the SWD method is explained. In section 3, we present experimental details and results. A final discussion is presented in section 4.

2. The SWD method

The SWD method consists of three stages (details are specified below in the subsections). Each stage has a more symbolic character than the previous stage. Stage 1 comprises a feature extraction followed by a data-driven segmentation of the speech

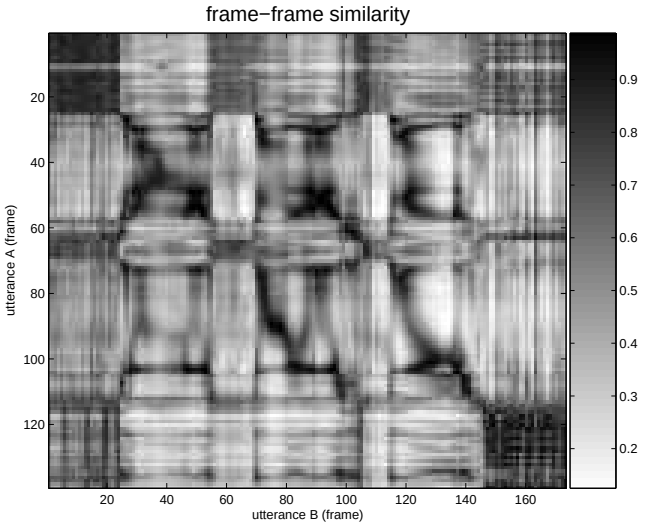

Figure 1: Frame-frame similarity matrix between two utterances. The left-top and right-bottom corner correspond to the starts and ends, respectively, of the utterances. Black cells indicate high similarity (i.e. a small distance).

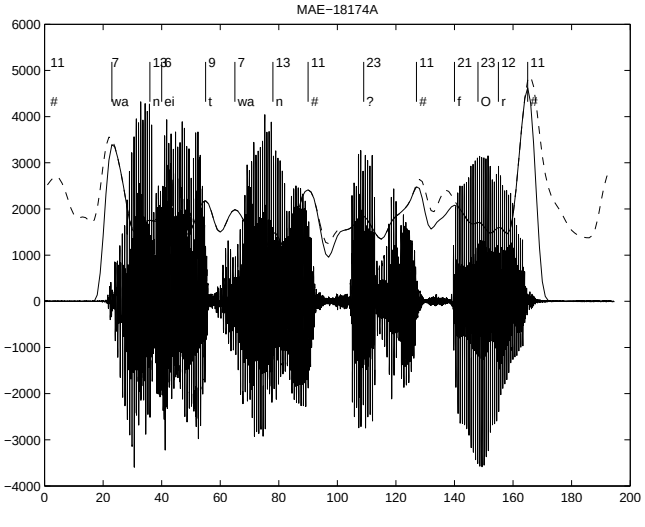

Figure 2: Result of automatic segmentation and labelling. The utterance is 'one eight one seven four'. The solid line presents the function at the left-hand side in Eq. 2. The vertical bars are located at the peaks of this function and indicate the hypothesized boundaries. For the labels see the text.

signal. Its output is a vectorial representation of the utterance in combination with hypothesized speech segment boundaries. The second stage reads in these segments and performs a k-means clustering. The result is an abstract label for each segment (inherited from the cluster it belongs to). In the third stage, the utterances and hypothesized labels are input for the statistical word discovery algorithm. It is in this third stage where the parallel abstract representation is used.

2.1. Stage 1: Automatic segmentation

Recently, several approaches for data-driven segmentation have been proposed (e.g. [14], [15]). The method that is adopted here is an extension of the one used in [15]. First, feature vectors are calculated (12 MFCCs, log energy, delta and delta-delta features). Each output frame is based on an analysis window of 0.032 sec (so e.g. 256 points for 8 kHz files) with 25 percent overlap between consecutive analysis windows. The distance between two frames v_1 and v_2 is defined by (^{t} indicating transpose):

$$d(v_1, v_2) = \arccos(v_1^t v_2 / (v_1^t v_1 v_2^t v_2)^{1/2}) \qquad (1)$$

As an example, figure 1 presents a plot of the frame-to-frame similarity matrix obtained for two files from the clean part of the Aurora 2.0 database ([5]). A high similarity corresponds to a small distance and vice versa. One utterance, displayed downward along the vertical axis, contains the word sequence 'one three' surrounded by silences, the other utterance (along the horizontal axis) 'one seven two', also surrounded by silences. Black (resp. white) cells correspond with high (resp. low) similarity values.

Next, segment boundaries are searched by using a sliding window. A boundary is hypothesized if a distance function that measures the difference between the average feature vectors before the boundary and after the boundary attains a local maximum which is above a certain threshold δ. In the current implementation, we use a window of 2 frames to either side of the boundary. Furthermore, $log(E)$ is used as a weighting factor, modifying the differences measured on the basis of the MFCC feature vectors, to avoid the flagging of many speech segments during the silent (low-noise) portions. The eventual criterion reads

$$\log(E) \cdot d((v_{i-2} + v_{i-1})/2, (v_{i+1}, v_{i+2})/2) > \delta \qquad (2)$$

When compared to human labelling, this text-independent segmentation method yields a boundary accuracy of around 80 percent (using a tolerance of 25 ms to either side of the boundary). On a Dutch database of read speech, about 80 percent of all manual boundaries were correctly detected within 25 ms ([15]). On a subset of TIMIT, a boundary accuracy of 76 percent within 20 ms has been obtained ([16]).
On the 4220 utterances from male speakers in the clean part of the Aurora 2.0 speech database ([5]), stage 1 results in 45356 boundaries (41136 segments).

2.2. Stage 2: Labelling of segments

The labelling of the resulting segments is performed in stage 2. The 41136 segments defined by the hypothesized boundaries were input for a k-means clustering algorithm (Matlab). In this clustering step, the segment-to-segment distance between any two segments S_1 and S_2 is exactly defined by a conventional DTW operating on the Trellis matrix (spanned by S_1 and S_2) in which the local frame-to-frame distances are consistently defined in the same way as in stage 1.
In practice, this clustering was not directly applied on the entire set. To initialise the clustering, a subset of 4100 (first 10 percent) was processed. In subsequent steps, more segments (in steps of 10 percent) were added and the clusters updated until all segments were taken into account. Since different k-means runs often yield different clusters, it was verified that the resulting clustering did not essentially depend on the initial set.
The number of eventual clusters N_c was a user-defined parameter. The value of N_c is to be defined such that the resulting clustering still provides sufficient information to distinguish the

Table 1: Each utterance is associated with abstract information that indicates the presence of a word, but not its acoustical representation or its position in the utterance. This table shows two abstract tags, related to the occurrence of 'two' and 'six'.

Audio file contains	Tag 1	Tag 2
One three two six	yes	yes
Two three five oh four	yes	no
Six six	no	yes
Two six three nine	yes	yes
Oh three three	no	no

relevant speech fragments. If N_c is too small, too few clusters remain and the eventual labelling is coarse. The value of N_c, which broadly approximates the number of 'phones' in the speech data, was optimized so as to capture sufficiently many details in the speech signal. $N_c = 25$ appeared to be an adequate value (understandably, this value is a little bit higher then the number of identifiable phonemes in this database, which is around 20). After clustering, each group was labelled by assigning a unique positive integer to each cluster, and each segment in the group then inherits the label of the cluster it belongs to. In figure 2, the result of the k-means algorithm is shown for one specific utterance. The wavefile ('one eight one seven four') is shown at the bottom. The vertical bars at the top indicate the automatically placed segment boundaries. The numbers at the top of each bar indicates the index of the resulting cluster to which the corresponding segments belongs. The transcription at the bottom of each vertical bar is a broad phone-like transcription of the found segments. In general, the segments are phonetically interpretable. However, in this case segment 23 was assigned to the speech fragment 'seven' and could not be assigned to a particular phone realisation.

2.3. Stage 3: Word discovery by DTW

Stage 3 is the word discovery stage. It takes as input the wave files, in combination with the sequence of labels from stage 2 and the abstract tags (see for an example table 1). Also this stage applies a DTW in which the likelihood of two utterances sharing a common word is estimated using a DTW on the two label sequences. This method is closely related to recent DTW-based matching techniques such as elastic partial matching ([17]).

The word discovery algorithm runs as follows. The input (audio plus tags) is assumed to be available in a list.

1 Select the next utterance.
   - Initialise two empty sets: B_{match} and $B_{no-match}$.
   - Compare the new utterance with all previously observed utterances by performing a DTW on the corresponding label sequences.
   - Find the best-matching subsequence on the best path found by DTW.
   - If both utterances share the same abstract tag (as in table 1), then put the best-matching subsequence into B_{match}, otherwise put it into $B_{no-match}$.
2 Order all items in these sets according to their occurrence.
3 Select the N-best in B_{match} that do not occur in $B_{no-match}$, and monitor this shortlist during the discovery algorithm.
4 Go back to the first step.

Table 2: Example of input for stage 3. For an explanation see the text.

audio file	segment labels	abstract information
1	1 2 5 6 3 6 23 23 1	yes
2	1 5 7 8 23 23 23 1	no
new	1 4 2 5 6 3 23 6 1	yes

Table 3: Example of sorted shortlist in B_{match} after processing 47 (left column) and 106 (right column) files in a particular word discovery experiment. On each line, the first integer represents the absolute count of the label sequence. After that, the integer representation of the label sequence follows. The bracketed integer indicates whether this sequence also occurs in $B_{no-match}$ [1] or not [0]. This list shows the emergence of the sequence (3, 6, 7, 9) among its competitor sequences during a training.

```
--after utt 47--        --after utt 106--
121 3 [1]               751 3 6 7 9 [0]
121 3 6 7 9 [0]         503 6 7 9 [0]
116 6 7 9 [0]           451 6 [1]
112 2 [1]               435 3 [1]
83 6 [1]                398 3 6 7 [1]
66 7 [1]                384 7 [1]
65 9 [1]                382 2 [1]
...                     ...
-----------------------------------
```

In table 2 an example is provided. The first column indicates the index of the utterance; the ones with index 1 and 2 have already been processed. The utterance indicated on the bottom line is the current input. The label sequences and tags are presented in column 2 and 3 respectively. A DTW between the new utterance and utterance 1 will select (5, 6, 23) as cheapest subpath. This solution will therefore be put into B_{match}. In contrast, the DTW between the new utterance and utterance 2 does not provide a clear low-cost subsequence and might for example select the short subsequence (23). Here, the tags do not match so this result is collected into $B_{no-match}$. After a few utterances have been processed, the B_{match} set will contain longer sequences of which a few will tend to stand out among the competitors. The $B_{no-match}$ mostly contains sequences that are shorter than those in B_{match}. When more utterances are processed, the sorted B_{match} lists show words emerging from the sets which were seemingly random in the beginning. $B_{no-match}$ is used as a set of negative examples to elimitate those solutions in B_{match} that also occurred between non-matching utterances. Table 3 shows an example of the 'emergence' of a particular label sequence.

3. Data and results

The database Aurora 2.0 has been used for selecting utterances for testing and comparison. The motivation for this digit-string database is twofold: (1) it provides low-perplexity speech data,

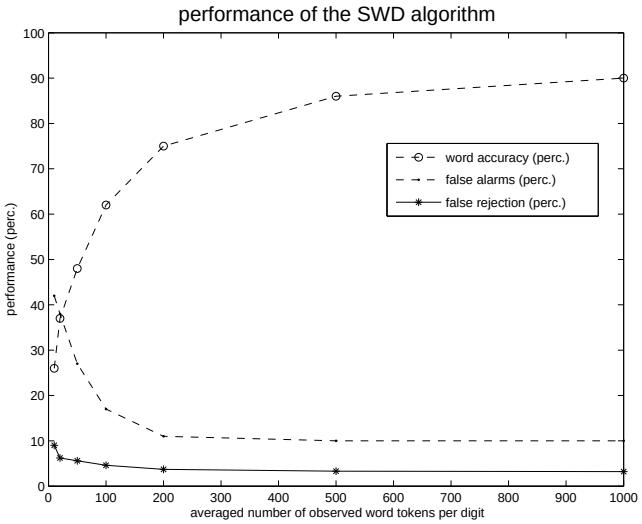

Figure 3: Performance of the SWD algorithm. The x-axis shows the average number of tokens of each digit in the comparison set. The y-axis represents the performance in terms of *percentage* for all plots.

which will simplify to give a proof of concept of the word discovery algorithm and (2) it contains multiple copies with different SNRs which makes it possible to study word discovery procedures in noisy conditions in subsequent studies. In the present study we focus on the 4220 utterances spoken by male speakers in the clean part of Aurora 2.0. These utterances contain on average 3.2 digit per utterance (min 1, max 7).

The performance of the algorithm is shown in terms of three criteria: (a) the accuracy to distinguish the eleven word types ('zero' to 'nine' plus 'oh', silence is discarded) (b) the number of false alarms (cases where SWD locates a word onset where there is none) and (c) false rejection (cases where SWD misses a word). In figure 3 the performance of SWD is presented in the case where SWD has observed $T = 10$, 20, 50, 100, 200, 500, and 1000 tokens of each word. At $T = 1000$, word accuracy is 90 percent, with room for improvement for larger T. Clearly, the number of false alarms and the number of false rejections decrease with increasing size of the contrast set. We note a relatively high amount of false alarms. This may indicate that the word representations that are built up tend to be a bit shorter than they should be, which is in turn related to the precise definition of the optimality of a symbolic subsequence as applied in Stage 3.

4. Discussion

The SWD algorithm is able to bootstrap from the speech signal itself without using any predefined lexical knowledge or phone models. The learning curves of SWD show improvements in terms of accuracy, false alarm rate, false rejection as a function of the number of words available in the set of utterances. In the word detection stage, the abstract tag information is essential to contrast hypotheses found in the matching condition (i.e. B_{match}) with the non-matching condition.

The SWD algorithm consists of a cascade of intertwined stages. The same distance is used in stage 2 and stage 3, and the same DTW principle is used to define segment-to-segment distances and to hypothesize the symbolic representations of shared word-like speech fragments. The choice of the number of clusters in the k-means step deserves some care; the optimal choice is likely to be related to the number of phones that can be identified in the speech material.

A conceptual point of considerable interest is whether the acquistion of phones preceeds the acquisition of words. Here phone-like units are hypothesized in a data-driven way, on top of which words are hypothesized in an hierarchical manner. Also the use of additional, paralinguistic information is of considerable interest. Prosody (pauses, energy and pitch contours) may facilitate the detection of potential words. The multiple use of knowledge sources, as well as the use of this algorithm in noisy conditions will be studied in the near future.

5. Acknowledgements

The first author participates in the EU FP6 FET project Acquisition of Communication and Recognition Skills (ACORNS) (Dec. 2006-2009).

6. References

[1] Bacchiani, M. (1999). Speech recognition system design based on automatically derived units. PhD Thesis, Boston University (Dept. of Electrical and Computer Engineering) (available online).

[4] Goldinger, S.D. (1998). Echoes of echoes? An episodic theory of lexical access. Psychological Review, 105, 521–279.

[5] Hirsch, H.G. and Pearce, D. (2000). The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. Proceedings ISCA ASR, 2000, p. 181-188.

[6a] Newport, E.L. (2006). Statistical language learning in human infants and adults. Plenary addressed at Interspeech 2006, Pittsburgh, USA (Sept. 2006).

[7] Park A., and Glass, J.(2006). Unsupervised word acquisition from speech using pattern discovery. Proceedings ICASSP-2006, Toulouse, France, pp. 409–412.

[7b] Roy, D., and Pentland, A. (2002). Learning words from sights and sounds: A computational model. Cognitive Science, 26 (1), 113-146.

[8] Werker, J.F. and Curtis, S. (2005). PRIMIR: a developmental framework for of infant speech processing. Language Learning and Development, 1: 197-234.

[11] Marcus, G.F., Vijayan, S., Bandi Rao, S., and Vishton, P.M. (1999). Rule-learning by 7-month-old infants. Science, 283, January, 77-80.

[12] Prince C.G. and Hollich, G. J. (2005). Synching infants with models: a perceptual-level model of infant synchrony detection. The Journal of Cognitive Systems Research, 6, 205-228.

[13] Saffran, J.R., Aslin, R. N., and Newport, E. L. (1996). Statistical learning in 8-month-old infants, Science, 274, December, 1926-28.

[14] Leelaphattakij, P., Punyabukkana, P., and Suchato, A. (2006). Locating phone boundaries from acoustic discontinuities using a two-staged approach. Proceedings ICSLP06, Pittsburgh, USA.

[15] ten Bosch, L., Baayen, H., and Ernestus, M. (2006). On speech variation and word type differentiation by articulatory feature representations. Proceedings ICSLP06, Pittsburgh, USA.

[16] Aversano, G., Esposito, A., Esposito, A., and Marinaro, M. (2001). A new text-independent nethod for phoneme segmentation. Proc. of the 44th IEEE Midwest Symposium on Circuits and Systems. Vol. 2, pp. 516-129.

[17] Latecki, L.J., Megalooikonomou, V., Wang, Q., Lakaemper, R., Ratanamahatana, C. A., and Keogh, E. (2005). Elastic partial matching of time series. In: Proc. Principles and Practice of Knowledge Discovery in Databases (PKDD), Porto, Portugal.

[18] Moore, R.K., and Maier, V. (2007). Preserving fine phonetic detail using Episodic Memory: Automatic Speech Recognition with Minerva2. Proceedings ICPhS 2007, Saarbrücken, Germany.