A Paradigm for Limited Vocabulary Speech Recognition Based on Redundant Spectro-Temporal Feature Sets

Sourish Chaudhuri, Bhiksha Raj

Language Technologies Institute
Carnegie Mellon University, Pittsburgh, PA-15213
sourishc, bhiksha@cs.cmu.edu

Tony Ezzat

Massachusetts Institute of Technology
43 Vassar Street, Cambridge, MA-02139
tonebone@mit.edu

ABSTRACT

Speech recognition techniques have come to rely almost completely on HMM based frameworks. In this paper, we present a novel paradigm for small-vocabulary speech recognition based on a recently proposed word spotting technique. Recent work using discriminative classifiers with ordered spectro-temporal features to detect the presence of keywords obtained encouraging improvements over HMM-based models. We propose to extend this approach to recognize continuous speech in our work. Our method uses discriminative models to predict which words are present in a speech signal and hypothesize their locations. A graph search using dynamic programming is then used to obtain the most likely sequence of words from the hypothesis set produced as a result of combining the results from the discriminative word classifiers. While this approach doesn't perform as well as state-of-the-art ASR systems, it can be particularly useful for languages with small amounts of annotated data available.

Index Terms— word-spotting, spectro-temporal features, speech recognition

1. INTRODUCTION

Hidden Markov Models have been the dominant statistical model employed for speech recognition for several decades, since they provide a simple and efficient framework for modeling speech signals, under the constraints of time and computational complexity [3,7]. It is generally recognized that it is unable to effectively model the *temporal patterns* of spectral features in speech. The assumption of independence implies that any permutation of a sequence of feature vectors is equally likely, provided they have all been generated from the same state. This is illustrated in Figure 1.

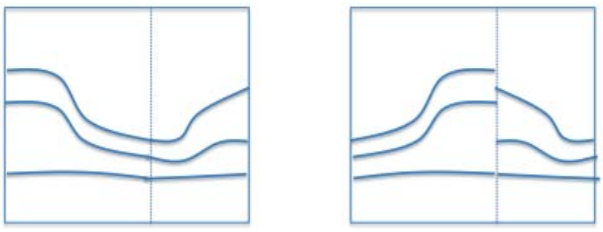

Fig. 1. Given $S_1, S_1, \ldots, S_1, S_2, \ldots, S_2$, a state sequence with the transition from S_1 and S_2 at the dotted line, both patterns above are assigned equal likelihood by an HMM.

In order to compensate for this lack of an explicit model of temporal continuity, the features computed from the speech signal are sometimes extended to include difference- and double-difference terms (representing velocity and acceleration of spectral patterns); nevertheless, this doesn't truly capture the continuity in the underlying patterns. Alternate modeling paradigms have been proposed, such as stochastic segmental models, and segmental HMMs [4], which attempt to model the temporal patterns more explicitly; however these models go to the other extreme of modeling the *entire* spectro-temporal pattern as a single unit, while remaining essentially state-based, like the HMM.

In this work, we introduce a new modeling paradigm for limited vocabulary continuous speech recognition that employs redundant sets of spectro-temporal patterns as the basic unit of representation. This paradigm is not state-based. Instead we perform recognition by searching for a large number of localized spectro-temporal patterns, which together bear evidence to the identity of the word. Our results show that this method performs especially well when working with limited training data, making it especially useful for low resource languages or applications.

Researchers have previously attempted to employ localized spectro-temporal patterns for recognition. However, these features, by their nature, span several analysis frames in an asynchronous manner and do not end themselves to efficient joint search for word boundaries and word hypotheses. As a result, prior research has mostly been restricted to simple problems. In [2,6] parts-based classifiers for phonetic units is described, but actual recognition of connected speech is not attempted. Localized patterns have also been used to rescore N-best recognition hypotheses, typically within a conditional-random field framework, but the hypotheses must be generated by an HMM-based system. Other similar attempts have similar restrictions.

In this paper we build upon the patch-based word spotting technique proposed by Ezzat and Poggio [5] (which reported considerable improvements over the prevalent HMM-MFCC based word-spotting techniques) to perform continuous speech recognition. The word spotter is employed to hypothesize entire word graphs that can be searched to recognize what was spoken. Preliminary results indicate on a small-vocabulary task, given give only a very small amount of training data, the proposed method outperforms an HMM-based system trained on the same data. While we have not yet achieved similar results with larger training data sizes, we believe the technique holds promise for larger data sets as well.

28 – 31 August 2011, Florence, Italy

2. WORD SPOTTING WITH REDUNDANT SPECTRO TEMPORAL PATTERNS

Our speech recognition paradigm builds on Ezzat and Poggio's word-spotting framework [5]. Word-spotting systems have typically relied on an HMM framework employing with Mel-frequency cepstral coefficients on frames [1,4] as features. In contrast, the Ezzat-Poggio technique uses a discriminative support-vector machine (SVM) classifier based on *redundant* features, that characterize the presence of *localized* spectro-temporal patterns in the signal.

For each target keyword, patches of random height and width are extracted from the spectrogram of an example utterance of the word as illustrated in Figure 2. The patches and their location in frequency and time (relative to the duration of the word) are stored in a dictionary.

The training data for the keyword spotter consists of positive and negative examples of the target word. Positive examples are utterances of the word. Negative examples are randomly drawn speech segments of the same average length as positive examples, that don't contain the word. For any training example with a spectrogram $S(f,t)$, the feature corresponding to the k^{th} dictionary patch P_k, of size (F_k, T_k) and centered at (f_k, t_k) is obtained as the largest correlation between P_k and any (F_k, T_k) sized region of $S(f,t)$ located in a small window around (f_k, t_k).

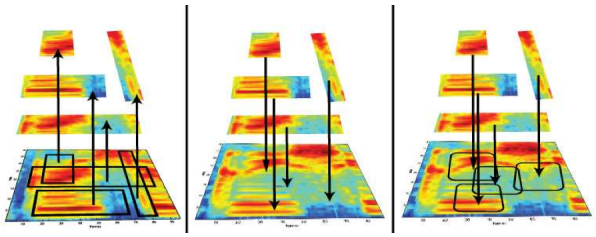

Fig. 2. Randomly drawn patches from a template (left) are correlated with other instances (center and right) to generate features.

Given a dictionary of K patches, we obtain a K-dimensional feature vector for each instance. Once features are computed for all positive and negative exemplars, an SVM is trained to discriminate between them. A further refinement is carried out, where the system tests the newly learnt classifier on additional negative data. False positives from this set are then used to retrain the SVM.

When evaluating a test utterance for the presence of a keyword, windows of width W equal to the average duration of the target keyword are evaluated by the classifier. This window is slid across the entire length of the utterance using a hopsize of $0.2 \times W$.

As reported in [5], this system consistently outperformed an HMM-MFCC baseline across varying amounts of training data used. Additionally, the amount of training data required for it was generally less than that required by HMM-MFCC frameworks. Our implementation replicated the original system- interested readers are directed to the original paper for additional implementational details.

3. SPEECH RECOGNITION BY WORD SPOTTING

The word spotting technique described above actually computes the *margin* of the SVM classifier at each time. Words are spotted by detecting the instants at which the margin crosses a threshold.

This is easily harnessed for limited-vocabulary continuous speech recognition as follows. A separate SVM word spotter is trained for each word in the vocabulary. On the test data, for each word, we use the sliding-window method described above to compute the margin of the SVM for each word as a function of time. We employ a word-specific threshold to determine candidate locations for the word. For each instant at which we have a margin that lies above the threshold, we hypothesize an instance of the word, with a duration equal to the detected duration at that margin, and with a score equal to the margin. This is illustrated in Figure 3. Once we

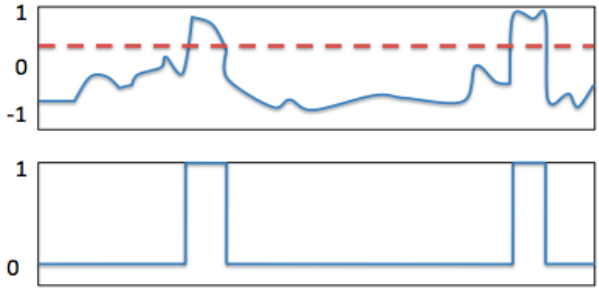

Fig. 3. Hypothesizing locations of a word.

have hypothesized candidate locations for all words, we compose a directed graph from all of them as follows:

Analysis frames in the spectrogram represent nodes in the graph. Words are represented by edges. If a word is detected between the i^{th} and j^{th} frames with margin $w(i,j)$, an edge with score $w(i,j)$ is introduced between nodes i and j to represent the word. The direction of the edges are naturally left to right in time. If no word has been hypothesized between two frames, an epsilon edge is added between them with a negative edge weight proportional to $|i-j|$. The objective here is to penalize the edge since it is likely a unit too small to be a word and is actually spanned by a word. We assume, in general, that silence in continuous speech is unlikely and the negative weights reflect that assumption. It is expected that the frame span will actually be a sub-span of another edge that actually carries a word, and the negative weight will not be preferred in that case.

To recognize the spoken word sequence, we find the best path through the resulting graph using a Viterbi decoding scheme. We define the *Score* at any frame i as the score of the best path that starts at the beginning of the utterance and ends at frame i. The *Score* at a frame is computed as follows :

$$Score[i] = \arg\max_{j<i} Score[j] + w(j,i) \quad (1)$$

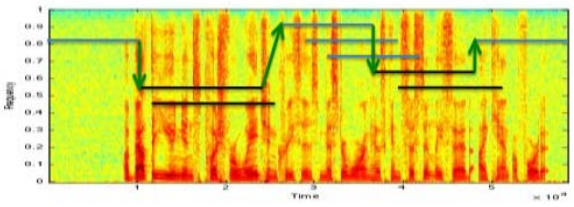

Fig. 4. Speech recognition by word-spotting

Figure 4 illustrates the procedure with a hypothetical example. Here, a vocabulary of two words, *blue* and *black* is used. The blue

Table 1. Comparison of sentence- and word-level accuracies.

System	Word Acc.	Sentence Acc.
Sphinx-R	71.73%	56.06%
Word-spotter based	74.33%	59.61%

and black lines show the hypothesized locations of the words. Notice that the same word has often been hypothesized at multiple points very close to each other. This is because the duration used for the sliding window is an average from the training data, and the real occurrence is often slightly longer or shorter, leading to multiple hypotheses corresponding to a single occurrence of the word. The Viterbi decoder will usually pick only one of these occurrences (but not always). The path shown in green across the length of the utterance is the best scoring path found by the Viterbi decoder through the utterance graph.

4. EXPERIMENTS

Preliminary experiments were conducted on the TIDIGITS database to evaluate the performance of the proposed framework. The vocabulary comprises eleven words, including the numbers "one" through "nine" and two variants of 0, "zero" and "oh".

We are aware that the regular speech recognizer trained on a large dataset can reach WER of 0.6%. Our system currently cannot reach that level of performance. However, the key advantage of the discriminative word spotting system-based speech recognizer is that it can perform well when the amount of available training data is small, making it useful for applications with low-resource languages or applications with limited training data. In this paper, our experiments simulated this condition.

The training data consists of utterances from the database such that the number of positive and negative exemplars for each word is 30 each. From this training set, we constructed our patch dictionary for each word, using the procedure mentioned in Sec 2. The training data is taken from the *clean1* section of the TIDIGITS dataset, while the test set consists of the *clean2* and *clean4* sections from TIDIGITS. Each of these data sets consists of just over a 1000 utterances. Note, however, that in order to train the word-spotting based system, the amount of data used per word is just the word itself from about 30 utterances, and negative data (i.e. randomly extracted patches from data that did not contain the word). The total amount of data directly used to train our system, therefore, is effectively only about 1 minute of data per word, or a total of about 11 minutes.

We compare our results against a baseline system that uses the CMU Sphinx speech recognition system. We train the baseline system in 2 ways- the first, which we refer to as as *Sphinx-Full*, uses the entire TIDIGITS training data to train the Sphinx system. The second uses the same restricted set of examples that we used to train the word-spotting system. We refer to this system as *Sphinx-Restricted*.

4.1. Results

The Sphinx speech recognition system is highly optimized for recognition and as expected, has a word error rate (WER) of only about 0.6% on the test data. The word- and sentence-level performance of the *restricted* Sphinx system (Sphinx-R) is compared to our word-spotting based system over the 2,003 utterances in Table 1.

As explained in Sec. 3, the search imposes a penalty for frames which were not spanned by any words. This penalty is proportional

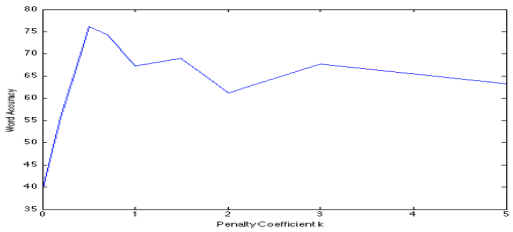

Fig. 5. Variation of word level recognition accuracy with the proportionality constant for the penalty

to the number of frames not spanned, with a proportionality constant k. This constant can be optimized. We used a held out set of a 100 utterances from the training set over which we tested for different values of the proportionality constant k. The variation of word level recognition accuracy with the value of k is shown in figure 5. The optimum value for the penalty coefficient, found using the held out data, was found to be 0.5 for our experiment.

4.2. Analysis of Results

We note that with small amounts of training data, the recognition accuracy obtained is significantly superior to that obtained with an HMM-based system trained on the same amount of data. More importantly, besides the reduced amount of training data, the HMM-based system is well tuned. Various penalties such as insertion penalties, silence handling etc. have been optimized. On the other hand, the word-spotter-based system is very preliminary. Thus the improved performance is very encouraging.

The confusion matrix for the recognizer is shown in Table 2. It not only identifies recognition errors, but also indicates insertions and deletions of words. It is largely unsurprising, and confusions are mostly acoustically plausible. For instance, the words *four* and *five* are misclassified as each other most often, possibly due to the similarity of patches from the initial parts of the two words. However in other cases the confusions are odd. *E.g.*, it is not immediately clear to us why *two* is most often incorrectly classified as *eight* – we further notice that the word *eight* is hardly ever misclassified as *two*. Anecdotally, however, we are aware that even some commercial speech recognition systems of only a few years ago would sometimes recognize the word sequence "two two two two two" as "six eight six eight six eight" for some subjects.

We also identified other systematic errors committed by the patch-based paradigm. The words *four* and *seven* were most often predicted when they were in fact absent. The individual words that are least often correctly predicted are *two*, *seven* and *one*, in that order. *Two* is also the word that is missed most often. This may be because the duration of the word varies from average most frequently. In general, however, the pattern of the errors does not appear to result from the form of the recognizer and is not different from that obtained with an HMM-based system.

A possible issue with the proposed formulation is that the sliding window used when detecting words has a width equal to the average width of the training examples of the words. Based on our observations from the results, it appears from word-spotting experiments on a small sample of data that when the ratio of the expected and actual durations are more than 2, the performance of the word-spotter drops well below its average performance.

Another issue is how silence and pauses must be modelled. Cur-

Table 2. Confusion matrix for each of the words. The words in each row are the target words, and each column are the candidates. Thus column *j* for row *i* says what percentage of word occurrences in row*i* were detected as the word in column *j*. All the entries are in %.

	one	two	three	four	five	six	seven	eight	nine	zero	oh	(deletion)
one	73.2	2.2	0	0	0	0	11.4	2.1	5.1	0	3.1	3.0
two	1.4	61.1	1.0	0.5	0	1.9	0	21.2	0	0	0	11.3
three	1.1	0.1	92.0	0	0	0	2.5	0	0	1.2	0	3.1
four	0	0.5	2.5	87.2	4.9	0.3	0	1.9	1.5	0	1.4	0
five	0	0	0	8.2	86.7	0	0	5.1	0	0	0	0
six	0	0	7.1	0	0	86.2	3.1	0	0	0.5	0	3.0
seven	7.1	0	1.4	5.1	0	2.0	71.0	4.1	0.8	0	0.3	8.3
eight	3.1	0.1	0	0	0	1.5	8.1	81.3	0	0	0	5.8
nine	4.9	2.7	0	0	0	0	5.1	3.1	76.6	0	1.9	5.8
zero	0.2	2.3	0	6.5	0	1.1	7.2	0	1.2	77.8	1.1	3.6
oh	2.1	0	0	11.2	0	0	2.1	0	0	6.1	77.5	1.1
(insertion)	0	0	0	0.4	0	0	0.6	0	0	0	0	0

rently we do not model them. Consequently, the process of constructing a lattice does not distinguish between a valid pause and a deleted word. This can result in an otherwise well-scoring hypothesis to be ranked poorly. Often, correct words were actually present in the lattices, but not included in the final hypothesis. Our analysis showed that about 81% of the utterances had the correct sequence of words present in the lattice but the Viterbi decoding did not find the correct one in over a quarter of these cases, implying that both lattice composition and decoding must be improved for the technique to be considered competitive.

5. DISCUSSION

The experimental results reported here are the first experiments with this new proposed paradigm for limited vocabulary speech recognition with small training datasets. The results are comparable to, or even better than those achievable with a state-of-art system trained on the same data, when the data are small.

The question remains: will the performance carry over to large data or bigger tasks? We have not yet optimized search heuristics, such as graph formation, various search penalties etc. The handling of noise and pauses must be improved. Speech recognition systems benefit from the use of a language model, since they were not required for the digits task. However, in the general case, word probabilities can be easily incorporated as node or edge weights, depending on the formulation.

In our current system, a word spotter is required for each word in the vocabulary. In general speech recognition, where the number of possible words is potentially infinite, the word spotting could be modified to detect phonemes or syllables instead (this would be particularly effective for syllabic languages like Japanese). We believe all of these to be surmountable problems and continue to actively explore them.

The paradigm provides two principal advantages over standard HMM-based recognizers. First, it needs far less data to train. Second, the use of *redundant* and *localized* spectro-temporal patterns as features make the system more noise robust than ones that use entire spectral frames for training. In non-redundant segmental or frame-based representations of signals, corruption of any single region of a unit (frame or segment) corrupts the entire unit, thereby making a large number of *uncorrupted* time-frequency components effectively unavailable for classification. In contrast, in the redundant localized patch based representation, each time-frequency component is represented in multiple localized patches, and most uncorrupted components are represented through at least a few patches that cover the uncorrupted regions of the spectrogram, but not the corrupted ones.

Our preliminary experiments were run only on clean speech data. The next step would be to investigate the performance of our system on noisy data. We would like to note here that similar features have been used in [2] for various isolated recognition tasks, and they also find the spectro-temporal feature sets beneficial when performing recognition in noisy conditions, as also reported in [6]. Our experiments also show the standard problem of using SVMs for a task like this one- the general formulation of the SVM has trouble handling a massive training corpus, and generative systems have a clear advantage in real-world tasks.

6. REFERENCES

[1] J.R. Rohlicek, W. Russell, S. Roukos, and H. Gish, "Continuous hidden markov modeling for speaker-independent wordspotting", *In Proc. of ICASSP*, pp. 627-630, 1989.

[2] K. Schutte. "Parts-based Models and Local Features for Automatic Speech Recognition", *PhD Thesis, Massachusetts Institute of Technology*, 2009.

[3] M. Gales and S. Young, "The Application of Hidden Markov Models in Speech Recognition", *In Foundations and Trends in Signal Processing*, volume1, No. 3, 195-304, 2007.

[4] M. Ostendorf, V. Digalakis and O.A. Kimball, "From HMMs to Segment Models: A Unified View of Stochastic Modeling for Speech Recognition", *In IEEE Transactions on Speech and Audio Processing*, 1996.

[5] T. Ezzat and T. Poggio, "Discriminative Word-Spotting Using Ordered Spectro-Temporal Patch Features", *In Proc. of SAPA Workshop, Interspeech, pp 35-40*, 2008.

[6] Gy. Kovacs and L. Toth, "Localized spectro-temporal features for noise-robust speech recognition", *Computational Cybernetics and Technical Informatics (ICCC-CONTI), pp.481-485*, May 2010

[7] X. Huang, A. Acero and H. W. Hon, "Spoken Language Processing: A Guide to Theory, Algorithm and System Development", *Prentice Hall*, 2001.