4th European Conference on
Speech Communication and Technology
EUROSPEECH '95
Madrid, Spain, September 18-21, 1995

WORD HYPOTHESIZER BASED ON RELIABLY DETECTED PHONEME SIMILARITY REGIONS

Philippe Morin and Ted H. Applebaum

Speech Technology Laboratory, Panasonic Technologies Inc.
3888 State St., Santa Barbara, CA 93105 USA

Abstract

This paper presents a time and memory efficient multistage word candidate hypothesizer suitable for medium-size vocabulary applications on small hardware. It is based on a novel compact speech representation: regions of high phoneme similarity values. The processing stages of the word hypothesizer are applied in sequence to reduce the search space for a more computationally expensive fine match word recognition system. The paper also presents a scoring procedure for combining information from each stage of the hypothesizer with the output of the fine match procedure to produce the final word decision. On a 100 word task, use of the word hypothesizer reduced alignment complexity by 93% (compared to exhaustive search by the fine match alone), with significant error rate reduction for clean and noisy test data due to score combination.

Concept

Speech representation by phoneme similarities has been applied in speaker-independent template-based word recognition systems [1-3] for their relative insensitivity to speaker variations. Phoneme similarity values are typically computed as the normalized Mahalanobis distance between a segment consisting of consecutive LPC analysis frames and a standard phoneme template. As shown in Figure 1, there is an overall consistency in the shape of the phoneme similarity time series for a given word. Similar behavior is observed in the phoneme plausibility time series of the VINICS system [4].

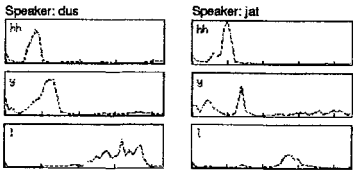

Figure 1: *Phoneme similarity time series for the word "Hill" spoken by two speakers*

Speech recognition systems which match each input utterance to reference templates composed of phoneme similarity vectors, as in the model speech method (MSM) of Hoshimi et al [1-3], have achieved high accuracies for small vocabulary tasks. Their reference speech representation is frame-based and requires a high data rate (typically 8 to 12 parameters every 10 to 20 msec). The frame-by-frame alignment that is required is computationally costly and makes this approach unsuitable for larger vocabularies, especially on small hardware.

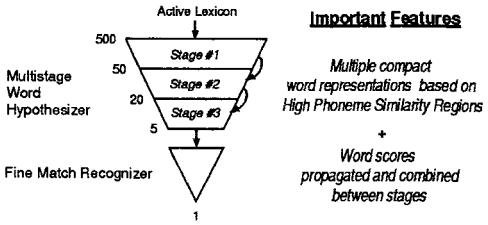

Figure 2: *Integration of multiple stages of word hypothesization with a fine match procedure*

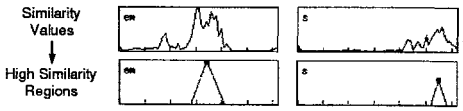

Figure 3: *Output of region picking procedure*

This paper presents a multistage word hypothesizer that is applied prior to the frame-by-frame alignment in order to reduce the search space and achieve real-time performance. As suggested in Figure 2, the number of stages in the hypothesizer, as well as the computational complexity of each stage and the number of word candidates preserved at each stage, can be adjusted to achieve desired goals of speed, memory size and recognition accuracy for a particular application. The word hypothesizer and fine match procedure share the initial representation of speech as a sequence of multiple phoneme similarity values. However the word hypothesizer further refines this speech representation to preserve only the interesting *regions of high phoneme similarity* (cf. Figure 3). By representing the speech as features at a lower data rate in the initial stages of recognition, the complexity of the matching procedure is greatly reduced.

The paper also presents a scoring procedure for propagating and combining the scores obtained at each stage of the word hypothesizer with the scores of the fine match procedure in order to produce a final word decision. By combining the quasi-independent sources of information produced by each step, a significant gain in accuracy is obtained.

System Overview

The system's architecture features three distinct components that are applied in sequence on the incoming speech to compute the best word candidate.

The first component is a phoneme similarity front-end that converts speech signals into phoneme similarity time series. Speech is digitized at 8 KHz, and processed by 8th order LPC analysis to produce 8 cepstral coefficients

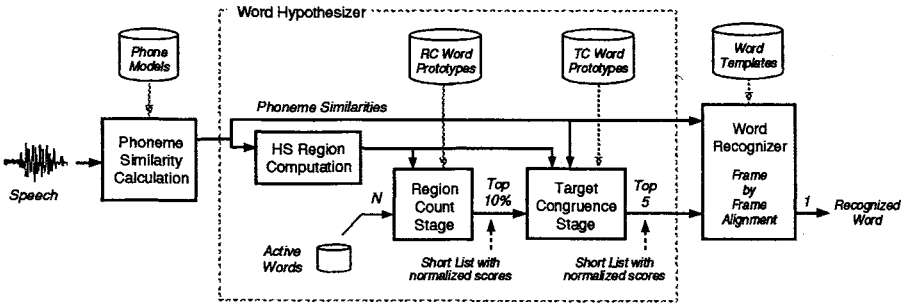

Figure 4: *Recognition System Overview*

per centisecond. Each block of ten successive frames of cepstral coefficients is compared to 55 phoneme reference templates (a subset of the TIMIT phoneme units) to compute a vector of multiple phoneme similarity values. The block of analysis frames is then shifted by one frame at a time to produce a vector of phoneme similarity values each centisecond.

The second component is the word hypothesizer itself. A peak-driven procedure is first applied on the phoneme similarity time series to extract High Similarity regions (HS regions). In the process, low peaks and local peaks of phoneme similarity values are discarded (see Figure 3). Regions are characterized by four parameters: phoneme symbol, height at the peak location and time locations of the left and right frames. Over our data corpus, an average of 60 regions per second of speech is observed. Then, two stages, *Region Count* and *Target Congruence* (described below) which use two different compact word representations, are applied to provide a short list of word candidates with their recognition scores.

Finally, the third component (English MSM [3]) performs fine match recognition on the short list and computes the best word candidate.

Region Count Modeling

The first stage of the word hypothesizer represents each reference word with statistical information on the number of HS regions over a pre-defined number of time intervals. Currently words are divided into three equal time intervals and each phoneme interval is described by 1) the mean of the number of HS regions occurring in that interval and 2) a weight inversely proportional to the variance which indicates how reliable the region count is. These parameters are easily estimated from training data. Each word requires exactly 330 parameters which correspond to 2 statistics × 3 intervals × 55 phoneme units.

Region count modeling was found very effective due to 1) its fast alignment time (0.33 millisecond per test word on a Sparc10 workstation), and 2) its high top 10% accuracy (see Figure 6).

Target Congruence Modeling

The second stage of the word hypothesizer represents each reference word 1) by a prototype which consists of a series of phoneme targets and 2) by global statistics, namely the average word duration and the average "match rate", which represents the degree of fit of the word prototype to its training data. *Targets* are generalized HS regions and require five parameters:

- Phoneme symbol
- Target weight (% occurrence in training data)
- Average peak height (phoneme similarity value)
- Average left and right frame locations

Word prototypes are automatically created from training data as follows. First HS regions are extracted from the phoneme similarity time series from a number of training talkers. Then for each training utterance of a word, reliable HS regions are computed by aligning the given training utterance with all other utterances of the word in the training data (region-to-region alignment). For each training utterance, the number of occurrences (or probability) of a particular region is then obtained. At that time, regions with probabilities less than *Reliability_Threshold* (typically 0.25) are found unreliable and are eliminated. The word prototype is constructed by merging reliably detected high similarity regions to form targets. At the end of that process, a *target rate* constraint (i.e. desired number of targets per second) is then applied to obtain a uniform word description level for all the words in the lexicon. The target rate also allows a reduction of the number of targets by keeping the most reliable ones. Once the word prototype has been obtained, the average match rate and average word duration are computed and stored.

The number of parameters needed to represent a word depends on the average duration of the word and on the level of phonetic detail that is desired. For a typical 500 millisecond word at 50 targets per second, the speech representation requires 127 parameters which corresponds to 5 values per target × 50 targets per second × 0.5 seconds + 2 global statistics.

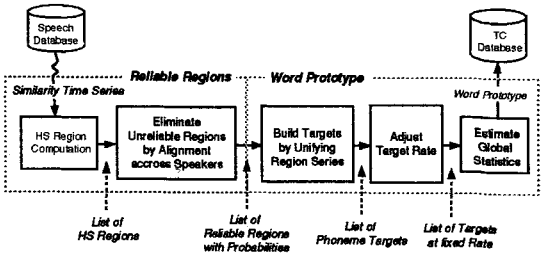

Figure 5: *Target Congruence Word Prototype Training Procedure*

Word Hypothesization

Given an active lexicon of N words, the region count stage is first applied to produce a short list of word can-

didates with normalized scores. A weighted Euclidian distance is used to measure the degree of fit of a test word X to a reference word P (RC format). It is defined as $D_{RC}(X,P) = \sum_{i=1}^{3}\sum_{j=1}^{55}(x_{ij}-p_{ij})^2 w_{ij} / \sum_{i=1}^{3}\sum_{j=1}^{55} w_{ij}$ where x_{ij} is the number of HS regions in time interval i for phoneme j, where p_{ij} is the corresponding average number of HS regions estimated on training data, and where w_{ij} is the corresponding weight. The N/10 highest scoring word prototypes are preserved as word candidates, and their scores are normalized by dividing each individual score by the highest score. This defines a normalized score S_{RC} for each reference word. Normalized scores range from 0 to 1 and are dimensionless, making it possible to combine scores resulting from different scoring methods.

The target congruence stage is then applied on each word candidate selected by the first stage. A region-to-target alignment procedure is used to produce a congruence score between the test word and a given word reference (TC format). The congruence score of a matched target CG_{match} (i.e. alignment found between target t of the prototype and region r of the test word) is defined as: $CG_{match}(t,r) = min(A_t/A_r, A_r/A_t)$ where A_t and A_r respectively represent the target's area and the aligned region's area in the time-similarity plane. The congruence score of an unmatched target $CG_{unmatch}$ is computed in the same way, using an estimate for the area A_r of the missing HS region. The estimated A_r is computed as the area under the similarity curve for the target's phoneme label, between the projected locations of the target's left and right frames. The word congruence score is computed as the weighted sum of congruence scores for all the targets, divided by the sum of their weights. Normalized congruence scores S_{TC} are computed by dividing the individual congruence scores by the highest congruence score. The final score output by the word hypothesizer is $S_{Hypo} = (S_{RC} + S_{TC})/2$. The five words having the highest combined scores are selected as word candidates for the fine match.

Word Recognition

Fine match recognition is then performed. Unlike the word hypothesizer, the word recognizer uses the phoneme similarity time series directly in a frame-by-frame dynamic programming match on the list of five word candidates given by the hypothesizer. Fine match recognition scores are normalized (S_{FM}) and combined with the scores of the hypothesizer. The global score of each word in the short list is then defined as: $S_{Global} = (S_{Hypo} + S_{FM})/2$

Evaluation Task

Recognition word accuracy was evaluated in isolated word speaker-independent mode on a speech database of 100 English proper names. Testing was performed in several noise conditions: clean test speech and speech with additive noise at 20 dB or 10 dB Signal-to-Noise Ratio. Two kinds of non-stationary additive noise were used in testing: *Car noise*, which was recorded in a moving Toyota Crown automobile; and *Datashow noise*, which was recorded in a large exhibition hall and contains multi-talker babble and music.

Phoneme models were trained on the TIMIT database SX sentences, downsampled to 8 kHz sampling rate. For training nominal clean phoneme models, each sentence was used twice: once as clean speech and once with artificially added stationary pink Gaussian noise at 20 dB SNR. (This combination was found to improve recognition results, even for clean test conditions). For training multi-style phoneme models, the additive noise was replaced by Datashow noise at 10 dB SNR.

Word level training and testing was done on one repetition of speech data from sixty-four talkers. Word prototypes were trained and tested on non-overlapping gender-balanced sets of 32 talkers each. Under the clean training condition, word prototypes were built using the nominal clean phoneme models and one training pass over the noise-free training speech data. Word prototypes, for the multi-style training condition, used multi-style phoneme models and two training passes over the speech data: once clean and once with 10 dB SNR additive Datashow noise. Each recognition data point resulted from 3200 trials.

Word Accuracy Results

Clean Speech Condition

Recognition rates are shown in Figure 6 for the different stages and combinations in the system. The output of the hypothesizer (list of Top 5 word candidates) shows no critical deterioration (99.6% accuracy) even when compared to original fine match alone (99.3% accuracy for top 5 candidates). Due to the independence of the errors made by the RC and TC stages, the word hypothesizer, which combines the scores from its two stages, achieves better top 1 recognition than any stage alone. The best top 1 recognition rate (96.5%) is achieved by the whole recognition system, where the fine match is run on the top 5 word candidates from the hypothesizer, and the final word decision is made by combining the normalized scores from the hypothesizer and the fine match.

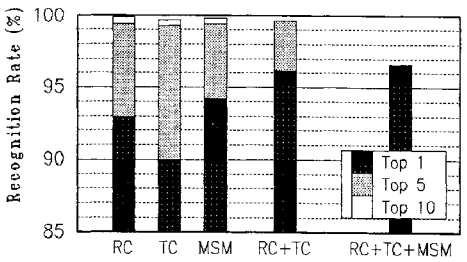

Figure 6: *Recognition results for clean test speech for the two word hypothesizer stages (RC, TC), the fine match (MSM), the word hypothesizer (RC+TC) and the recognition system as a whole (RC+TC+MSM). Word prototypes were trained on clean speech only.*

Noisy Speech Conditions

Top 1 recognition rates under two training speech conditions and five test speech conditions are shown in Figure 7. The effect of multi-style training on error rate was not found to be significant (p=0.05, by McNemar test [5]) in clean test conditions, and was found to be significantly reduce the error rate by 22% to 66% in noisy test

conditions. Use of the word hypothesizer improved recognition performance (compared to exhaustive search by the fine match alone) for every test condition under multi-style training. The error reduction due to the hypothesizer was insignificant (2%) for 10 dB Car noise, but was 25% or more for each of the four other test conditions.

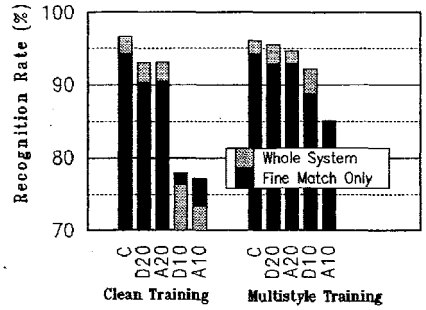

Figure 7: *Top 1 recognition rates for the fine match procedure (black) and recognition system as a whole (grey), under five test speech conditions: Clean, Datashow 20 dB, Car 20dB, Datashow 10 dB and Car 10 dB SNR. Word prototypes were trained on clean speech (left) or on both clean and noisy (Datashow noise at 10 dB SNR) speech (right).*

Alignment Time Results

The measured time for the alignment portion of the matching (independent of fixed overhead for analysis and phoneme similarity computation) is shown in the left side of Figure 8. The times reported here were for non optimized software. For a 100 word lexicon the whole system requires only 7.3% of the alignment time used by the fine match alone. For larger lexicons the alignment time reduction is yet larger, as shown in the right side of Figure 8.

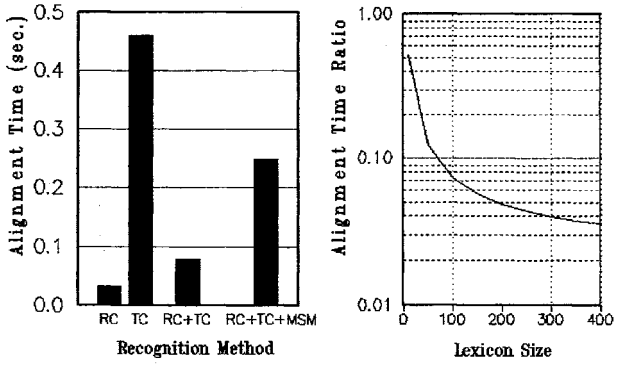

Figure 8: *Alignment time by stage (left) and alignment time ratio vs lexicon size (right). Alignment time is given for one test word aligned to 100 reference words. Corresponding alignment time for the fine match is 3.4 seconds. Alignment time ratio is the ratio of alignment time required by the whole system to the alignment time required for exhaustive search by the fine match alone.*

Conclusion

A multistage word hypothesizer based on reliably detected regions of high phoneme similarity has been proposed. A summary of the recognition performance and resource requirements of the hypothesizer alone, and in combination with the MSM fine match procedure, is shown in Table 1. On the 100 word Name recognition task, use of the word hypothesizer decreased alignment time to 7.3% of the time required by the fine match, while increasing the memory size of the reference data by 76%. Error rate was decreased significantly: by 30% or more ($p < 0.001$) for clean or Datashow noise-corrupted test speech at up to 10 dB signal-to-noise ratio.

	Alignment Time Ratio	Memory Size	Top 1 Error Rate (%)		
			Clean	20 dB	10 dB
Fine Match	100%	600	5.8	7.2	11.2
Hypothesizer	2.3%	457	5.1	5.7	12.2
Whole System	7.3%	1057	4.0	4.4	7.9

Table 1 *Alignment time ratio, number of parameters per reference word, and system error rate for three test speech conditions (Clean, 20 dB and 10 dB SNR Datashow noise). Multi-style training done on both clean and noisy conditions (Datashow noise at 10 dB SNR) was used to train all word models.*

As suggested by Table 1, the word hypothesizer may be useful by itself, as a low complexity speech recognizer. Alignment time, memory size and error rate under clean or mild noise conditions are in fact superior to the fine match procedure's. The robustness of the word hypothesizer's top 1 recognition performance under various other adverse conditions is under current investigation.

The multistage word hypothesizer, combined with the MSM fine match procedure, achieves low complexity, speaker-independent, medium-size vocabulary word recognition, suitable for implementation in inexpensive, small hardware. The word hypothesizer produced large reductions of computational complexity. On a 100 word task, alignment complexity was reduced by 93%, with significant error rate reduction for clean and noisy test conditions.

References

1. Hoshimi, M., M. Miyata, S. Hiraoka, and K. Niyada, "Speaker Independent Speech Recognition Method Using Training Speech from a Small Number of Speakers," Proc. ICASSP, vol. I, pp. 469-472, 1992.
2. Hoshimi, M., M. Yamada, and K. Niyada, "Speaker Independent Speech Recognition Method Using Phoneme Similarity Vector," Proc. ICSLP, vol. 3, pp. 1915-1918, 1994.
3. Ohno, Y., M. Hoshimi, S. Hiraoka, K. Niyada, and T. H. Applebaum, "A Study of English Model Speech Method," Proc. Acoustical Society of Japan, Spring 1995 (in Japanese).
4. Gong, Y. and J.-P. Haton, "Plausibility Functions in Continuous Speech Recognition: the VINICS System," Speech Communication, vol. 13, pp. 187-196, Oct. 1993.
5. Gillick, L. and S. J. Cox, "Some Statistical Issues in the Comparison of Speech Recognition Algorithms", Proc. ICASSP, pp. 532-535, 1989