Presented at ICASSP-93, Minneapolis Convention Center, Minneapolis, Minnesota, USA, April 27-30, 1993.

IMPROVING THE MS-TDNN FOR WORD SPOTTING

Torsten Zeppenfeld *Rick Houghton* *Alex Waibel*

School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213-3890 USA

ABSTRACT

Word spotting systems for continuous, speaker independent speech recognition are becoming more and more popular [1,2] because of the many advantages they afford over more conventional large scale speech recognition systems. Because of their small vocabulary and size, they are a practical and efficient solution for many speech recognition problems that depend on the accurate recognition of a few important keywords. We have implemented and tested an MS-TDNN version of such a system on two spontaneous continuous speech databases. These results, as well as several improvements are described below.

1. ARCHITECTURE

The basic Multi-State Time Delay Neural Network (MS-TDNN) word spotter is described in detail in [3]. This section gives a short summary of the highlights of our basic system.

1.1. MS-TDNN Word Spotter

Our word spotting system architecture is based upon the Time Delay Neural Network (TDNN) [4], and more recently the Multi-State Time Delay Neural Network (MS-TDNN) [5]. A diagram of the basic network architecture is shown in figure 1. This keyword spotting network consists of an input layer and a hidden layer, connected to a state layer and an output layer for each word to be spotted. The connections are represented by TDNN style weights, shifted through time. The activations for all units and states in the hidden and state layers are found using a standard TDNN feed-forward network algorithm. At the state layer, each keyword to be spotted is represented by a series of independent states. (see figure 2). A dynamic programming algorithm is performed starting at each time frame in order to find the best path through these state activations. The score of this optimal path represents the output score for the keyword at the time frame in question. The network thus outputs a score for each keyword at each time frame.

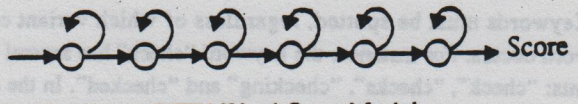

Fig. 2. Basic DTW Word State Model

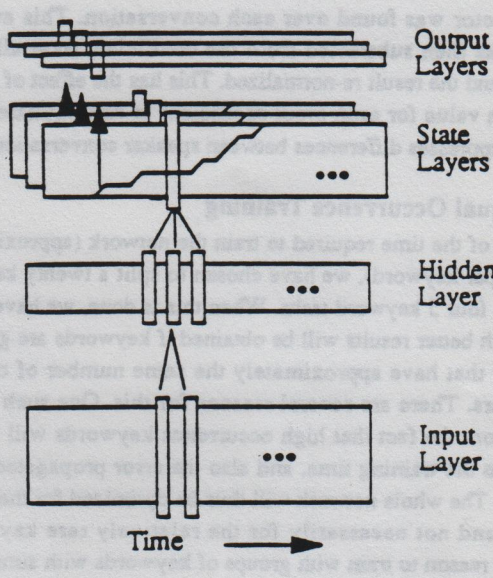

Fig. 1. System Architecture

1.2. Training the System

As shown in [6], Consistent training from the word level is essential for obtaining an optimal performance level. Thus, training of the word spotter is accomplished by backpropagating the word error from the word output unit, through the word states in the optimal path and down through the network whereever a keyword hit or false alarm occurs.

2. RECENT IMPROVEMENTS

Since our ICASSP-92 paper, we have studied several improvements to the system, such as: training with noise, average spectrum removal, equal occurrence keyword training, word duration modelling, state duration modelling, enforced minimum state durations, training with context frames, and keyword variant modeling. Each of these enhancements has given us an increase in performance. Below is a short description of these changes.

2.1. Training with Noise

The input vector to our network consists of mel-scale frequency coefficients. Currently we use 16 coefficients every 10 millisec-

onds. We have found that adding a small amount of random noise to these spectral vectors will improve our network's generalization performance. Currently we add a linear random value from -0.05 -> +0.05 to our input coefficients, which at this point have a range from zero to one.

2.2. Average Spectrum Removal

Because there are large differences in the quality of the recordings between the Stonehenge and Waterloo sections of the RoadRally database, and in order to help make the system more speaker independent, some preprocessing of the input signal was applied. The average value for each of the sixteen coefficients in the mel-scale input vector was found over each conversation. This average value was then subtracted from the coefficient over all input frames, and the result re-normalized. This has the effect of setting the mean value for each input coefficient to approximately 0.5, which suppresses differences between speaker conversations.

2.3. Equal Occurrence Training

Because of the time required to train the network (approximately 8 hours per keyword), we have chosen to split a twenty keyword task into four 5 keyword tasks. When this is done, we have found that much better results will be obtained if keywords are grouped together that have approximately the same number of training exemplars. There are several reasons for this. One such reason stems from the fact that high occurrence keywords will tend to dominate the training time, and also the error propagated to the weights. The whole network will thus be optimized for these keywords, and not necessarily for the relatively rare keywords. Another reason to train with groups of keywords with similar frequencies of occurrence is that the number of training tokens per keyword has a great effect on the number of epochs needed to train the network. It is thus important to match the number of training tokens available for each keyword.

2.4. Word Duration Re-scoring

One major difference between a false alarm and a true keyword hit in our system is the duration of the corresponding putative hit. To take advantage of this fact, we have taken statistics on the length of the keywords that appear in the training corpus, and use this information to re-score putative hits according to the length of the occurrence. After poisson distributions are fitted to the training data keyword lengths, the following equation is used to rescore putative hits in the testing corpus:

$$score = score + (WDW \times Poisson(Length))$$

where Length is the length of the putative hit, and WDW is the Word Duration Weight, a constant for each keyword, found to maximize the performance of the system on the training set.

2.5. State Duration Rescoring

In the same spirit as word duration rescoring, state duration rescoring can and was applied to the network. We noticed that there was a difference between false alarms and true hits in the length of time spent in some of the keyword states. After training, we collect statistics on the duration of each state in the optimal path for all keywords in the training set. Poisson distributions are then again fitted to the training data and used to re-score putative hits during the testing phase. The following equations are used to re-score the putative hits

$$LengthScore = \sum_{states} Poisson(StateLength)$$

$$score = score + (WSW \times LengthScore)$$

First, a score is found which represents a 'goodness' measure of the time spent in each state of a keyword. This LengthScore is then weighted by an optimal weighting factor (WSW) found from the training set. This score is then added to the original word score.

2.6. Minimum State Durations

Training along the optimal path in the state layer can develop unstable positive feedback, since states with higher activations will get more training, and will thus be even more active next epoch, while the less active states are ignored or even pushed down. After several epochs, this has the effect of having one state 'gobble up' most or all of the time during a keyword hit. This leads to poor performance. We found the easiest way to alleviate this problem is to force each state to be on for a minimum duration during the dynamic programming phase. This means that even very untrained states will get some minimum training during a keyword hit. This has a great effect on stabilizing the learning algorithm.

2.7. Training with Context

Our basic keyword spotter performs very well for longer keywords, but not so well for very short ones. To alleviate this problem, we notice that most keywords are imbedded in context speech which has some positive correlation to the keyword of interest. Thus, we add several context states at the beginning and at the end of each keyword, then train these states to be active for several frames before and after the keyword appears. This effectively increases the length of the keyword, making it easier to spot.

2.8. Keyword Variant Modeling

Keywords must be spotted, regardless of which variant of the word occurs. For instance, the keyword "check" has several variants: "check", "checks", "checking" and "checked". In the database markings, there is no marker to indicate the beginning of a

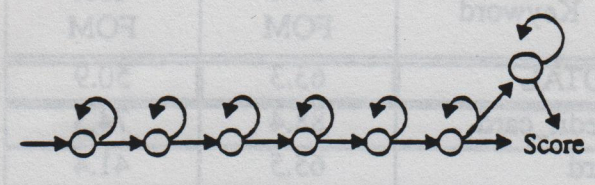

Fig. 3. Variant DTW Word State Model

variant ending. With the basic keyword model shown in figure 2, it is impossible to use variants of the basic keyword as training tokens. In order to be able to use the keyword variants as training tokens we have moved to the alternative keyword model shown in figure 3. We add one (or more) states for each alternative variant ending that we would like to model better. During training, we use the appropriate model when calculating the optimal path through the word. During testing, we let all alternative paths be possible, and the system chooses the state model which best fits the incoming speech.

3. RESULTS

Training and testing of our system was performed on two separate databases, the Roadrally corpus, and the new Switchboard corpus. A description of the scoring procedure, as well as the databases and current results are indicated below.

3.1. Performance Measure

The system's performance is measured by plotting the keyword detection rate for several false alarm rates per keyword per hour (fa/(kw*hr)). The keyword detection rate at a certain (nth) false alarm level is the ratio of the number keywords spotted to the number of keywords present, found before the nth false alarm occurs. By changing the threshold of the word-output units, the detection rate can be improved at the expense of increasing the number of false alarms. Thus one can obtain a Receiver Operator Curve (ROC). The Figure of Merit (FOM) for the system is the averaged keyword detection rate over the false alarms from 0 to 10 fa/(kw*hr).

3.2. RoadRally (RdR) Corpus

With the hope of creating a standard word-spotting database, NIST has distributed a database called the 'Stonehenge' Road Rally task (and an additional extension called 'Waterloo'). The database consists of approximately 140 speakers (both male and female) recording conversations, read paragraphs, and/or read keyword sentences. This speech contains 20 keywords, embedded in extraneous speech. Keywords can have variable suffixes, such as -s, -ed, -ing. The task is to spot the occurrences of these twenty different keywords, while minimizing the number of false detections. The Stonehenge portion of the database was recorded at 10KHz using a high quality microphone, while the Waterloo extension was recorded over telephone lines (also at 10KHz). Both sections are band-passed to simulate telephone quality speech. The database labelling consists of markers at the beginning and ending of all keywords present. The speech is not labelled phonetically, nor is the extraneous speech labelled.

3.2.1. Training and Testing Set

The official training set (ATSM) for the March 1992 Darpa Word Spotting evaluation consisted of 28 male read paragraphs from Waterloo plus 12 male conversations from Stonehenge. The official test set consisted of 10 male conversations from Stonehenge.

3.2.2. Results

The official results of the evaluation indicate our system to have an FOM = 72.2%. This figure compared favorably to those of other keyword spotting systems that took part in the evaluation. Table 1 shows the breakdown of the results according to the above improvements. Note that the improvements are additive. Several of the improvements became standard when we switched to the Switchboard corpus, so statistics for several individual tests are not available. Also, the experiments with minimum state durations, context states and variant word models were performed after switching to the Switchboard database.

Experiment	RdR	Swb
Basic Word Spotter	52.0	N/A
Noise Addition	55.2	42.0
Ave. Spect. Removal	64.4	43.8
Learn Equal Occurr.	69.2	N/A
Word Durations	69.3	N/A
State Durations	72.2	N/A
Min. State Durations	N/A	48.6
Context States	N/A	54.2
Variant Word Models	N/A	63.3

Table 1: Experimental Development Results

3.3. Switchboard (Swb) Corpus

There are several problems with the RoadRally Corpus, among them the fact that much of the database is read speech, the fact that some of it is not telephone quality speech, and the fact that the size is not very large. In the hopes of alleviating these problems, Switchboard was chosen. This Texas Instruments [7] created database was picked by NIST as the new official corpus for

future Word Spotting tasks. It consists of many (2500+) topic related recorded telephone conversations. These telephone (sometimes of poor quality) conversations were recorded at 8 KHZ. Out of all possible topics, "Credit Card" conversations were picked by NIST & DARPA for preliminary word spotting system comparisons. The official evaluation using this database took place in September 1992.

NIST has distributed conversations from 70 speakers. As in the Roadrally task, 20 keywords and their variants were chosen to be spotted. The task is again to spot the occurrences of these twenty keywords while minimizing the number of false detections.

3.3.1. Training and Testing Set

Currently, we use the first 25 male and 25 female speakers for training, and last 10 male and last 10 female speakers for cross-validation and development purposes. The Test set contains ten additional conversations never seen before.

3.3.2. Results

The official test results indicate our system to have an FOM = 50.9%. on the Switchboard database. Table 2 shows the breakdown for each keyword. The rows are ordered according to the number of instances of the keyword in the CVS set (keyword importance). It also shows the performance on the cross validation set (CVS) for comparison. While looking for an explanation for the rather large performance discrepancy between the CVS and the Official test set, we saw that the performance on the cross-validation set over time tended to be oscillatory. When we tried to decrease the learning rate to counter this, the average performance decreased. The network weights that were chosen to optimize the cross-validation set thus apparently were optimal for the cross validation set only.

4. CONCLUSIONS

Our state of the art MS-TDNN word spotter has shown its strength on the RoadRally database. With recent improvements, it has been adapted to the much more difficult Switchboard corpus with good results. Our word spotting system has proved to be a viable alternative to the much larger full vocabulary speech recognition systems. With relatively few parameters we are able to achieve good performance and speed on noisy, telephone quality spontaneous recordings.

5. ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of DARPA for this research.

Keyword	CVS FOM	Test FOM
TOTAL	63.3	50.9
credit_card	88.4	74.3
card	65.5	41.4
month	37.5	19.4
charge	61.8	63.8
interest	67.5	67.3
credit	76.8	68.5
money	39.0	42.8
dollar	45.2	53.1
cash	57.5	53.1
percent	54.5	61.7
check	63.6	57.5
visa	49.3	42.4
bank	18.1	35.9
american-express	93.8	83.6
twenty	72.6	39.9
discover	50.5	66.6
hundred	48.1	20.6
account	28.5	19.1
limit	33.3	9.2
balance	80.0	46.3

Table 2: SWB Results

6. REFERENCES

[1] Rose, R.C. and Paul, D.B., "A Hidden Markov Model Based Keyword Recognition System," ICASSP'90.

[2] Wilpon, J.G., Miller, L.G. and Modi, P., "Improvements and Applications for keyword Recognition Using Hidden Markov Modeling Techniques," ICASSP'91.

[3] Zeppenfeld, T. and Waibel, A., "A Hybrid Neural Network Dynamic Programming Word Spotter," ICASSP'92.

[4] Waibel, A.H., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K., "Phoneme Recognition Using Time-Delay Neural Networks," in IEEE Transactions on Acoustics, Speech and Signal Processing. 1989.

[5] Haffner, P., Franzini, M., and Waibel, A., "Integrating Time Alignment and Neural Networks for High Performance Continuous Speech Recognition," to be published.

[6] Tebelskis,J., "Performance through Consistency: Connectionist Large Vocabulary Continuous Speech Recognition," ICASSP-93

[7] Godfrey,J., Holliman,C., McDanial,J., "SWITCHBOARD: Telephone Speech Corpus for Research and Development," IEEE 1992.