ACOUSTICS-ONLY BASED AUTOMATIC PHONETIC BASEFORM GENERATION

B. Ramabhadran, L. R. Bahl, P.V. deSouza, M. Padmanabhan

IBM T. J. Watson Research Center, P. O. Box 218, Yorktown Heights, NY.
email:bhuvana@watson.ibm.com, phone: (914)-945-2976

ABSTRACT

Phonetic baseforms are the basic recognition units in most speech recognition systems. These baseforms are usually determined by linguists once a vocabulary is chosen and not modified thereafter. However, several applications, such as name dialing, require the user be able to add new words to the vocabulary. These new words are often names, or task-specific jargon, that have user-specific pronunciations. This paper describes a novel method for generating phonetic transcriptions (baseforms) of words based on acoustic evidence alone. It does not require either the spelling or any prior acoustic representation of the new word, is vocabulary independent, and does not have any linguistic constraints (pronunciation rules). Our experiments demonstrate the high decoding accuracies obtained when baseforms deduced using this approach are incorporated into our speech recognizer. Also, the error rates on the added words were found to be comparable to or better than when the baseforms were derived by hand.

1. INTRODUCTION

There has been considerable interest in telecommunications based speech recognition services that provide user configurable vocabularies. Name dialing is one such example of a telephony application, where it is necessary to have the ability to provide speaker dependent vocabularies for repertory dialing. This feature will enable the user to add a word(s) to their personalized vocabulary, for which an apriori spelling or acoustic representation does not exist in the speech recognition system, and associate that word(s) to a phone number to be dialed. Once the personalized vocabulary is configured, the user can subsequently dial the phone number by speaking the new word(s) just added to the vocabulary. In order to do this, proper deduction of the phonetic baseform is essential. This paper presents an automated method for generating speaker dependent acoustic representations (baseforms) from speech in terms of speaker independent subword acoustic units (phones), in order to build the personalized vocabulary. The baseform is generated using a sample utterance of the word to be added. Using this segment of speech, a phonetic representation is extracted using the ballistic labeler (described in Section 2). Subsequent utterances of this word are decoded by the speech recognition system using the newly added representation of the word.

Phonetic baseforms are the basic recognition units in most speech recognition systems. This paper describes a fast and novel method for generating phonetic transcriptions (baseforms) based on acoustic evidence alone without using any prior information regarding the pronunciation of the newly added words. This method is vocabulary independent and does not have any linguistic constraints (such as prior pronunciation rules). This paper also describes a series of experiments in which the phonetic baseforms are first deduced automatically using actual utterances of the new word, and subsequently used by the speech recognizer for decoding.

Many speech recognition systems (desktop applications, such as dictation transcription, and telephony applications, such as name dialing) provide the user with the ability to configure personalized vocabularies. Existing systems which provide this functionality fall into three main categories.

1. A speech recognizer that requires the user to enter the spelling of the item to be added to the vocabulary as well as pronounce it. A series of experiments in which the phonetic baseform is deduced automatically for a new word given its spelling by utilizing actual utterances of the new word in conjunction with a set of automatically derived spelling-to-sound rules is described in [1, 2].
2. A speech recognizer that uses just the user's pronunciation to build a personalized vocabulary for that user.
3. There is no speech recognition system involved. The user's audio is saved under a voice label. Subsequent utterances are compared with the saved audio before a decision is made. In the name-dialing task, this involves the selection of the phone number to dial.

All the above methods have their shortcomings. The first method cannot be used for telephony applications as it requires typed input from the user. It is possible for the user to spell out the word through the telephone, but this increases the enrollment time and also decreases the accuracy of the recognizer for subsequent decoding. A second problem relates to the fact, that sometimes the spelling of a word has very little correlation with its pronunciation. Many implementations of the second method exist in service over the public telephone network today. These are relatively simple unsupervised methods of obtaining speaker dependent pronunciations and require at least three utterances of the word during the enrollment procedure. Some of these systems may also require that prior information concerning the pronunciations of words being added exist either in the form of pronunciation networks or phonological rules.

The third method suffers from the drawbacks of having to store the audio for every new word added to the vocabulary and performance in terms of poor accuracy because of the simple comparison measures. A comparison of some unsupervised maximum likelihood decoding procedures is given in [4, 5]

This paper describes a method for generating phonetic transcriptions (baseforms) for obtaining speaker dependent word pronunciations that does not use any prior information regarding the pronunciation of words. For the name-dialing task, we obtain good recognition accuracies during the retrieval phase with only one utterance used during the enrollment procedure. The baseform thus generated is used by the speech recognizer during recognition, when the personalized vocabulary is active along with other speaker independent navigational commands such as "forward my calls to", "delete name", etc. The accuracy of the recognizer improves if an additional utterance is used during the enrollment procedure.

The structure of this paper is as follows. Section 2 describes the ballistic labeler algorithm used for the generation of phonetic baseforms. Section 3 describes a set of experiments that evaluate the recognition performance of the speech recognizer that uses the automatically deduced baseforms. Section 4 discusses the results and has suggestions for future work.

2. ALGORITHM FOR GENERATING BASEFORMS

This section describes the algorithm that is used for generating the phonetic baseforms from the acoustics alone. The goal here is to find the phone string P that maximizes $p(P|U)$, where U is the utterance for which the baseform is to be generated. The algorithm proceeds as follows. The acoustic data from the enrolled utterance is labeled in less than real time using the ballistic labeler. This involves the construction of a trellis of arc (sub phone units) nodes from the speech utterance. The probability of a transition occurring from one arc to another is determined by weighting the score obtained from a Hidden Markov Model (HMM) [3] with a precomputed arc to arc transition probability obtained from any training corpora. At any time frame the set of active nodes in the trellis is defined as the nodes with scores greater than a certain pruning threshold. Once the entire utterance has been processed, a back-tracking procedure is employed that traces the best arc-predecessor from the end of the utterance, forcing silence at the beginning and the end of the utterance. Thus, a sequence of phonetic arcs are obtained from which a phone sequence (baseform) is derived for that enrolled utterance. This corresponds to one pronunciation for the word enrolled by the user that will be used subsequently for recognition [6].

2.1. Precomputation of arc transition probabilities

This algorithm requires precomputation of some statistics to be used in the construction of the trellis. The first step involves the computation of the arc to arc transition probabilities. In order to do this, leaf to leaf transition probabilities have to be computed first. A leaf is a context dependent arc. A phone is made of three sub units called arcs, corresponding to the three states of a HMM (for eg., the phone AA is made of arcs, AA-1, AA-2 and AA-3). An initial set of acoustic models are built using any training corpus. Using the same training corpus, alignments of the speech segments at the leaf level to the transcribed text is obtained using the Viterbi method. From these alignments the number of times that any one leaf transitions to any other leaf can be computed. The leaf to leaf transition probabilities can therefore be computed as, $prob_{leaf-trans}(i,j) = N_{ij} / \sum_j N_{ij} \forall j \in N_L$

where $prob_{leaf-trans}(i,j)$ is the probability of transition from leaf i to leaf j, N_{ij} is the number of transitions from leaf i to leaf j and the denominator term is the summation of the number of transitions from leaf i to all possible successors of leaf j.

The arc to arc transition probabilities are computed from the leaf to leaf transition probabilities by mapping each leaf to its corresponding arc and computing $prob_{arc-trans}(i,j) = Na_{ij}/Na_i$ where Na_{ij} is the sum of transitions of all leaves of arc i to arc j and Na_i is the sum of all transitions of all leaves of arc i to all leaves of any other arc.

The score of each node in the trellis is weighted by its corresponding arc to arc transition probability.

2.2. The ballistic labeler

The processing of the speech utterance occurs on a frame to frame basis, where each frame is a centi-second of speech. A trellis of arc nodes is constructed where each node corresponds to a frame of speech. The utterance is assumed to start and end in silence, i.e., the first and the last frames of speech correspond to a silence arc node in the trellis. The scores for all leaves for the current frame is obtained from the HMM and the arc score (node score) is the score of the highest scoring leaf for that frame. A trellis of arc nodes is constructed, where every arc node can transition to any other arc node with no constraints. For the first frame, the initial probability for the source arc node is defined as $1/N$ where N is the number of *live* nodes in the system. Since it is forced to be a silence node, N is 1 for the first frame. Before proceeding to the next frame of speech, the scores of all nodes that are *alive* at this time frame are multiplied by the precomputed arc transition probabilities.

$$alphaScore_t(arc,j) = prob_t(arc) * prob_{arc-trans}(arc,j) \forall j$$

The updated scores will be referred to as the *alphaScore* values. As every node can transition to any arc node, the number of possible successors for every node is the same as the number of arcs in the system. The best predecessor of every node at time $t+1$ is computed as the node which results in the best *alphaScore.* The alpha scores of the nodes are multiplied by the output probabilities obtained from the HMM. The HMM's are trained at the leaf-level.

$$alphaScore_t(arc,j) = alphaScore_t(arc,j) * prob_{HMM}(j,t)$$

Maxalpha is computed as the the maximum value of the alpha scores computed at time t between arc node i and j as max $(alphaScore_{best-predecessor}(j)\ )$ for all j. To make for efficient back tracking, the best predecessor for every arc node and its corresponding alpha scores are stored. The scores of all successor nodes are normalized with the *Maxalpha* value. These updated scores will serve as the initial set of probabilities for the next frame of speech.

Next, the set of arc nodes that will be alive for the next frame of speech is determined. A simple pruning procedure is used, wherein if the alpha score of a node exceeds a pruning-threshold the node is marked *alive*. All nodes with scores below this threshold are not expanded further in the trellis. Since each node is an arc, or a sub-phone, each alive node in the trellis best describes the local region of speech. The above steps are repeated for all the frames, while storing the best predecessor for every time frame. The pruning procedure is applied at every time frame, and only the *live* nodes are allowed to proceed to the next iteration of the algorithm. This is the forward-pass of the algorithm. (See Figure 1)

2.3. Back tracking

Once the trellis for the entire speech utterance has been constructed, the final step is to determine the best acoustic representation of the utterance. As stated earlier, silence is forced at the beginning and end of the utterance. Starting from the last frame of speech, which corresponds to a silence node in the trellis, the back-tracking procedure is applied. In the forward-pass of the algorithm, the index or label of the arc node that gave the best *alpha* value at any time instant was stored. From the silence node, the arc corresponding to the previous time slice in the trellis $(arc_{n-1}(i))$ that was determined to be the best predecessor is the arc that best describes the speech in that time frame, i.e., arc i best represents the speech at the $n-1$ th frame. From arc i, the next best predecessor at time n-2 is retrieved and this procedure continues until the silence node corresponding to the first frame of speech is reached. The sequence of arcs thus obtained is the acoustic representation of the speech utterance. (Refer Figure 1)

Next these arcs are mapped to their respective phones using a simple map. The time frame at which two consecutive arcs belong to different phones is defined as a transition point and a new phone starts at this instant. An entire phonetic sequence is thus obtained and this is the baseform of the speech utterance.

The pruning procedure is used for reduction of storage requirements, while the tracking of best predecessor in the forward pass of the algorithm provides for fast back tracking. This enables the ballistic labeler to operate under faster than real time speeds, which is essential in order to make this a viable technique for practical applications.

An example of an arc sequence and a phonetic sequence derived thereafter for a sample utterance 'ROSE' is illustrated in Figure 1.

3. EXPERIMENTS

This method of generating baseforms was evaluated on the name-dialing task, using an in-house data collection software for telephony data collection. Two local databases were built for evaluating the baseform generation algorithm. The first database(DB I) was built using ten speakers and each speaker asked to enroll twelve different names. These names were chosen at random from a currently operational name-dialing application and were complicated enough to produce user-specific pronunciations (for example, CELESTINO DOMINGUEZ, THOMAS CABAN, TONY VAIANISI, etc.). Each participant made ten calls, from as many different phones as possible. These included cellular, digital and analog phones under different environment conditions, such as hallways and the cafeteria. The vocabulary for the recognition task included these names along with other navigational commands such as 'forward my calls', 'call return', 'delete a name', etc. resulting in a 65 word vocabulary. For the second database (DB II), ten speakers with a variety of accents called in from three different phones (digital, cellular and speaker phone conditions). These speakers were asked to read fifty items, that included names, such as, ROSE, ANTHONY FABRIZIO, DAD, MOM, etc.), and other command words, such as, 'HELP', 'CANCEL', 'CALL RETURN', etc.. The vocabulary for this database was made up of 105 words. For both the databases, baseforms were generated from one speech utterance and the remaining calls (9 for DB I) were used as the test bed for recognition with the newly generated baseforms. The decoding results obtained when baseforms were generated with one acoustic token (utterance) and two tokens are tabulated in Table 1. The numbers presented in the table are the average percentages obtained over all the speakers under all the conditions under which the data were collected. While it can be seen that the recognition accuracy improves marginally with the use of an additional enrollment utterance, the accuracy of the recognizer is very good for the newly added words even when one utterance was used for the baseform generation.

Method Employed	DB I	DB II
Baseforms from 1 acoustic utterance	96.4%	98%
Baseforms from 2 acoustic utterances	96.6%	98.2%
Hand-written Baseforms	97%	94.5%

Table 1. Recognition performance for two databases using baseforms generated with acoustics alone and hand-written baseforms

4. CONCLUSIONS AND FUTURE WORK

The goal of this work was to automatically determine phonetic baseforms for words not used in the recognizer and evaluate the performance of the current speech recognition system for those words. Visual inspection of the baseforms generated, indicated substantial discrepancies between the automatically generated baseforms using acoustics alone and hand-written ones for almost one-tenth of the data. Despite this, the recognition accuracy seemed to be better when using these baseforms as opposed to the ones generated by a linguist. Most of the errors were made in situations where the telephone channel conditions were really noisy, with several people talking in the background, or when the user pronounced the word very differently from call to call. As expected, it was observed that if baseforms were generated with two utterances, both from clean and noisy scenarios, the recognition accuracies improved slightly. Our algorithm produced the following baseform for the utterance 'SHARON KEN', D$ SH EH R AE N K EH N X S X D$ as opposed to the hand written version which had two pronunciations, namely, SH AE R AX N K EH N and SH EH R AX N K EH N. It can be seen that a 'S' phone has been introduced into the baseform sequence. Most of the errors observed arise from these sporadic 'S' , 'V', and 'F' phone insertions, particularly under noisy conditions. Our system uses a 54-phone set where D$ and X

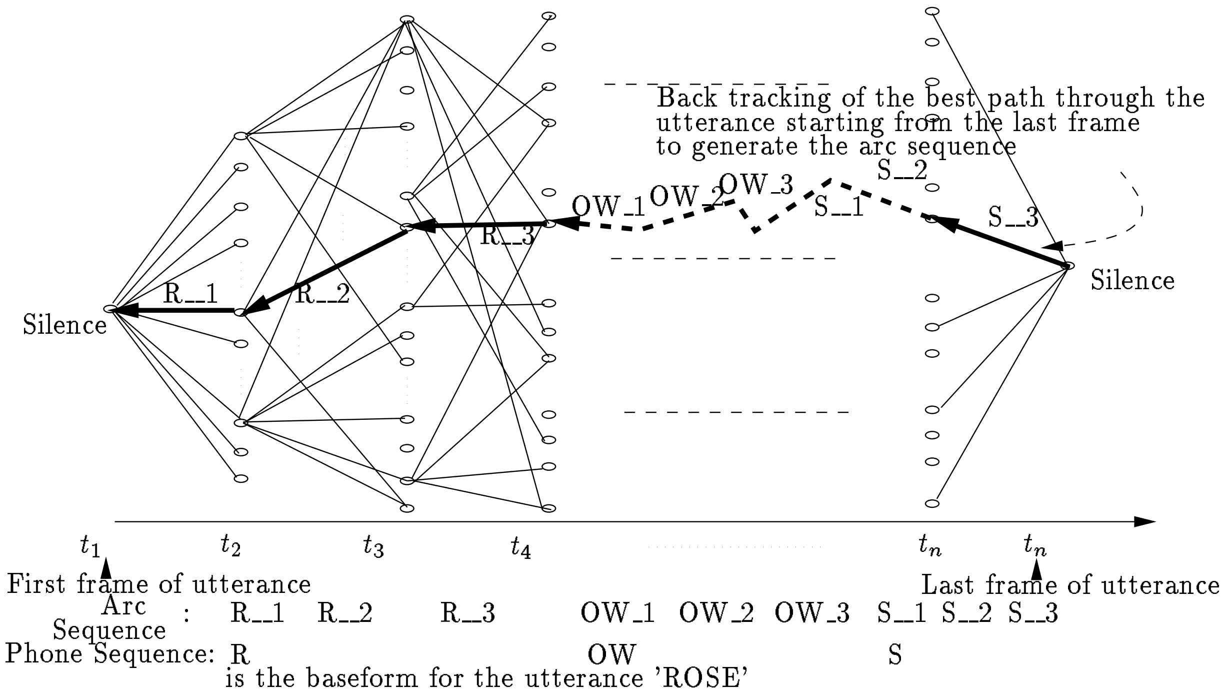
Back tracking of the best path through the utterance starting from the last frame to generate the arc sequence
Silence
R_1
R_2
R_3
OW_1
OW_2
OW_3
S_1
S_2
S_3
Silence
t_1
t_2
t_3
t_4
t_n
t_n
First frame of utterance
Last frame of utterance
Arc Sequence : R_1 R_2 R_3 OW_1 OW_2 OW_3 S_1 S_2 S_3
Phone Sequence: R OW S
is the baseform for the utterance 'ROSE'

Figure 1. Trellis construction and the generation of a baseform for a sample utterance.

are the silence phones.

The good performance of this algorithm can be attributed to the following fact. The training of the HMM's are done at the leaf level even though the trellis computation and Viterbi search is done at the arc level (a third of a phone) and this fine level of detail that is included in the models, accounts for the high performance accuracy. In order for this algorithm to be of any use in a practical application, the generation of baseforms should be done at speeds much less than real time. To save on computations, a trellis of arc nodes is used. However, a trellis of leaf nodes can also be used for the same purpose without considerable loss in accuracy.

Secondly, in the mapping from arc to phone nodes to obtain the final phonetic sequence, a language model constraint (such as bi-phone or tri-phone statistics) can be imposed which will eliminate certain weird phone transitions that could be errors generated either by the pruning procedure or by the presence of noise in the speech frame.

At present, alternate pronunciations (baseforms) are generated only with an additional utterance during enrollment. Instead, from the trellis, the N best predecessors can be used to generate several combinations of phone sequences that would combinatorily generate multiple pronunciations for that utterance, thereby improving the performance of the speech recognizer.

In conclusion, we believe that we have a viable technique for generating good phonetic baseforms that give a high decoding accuracy with our speech recognizer. This is particularly useful for our telephony toolkit, where personalized vocabularies are a must. Work is currently under way to employ this algorithm in other components of the speech recognizer, so that phonetic baseforms existing in the system can be adapted to provide improvements in accuracy for spontaneous speech applications.

Acknowledgment The authors would like to thank Abraham Ittycheriah at IBM T. J. Watson Research, for several fruitful discussions and help provided during the course of this work.

REFERENCES

[1] L. R. Bahl, S. Das, P.V. deSouza, M. Epstein, R. L. Mercer, B. Merialdo, D. Nahamoo, M.A. Picheny, J. Powell, "A utomatic Phonetic Baseform Determination", Proc. Speech and Natural Language Workshop, pp. 179-184, June 1990.

[2] J. M. Lucassen and R. L. Mercer. "An Information Theoretic Approach to the Automatic Determination of Phonemic Baseforms", in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 4 2.5.1-42.5.4, 1984.

[3] L.R. Bahl et al., "A Fast Approximate Acoustic Match for Large Vocabulary Speech Recognition", IEEE Transactions on Speech and Audio Processing, vol. 1, no. 1,pp 59-67. January 1993.

[4] R. C. Rose and E. Lleida "Speech Recognition using Automatically Derived Baseforms", pp 1271-1274, ICASSP 1997.

[5] R. C. Rose et al., "A User-Configurable System for Voice Label Recognition" ,Proc. Int. Conf. on Spoken Lang. Processing, October 1996.

[6] L.R. Bahl et al. "Performance of the IBM Large Vocabulary Continuous Speech Recognition System on the ARPA Wall Street Journal Task." vol 1, pp 41-44, ICASSP 1995.