VOCABULARY INDEPENDENT SPEECH RECOGNITION USING PARTICLES

E. W. D. Whittaker, J. M. Van Thong and P. J. Moreno

Compaq Cambridge Research Laboratory
Cambridge, MA 02142 USA

ABSTRACT

A method is presented for performing speech recognition that is not dependent on a fixed word vocabulary. *Particles* are used as the recognition units in a speech recognition system which permits word-vocabulary independent speech decoding. A particle represents a concatenated phone sequence. Each string of particles that represents a word in the one-best hypothesis from the particle speech recognizer is expanded into a list of phonetically similar word candidates using a phone confusion matrix. The resulting word graph is then re-decoded using a word language model to produce the final word hypothesis. Preliminary results on the DARPA HUB4 97 and 98 evaluation sets using word bigram re-decoding of the particle hypothesis show a WER of between 2.2% and 2.9% higher than using a word bigram speech recognizer of comparable complexity. The method has potential applications in spoken document retrieval for recovering out-of-vocabulary words and also in client-server based speech recognition.

1. INTRODUCTION

Most speech recognition systems ignore the problem of words that are not in the recognizer's vocabulary. At most an attempt is only made to minimise the effect of such out-of-vocabulary (OOV) words by selecting a vocabulary that is closely matched to the domain and that is as large as possible. In the area of spoken document retrieval there are almost always OOV words with respect to the vocabulary that is used. Potentially it is these words that are also the most interesting for indexing purposes so it is undesirable to simply ignore them. Methods for surmounting this problem have focused primarily on phone or syllable-based recognition systems [1] which place no restrictions on the words to be recognized. Due to the poor error-rates of phone-based recognizers, in general phone lattices must be stored and searched anew each time a query is made. The search thus scales approximately linearly in the size of the data. This represents a serious deficiency compared to word-based retrieval techniques. Word-based indexing involves a simple look-up of the query word in a hash table to retrieve documents in which the query word occurs. This search is approximately constant in the size of the data.

Alternative methods have been proposed that combine the advantages of both methods for example by storing concatenated sequences of three or four phones in an index [2, 3] as for word-based retrieval. In this paper we present a further variant in which recognition is performed using units that may be thought of as lying somewhere between words and phones. These so-called *particle* units represent word-internal concatenated phone units and are determined automatically. A speech recognizer is built using concatenated triphone acoustic models to represent the particle units and a language model to represent the linguistic dependencies between particle units. By recognizing particles from speech, the decision on a set of words to recognize can be postponed. The recovery of words from the particle hypothesis is performed as a post-processing step once the word vocabulary has been defined and a corresponding dictionary and language model constructed. Conventional word-indexing techniques can then be used.

Another potential application of particle-based recognition is in client-server speech applications. By incorporating a particle-based recognizer on the client side, only the particles themselves need to be transmitted to the server. The server then performs the necessary post-processing to convert the particle hypothesis into a word hypothesis which is then transmitted back to the client. Such a method constitutes a parsimonious representation for data flow between client and server. In addition, the vocabulary of the recognizer on the client side can be fixed while the vocabulary on the server side can be adapted easily to account for the changing context or environment of the client.

2. SYSTEM OVERVIEW

The system to perform word-vocabulary independent speech recognition can be divided into three components: 1) the particle-based speech recognizer 2) the expansion of the particle hypothesis into a graph of word candidates and 3) a search for the highest scoring word sequence. The steps involved are shown in Figure 1.

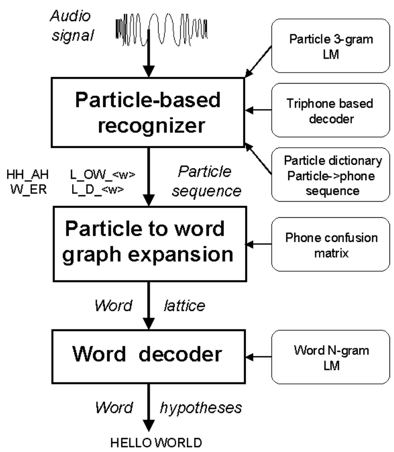

Fig. 1. Digram showing steps in conversion of audio signal to word hypothesis.

In order to construct the particle-based speech recognizer, a set of particle units is determined automatically in isolation from the acoustic data. The algorithm, which is described in Section 3.2, decomposes words into particles so as to maximise the leaving-one-out likelihood of a particle bigram language model on the training data. The particles are phoneticized word internal units and particles which occur at the ends of words are attached to an identifier that defines the particle as forming a word boundary. Acoustic models are constructed for each particle by concatenating triphone HMMs that have been trained in a manner similar to that for training triphone HMMs for acoustic word models. Acoustic modelling is described in Section 3.3. A conventional back-off language model is also built using particles instead of words as the modelling units.

During recognition, the top scoring particle hypothesis from the particle-based recognizer is first expanded into a word graph of phonetically similar word candidates using a phone confusion matrix. Each word in the graph has an associated pseudo acoustic score. This expansion is described in Section 4.1. The word graph is then re-decoded using a conventional stack search algorithm with a word language model to produce the final word hypothesis. Re-decoding is described in Section 4.2.

3. DETERMINING PARTICLE RECOGNITION UNITS

In this paper, particles are defined to be within-word sequences of phones obtained from the phonetic representation of words. The particles are used as the recognition units in a speech recognition system which also uses a particle-based language model to provide conditional probabilities between sequences of particles.

3.1. Particle-based language modelling

To obtain particles we consider the automatic generation of a deterministic decomposition function U between words and a set Ψ of particles u_i

$$U : w \rightarrow U(w) = u_0, u_1, \ldots, u_{L(w)-1} \quad u_i \in \Psi.$$

where word w is decomposed into a sequence of $L(w)$ particles. Identification of word boundaries at the particle level is necessary to ensure a deterministic mapping from a sequence of particles back to the word-level, even if the identity of the words themselves is ambiguous. Therefore, a <w> symbol is always attached to the terminal particle $u_{L(w)-1}$ in the decomposition of word w to denote a word boundary.

Given an algorithm to determine U, the words in some training text can be decomposed into their component particles. A language model can be built in an identical manner as for word language models. Relative frequencies of the occurrences of particle N-tuples are used to compute the conditional particle N-gram probabilities and smoothed accordingly.

3.2. Particle selection algorithm

The greedy particle selection algorithm used in this paper is described in [4, 5] but differs in three main respects. Firstly, a leaving-one-out optimisation criterion is employed; secondly all vocabulary words are first mapped into their phonetic representation using one character per phone; and thirdly, the set of all unique words in the training corpus is chosen as the vocabulary from which to determine particles.

The particle selection algorithm uses only the word unigram and bigram statistics from the training data and a list of all possible candidate particles of different lengths. This list only contains those particles which actually occur *within* words of the vocabulary. Initialising the algorithm involves decomposing all words into their constituent single phones. The contents of the set of particles Ψ at initialisation therefore comprise all single phones which occur in words of the vocabulary and all single phones with a <w> appended. Single phones must always appear in the final set since they may be necessary as *filler* particles to complete a decomposition which does not divide exactly into larger particles. The algorithm is described concisely by the following steps:

1. **Initialisation**:
   - $l = 1$
   - decompose words into l-phone particles
   - compute leaving-one-out log-likelihood of training data
2. $l = l + 1$
3. **Iterate** $\forall$ l-phone candidate particles u^{can}:
   - 'insert' particle u^{can} in all words w
   - compute change in training set leaving-one-out log-likelihood
   - 'remove' particle u^{can} from all words w
4. Insert best l-phone particle into Ψ and permanently in all words
5. If desired number of particles obtained then **terminate**
6. If no particles remaining then **terminate**
7. If improvement goto **step 3**, else goto **step 2**

Each iteration involves a search over a set of particles of a fixed length l phones, at the end of which the particle that gave the greatest increase in leaving-one-out log-likelihood is permanently added to the final set of particles. The leaving-one-out log-likelihood of the training data computed using a particle bigram language model is given by:

$$LL_{loo} = \sum_{i=1}^{N_P} \log P_{loo}(u_i \mid u_{i-1}), \tag{1}$$

where each word w in the text is decomposed using $U(w)$ into its constituent particles and N_P is the total number of particles in the text when all words have been decomposed. All particle unigram and bigram counts were discounted using absolute discounting. Backed-off probability estimates are necessary for events that only appear once in the training data.

The best particle from the inner loop in the algorithm above is chosen to be the particle that gives the greatest increase in leaving-one-out log-likelihood and for which the increase is greater than some threshold value. The threshold value can be used to determine the number of particles that end up in the final set of particles.

The order in which particles are selected affects the selection of all subsequent particles. Since the algorithm only accepts configurations which result in an increase in the optimisation function, the algorithm is guaranteed to converge, however due to its greedy nature it is only likely to find a locally optimal solution. In these experiments the algorithm is only used to determine a set of particles up to some maximum size l_{max}.

3.3. Particle-based Acoustic Modeling

The acoustic training followed the approach commonly used in large vocabulary speech recognizers [6]: Monophone or Context-Independent (CI) training starting from flat distributions, unclustered state Context-Dependent (CD) triphone training starting from cloned CI models, CART based tree clustering of unclustered states, and finally mapping of each unclustered triphone state sequence into clustered states followed by several iterations of the Baum Welch algorithm increasing the number of learned Gaussians per clustered state.

The HMM architecture of our system is based on three states per triphone with self transitions and transitions to the next state, 39 phone units and 10 filler models to cover spurious sounds, 156,000 possible triphones, 6,000 clustered states, and 16 Gaussians per state.

One advantage of a word boundary particle recognizer is the reduced vocabulary size. Unlike word based large vocabulary speech recognition systems, where training and testing vocabulary sets are often different, our particle based system uses a fixed vocabulary of 8155 particles. This small vocabulary makes recognition faster by reducing the search space.

4. RECOVERING WORDS FROM PARTICLES

4.1. Expanding word hypothesis from particles

The particles output by the particle speech recognizer are decomposed into a sequence of corresponding phones. As word boundaries are known and labelled, each hypothesized word is now described as a phone string. Each phone string is then compared to every word of the vocabulary using a pronunciation distance metric.

Computing this metric uses a standard string alignment algorithm. The insertion, deletion, and substitution costs are obtained from a pre-computed phone confusion matrix. In addition to the matching cost, a length penalty is applied. This is computed by evaluating the phone string length difference between the decoded phone string and the pronunciation from the dictionary. The word pronunciation distance is then used to sort the whole vocabulary, the most likely word being placed at the top of the list [7, 8].

The phone confusion matrix used to compute the word distance metric was trained using the TIMIT corpus, a collection of 6,300 short, hand-labeled utterances. Training consisted of running phone recognition on all utterances, then aligning the hypothesized results with the hand labelled transcriptions. The alignment routine used the same cost for deletion, insertion, and substitution, regardless of the phones involved. Alternative approaches are possible for training the confusion matrix, including the use of phone classification, EM, or genetic algorithms.

4.2. Re-decoding with word language models

The expanded word list for each phone string forms a lattice of words along time. Each word frame contains the n-best words sorted by the word pronunciation distance as described previously. By construction, all words within the same frame have the same time boundaries. A standard stack decoder is used to compute the most likely sequence of words through the lattice [6].

The best score computed at every step of the search combines the pronunciation score as previously described, and an N-gram word probability. For every word of the lattice, a look-ahead score is pre-computed that will provide an upper bound value of the expected score at the end of the word sequence from that word cell. The sum of the best score and look-ahead scores is used as a key for inserting an active word lattice cell into a sorted list of active cells, or stack. At every step of the search, the word cell with the highest score is popped off the stack, and scored against all the possible next words of the next frame. To make the search more tractable, only the top 100 active paths within each frame are kept. When the last frame is scored, the most likely sequence of words, or best path through the lattice, is returned.

The depth of the lattice depends on how many words are expanded per word phone string. Since a very large vocabulary of over 250k words is used, experiments showed that at least 10 words needed to be generated for each word phone string to account for words that are mis-spelled in the dictionary that have the same pronunciation.

5. EXPERIMENTAL WORK

Speech recognition experiments were performed on both the 1997 and 1998 DARPA HUB4 evaluation sets using both a particle-based and a word-based speech recognizer.

The language model training data used for determining particles and also to build the language models for both the particle-based and word-based recognizers comprised around 160 million words of broadcast news texts from the 1996 HUB4 evaluation provided by the LDC. The particle selection algorithm described in Section 3.2, was used to select 8155 particles with $l_{max} = 3$. A Katz back-off particle trigram language model was built that had 8.7 million N-gram parameters. A Katz back-off word bigram model containing 8.3 million N-gram parameters and a word unigram model were also built. The top 65,000 words in the language model training were used for the vocabulary.

The same acoustic training data was used to build the acoustic models for both the particle-based and word-based speech recognizers. Three-state triphone acoustic models with 16 Gaussians per state were trained separately for each recognizer using the 62 hours of HUB4 1996 acoustic training data provided by the LDC. The states were then tree-clustered into 6000 tied states.

The word expansion and re-decoding used a word dictionary of 250k words as did the word unigram and bigram language models which had 250k and 9.2 million parameters respectively. Each particle sequence between word boundaries was expanded into 200 confusable words in the word graph.

The word error rates obtained when the particle hypothesis was decoded into words is compared against the word bigram speech decoder output in Table 1.

System	Word N-gram	WER% 1997	WER% 1998
Particle	1	33.2	30.7
Particle	2	31.6	28.6
Word	2	28.7	26.4

Table 1. Word error rate of word bigram decodings and particle decodings recovered using word unigram and bigram language models on the DARPA HUB4 97 and 98 evaluation sets.

It was found that the word bigram re-decoding using the 250k vocabulary of the one-best particle hypothesis on the 1997 data recovered 18 OOV words (i.e. words outside the word decoder's 65k vocabulary) of which there were 12 unique words. In the 31,532 words of the evaluation reference file there were 89 OOV words that could potentially have been recognized with the 250k dictionary of which 60 were unique. This represents a 20% recovery of previously OOV words.

An additional advantage of particle speech recognition was found to be its speed. Using a particle trigram was found to be three times faster than decoding using a word bigram.

6. DISCUSSION

The preliminary experimental results show that our approach still performs well although the decoding has been split into several stages for functional and performance reasons. First, our approach allows the recovery of OOV words without rescoring the acoustic models. Second, the final word decoding stage can be easily performed with specialized dictionaries on the particle-based representation which is very compact.

The experiments reported in this paper represent very preliminary results. No efforts have been made to optimize parameters in the multiple modules that compose our vocabulary independent speech recognizer. For example, the language weight in the particle-based recognizer has not been fine tuned. Similarly, alternate pronunciations for particles have not been investigated.

The particle building procedure is also very preliminary. For example, no effort has been made to guarantee some poorly pronounced and yet common words such as function words ("THE", "BUT"...) are modelled with their own particles.

The phonetic confusion matrix has also been derived from a different corpus. Ideally, we would like to use a similar corpus for the construction of this matrix.

Finally, because we are using a lattice where all words within the same frame have the same time boundaries we are severely limiting the final stack decoder search. In particular, we do not allow word deletions nor word insertions. Better results can probably be obtained by using the particle lattice as input to the word decoder, rather than the most likely particle sequence.

Our analysis of the OOV shows that we can recover close to 20% of the OOV words. This is done at a reduced computational cost as compared with a full acoustic word decode.

7. CONCLUSION AND FUTURE WORK

We have shown that a particle-based recognizer can give comparable word error rates to a conventional word-based recognizer. Our method has the potential of greatly reducing the processing time of large corpora. When new words appear our approach allows a simple re-processing of previous acoustic particle decodes. It also makes feasible the use of extremely large vocabularies since we only re-score a particle based input.

In the future we plan to address the shortcomings of this approach. We will explore the use of alternate pronunciations. We will also pay more attention to the particle selection procedure and will explore the use of longer context word language models (6-grams) using the word stack decoder.

Finally, we plan to explore the use of this approach for audio indexing of very large corpora. One major limitation of word-based indexing is OOV query words. Since re-processing the audio of whole corpora is infeasible, our approach would allow us to perform word decoding with specialized vocabularies at low cost.

8. ACKNOWLEDGEMENTS

The authors wish to thank Beth Logan, James Christie, Dave Goddeau and Bhiksha Raj for helpful discussions on the work in this paper.

9. REFERENCES

[1] U. Glavitsch and P. Schäuble, "A System for Retrieving Speech Documents," in *Proceedings of Special Interest Group on Information Retrieval*, 1992.

[2] K. Ng, "Information Fusion for Spoken Document Retrieval," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2000.

[3] H. Wang, H. Meng, P. Schone, B Chen, and W.K. Lo, "Multi-scale Audio Indexing for Translingual Spoken Document Retrieval," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2001.

[4] E. W. D. Whittaker and P. C. Woodland, "Particle-based Language Modelling," in *Proceedings of the International Conference on Spoken Language Processing*, Beijing, China, 2000.

[5] E.W.D. Whittaker, *Statistical Language Modelling for Automatic Speech Recognition of Russian and English*, Ph.D. thesis, Cambridge University, 2000.

[6] F. Jelinek, *Statistical Methods for Speech Recognition*, The MIT Press, Cambridge Mass., 1997.

[7] Coletti P. and Federico M., "A two-stage speech recognition method for information retrieval applications," in *Proceedings of the European Conference on Speech Communication and Technology*, 1999.

[8] E. Pusateri and JM. Van Thong, "N-Best List Generation using Word and Phoneme Recognition Fusion," in *Proceedings of the European Conference on Speech Communication and Technology*, Aalborg, Denmark., 2001.