Two-pass Algorithm for Large Vocabulary Continuous Speech Recognition

Valeriy Pylypenko
Department of Speech and Synthesis
International Research/Training Center for Information Technologies and Systems,
Kyiv, Ukraine
valery_pylypenko@mail.ru

Abstract

This paper presents a two-pass algorithm for Extra Large (more than 1M words) Vocabulary COntinuous Speech recognition based on the Information Retrieval (ELVIRCOS). The principle of this approach is to decompose a recognition process into two passes where the first pass builds the word subset for the second pass recognition by using information retrieval procedure. Word graph composition for continuous speech is presented. With this approach a high performances for large vocabulary speech recognition can be obtained.

1. Introduction

There exists a necessity for speech recognition with a huge numbers of alternatives without any syntax restriction for input word sequence.

For example, during the voice control of a computer it is impossible to predict the subsequent word on the basis of several previous ones because it is defined by control logic, instead of text properties. From the other hand there is a necessity for growth the volume of the dictionary to capture all possible synonyms of the same command caused by difficulty for users to remember the single command name variant.

A next example concerns the text dictation. Using such systems is limited by the texts, which are statistically similar to one where statistics were collected. Additional editing of the collected text demands the presence of all words in the actual dictionary.

Thus, there are applications where it is desirable to have as large active dictionary as possible, in future to cover all words for the given language.

The additional information to restrict the number of alternatives can be received from a speech signal immediately. For this purpose it is proposed to execute preliminary trial recognition by using the phonetic transcriber. Phonemes sequence analysis allows to build the queries flow. Applying the information retrieval approach considerably limits the number of alternatives for recognition.

The next sections describe a two-pass algorithm. First, the baseline recognition system is presented to compare with the proposed algorithm. Then, two variants of algorithm for isolated and continuous speech are described. The fulfilled experiments have shown efficiency of the offered techniques.

2. The Baseline Recognition System

Proposed approach is applicable for any recognition system where phonemes and phoneme-by-phoneme recognition (phonetic transcriber) are present but the number of phonemes no more than approximately 500 units. As a reference system HMM-based HTK toolkit [1] is used.

2.1. Feature Extraction

The speech signal is converted into a sequence of vector parameters with a fixed 25 ms frame and a frame rate of 10 ms. Then each parameter is pre-emphasized with filter P(z) = 1-0.97z-1. Hamming

window is applied. A fast Fourier transform is used to convert time domain frames into frequency domain spectra. These spectra are averaged into 26 triangular bins arranged at equal mel-frequency intervals. 12 dimensional mel-frequency cepstral coefficients (MFCCs) are obtained from cosine transformation and lifter. The log energy is also added as the 13th front-end parameter.

These 13 front-end parameters are expanded to 39 front-end parameters by appending first and second order differences of the static coefficients.

Cepstral mean normalization was applied to deal with the constant channel assumption.

2.2. Acoustic Model

Hidden Markov Models (HMMs) with 64 mixtures Gaussian probability density function for acoustic modeling is used. Diagonal covariances Gaussians are used. All units are modeled by 3 left-to-right states with skip transition. 47 Russian context-free phonemes with pause unit are chosen.

The pronunciation dictionary was created automatically from word orthography using a set of Russian context sensitive rewrite rules.

The Viterbi algorithm is used for the pattern matching.

2.3. Baseline Performance

All acoustic models were trained by using 12000 utterances from 2037 words dictionary. Training set consists of 2000 isolated words and 1000 continuous phrases. All utterances were collected by the single speaker. The recognition experiments are carried out on the PC P-IV 2.4GHz.

1000 utterances from the same speaker for isolated word recognition were collected. The word recognition accuracy and average time for different vocabulary sizes are given in table 1. Recognition time per one second of speech is linear from vocabulary size and may be evaluated as 2300 sec for 1987K vocabulary.

Table 1: Baseline recognition performance for isolated word

Vocabulary size	1K	15K	95K
Accuracy, %	99.9	97.9	94.7
Time, sec	1	16	115

Additional 1000 utterances with numbers from 0 up to 999 for continuous speech recognition were collected. The word recognition accuracy and average time for different vocabulary sizes are given in table 2.

Table 2: Continuous speech performance

Vocabulary size	1K	15K	95K
Accuracy, %	98.0	96.5	92.6
Time, sec	2.1	29	205

3. ELVIRS Algorithm for Isolated Words

The ELVIRS algorithm is described in [2]. Building blocks like features extraction, acoustic models reused from baseline system. Common pattern matching with subset of vocabulary is used on the second pass also. Changes are concentrated in the new first recognition pass when phonetic

transcriber is applied to make the sequence of phonemes. Then information retrieval procedure builds the sub-vocabulary for second pass.

3.1. Phoneme-by-Phoneme Recognizer

The phonetic transcribing algorithm [3] builds a phonetic sequence for speech signal regardless to the dictionary. For this purpose it was constructed some phoneme generative automata which can synthesize all possible continuous speech model signals for any phoneme sequence. Then the phoneme-by-phoneme recognition of unknown speech signal is applied.
The same context-free phonemes as in baseline recognition system are used.
The experimental accuracy of finding phoneme at the right place equals to approximately 85%.

3.2. Sub-Vocabulary Retrieval Procedure

Preliminary transcription dictionary is prepared to build the phoneme triple - transcription index. The index entry key is a phoneme triple. So the index consists of M^3 entries where M is the number of phonemes in the system. Each entry contains the list of transcriptions that include key phoneme triple. Additional memory usage is approximately 50 MB for vocabulary with 1 M words.

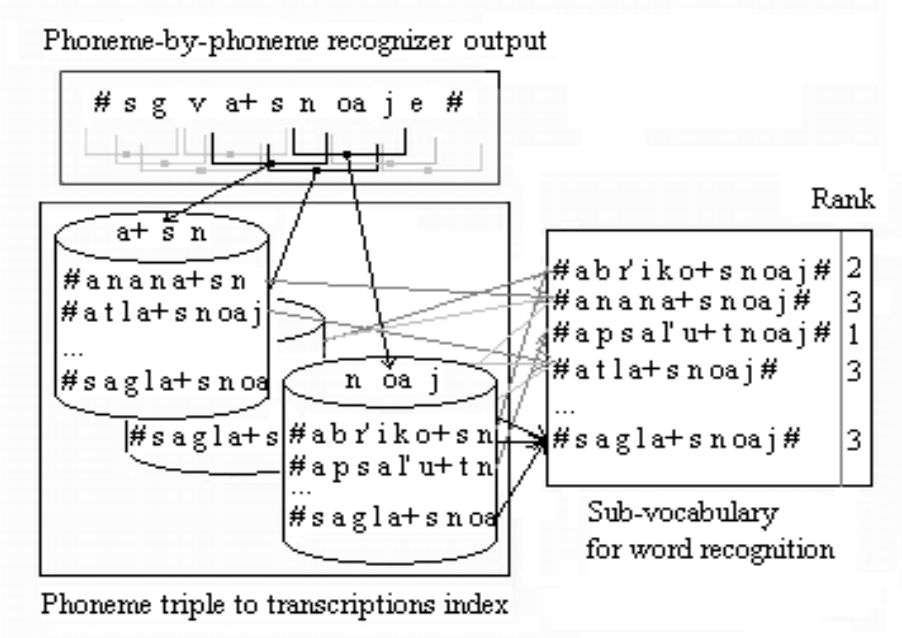

Fig. 2. Sub-vocabulary retrieval process

Sub-vocabulary retrieval process is illustrated in figure 1. Phoneme-by-phoneme recognizer output is split into overlapping phoneme triples. Resulting phoneme triple becomes the query. Now, in this system the simple query is used where phoneme triple and query are the same. In the future it should be modified to take into account the insertion, deletion and substitution of phoneme sequence by using Levensteine dissimilarity. Thus phonetic sequence produces the query flow for database.
The one query's answer consists in a list of transcriptions in which the given triple is included. Next queries produce new transcription's portions to be copied into the sub-vocabulary for the second pass. The counter for word repetition is supported to make the rank of word.
All transcriptions in resulting sub-vocabulary are arranged according to the word rank (repetition counter). First N transcriptions are copied into a final sub-vocabulary for the second pass recognition. So the recognition sub-vocabulary consists of highest ranks of transcriptions but its size does not exceed a fixed limit N.

4. The Information Consideration

The phoneme-by-phoneme recognizer output can be considered as a correct phoneme sequence passed through a noisy channel and converted into an output sequence. Denote a right phoneme in output sequence as 1 and wrong one as 0. Let 1 occurs with probability u. The probability P to find k and more successive 1 in a binary set with length of n can be computed with the help of the following recurrent expression:

$$P_n = \begin{cases} 0, n < k \\ u^k, n = k \\ P_{n-1} + u^k (1-u)(1-P_{n-k-1}), n > k \end{cases}$$

Probabilities P to find three and more successive 1 in a binary sequence for different lengths n and probabilities u are shown in table 3. Average transcriptions length is equal to approximately 8 and the accuracy of finding phoneme at the right place for known utterance is equal to approximately 85%. For these values the probability to find right word in chosen sub-vocabulary is equal 0.953.

Table 3: Probability to find three and more successive 1 in a binary sequence with length of n

u / *n*	0.75	0.8	0.85	0.9
6	0.738	0.819	0.890	0.948
7	0.799	0.869	0.926	0.967
8	0.849	0.908	**0.953**	0.982
9	0.887	0.937	0.971	0.991
10	0.915	0.956	0.981	0.995

5 ELVIRCOS Algorithm for Continuous Speech

5.1 Architecture

Additional procedure word graph composition is applied after transcriptions list retrieval procedure. It produces the words network for second pass recognition.

5.2 Word graph composition

Word graph composition procedure is illustrated in figure 2. Words network starts from vertex S and finishes into vertex F. Each triple from phoneme output burns intermediate vertexes with numbers synchronous the occurrence time. From other hand each triple became query to data base index, which returns the transcription list as result. Transcriptions are interlaced with intermediate numbered vertexes as base vertexes so that burning phoneme triples are placed in coordination.

The rank of transcription is increased in case when intersection between same transcriptions burned from different phoneme triple occurs. For each moment of time (synchronous with phoneme sequence) it may be calculate the number of involved transcriptions.

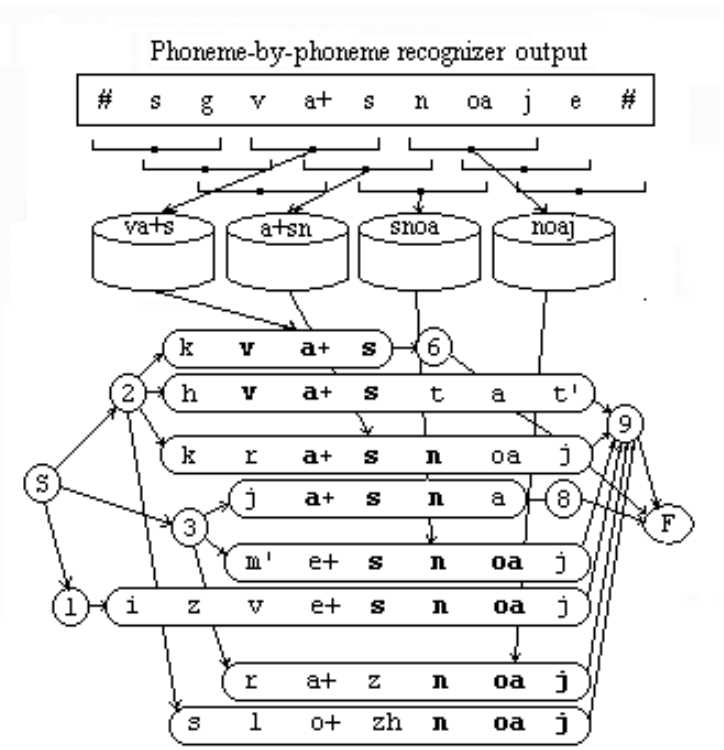

Fig 2*:* Word graph composition for continuous speech

To reduce the word graph complexity the fixed limit N is applied. For each moment of time transcriptions with small ranks are removed from word graph so that N transcriptions are remain.

Due to the word graph is composed from left to right it is possible to construct one in real time with the delay is equal of largest transcription length.

5.3. ELVIRCOS Algorithm Overview

The ELVIRCOS algorithm works as follows.

1. Prepare the vocabulary for recognition.
2. Chose the phoneme set and build transcriptions for words from vocabulary by rules.
3. Create database index from phoneme triple to transcriptions.
4. Train the acoustic models from collected speech signals.

Recognition stage:

1. Apply phoneme-by-phoneme recognizer to the input speech signal to produce the phoneme sequence.
2. Split phoneme sequence into overlapping phoneme triples.
3. Make queries from phoneme triples.
4. Retrieve transcriptions lists by queries from database index.
5. Compose word graph network.
6. Recognize the input speech signal with composed word net.

6 Experimental Results

Some modifications of HTK toolkit are made to take into account the first pass of algorithm and experiments are executed.

In case of isolated word the influence of the sub-vocabulary limit N on the average recognition time and word recognition accuracy for different vocabulary sizes was investigated (tab. 4). The useful of limit N is shown for large vocabularies.

The considerable reduction of the recognition time with relatively small accuracy degradation (approximately 5%) in comparison with baseline system has been achieved. The accuracy degradation has a good agreement with the information consideration.

Table 4: ELVIRS recognition performance

Vocabulary size / Subvocab. limit N	15K		95K		1987K	
	Acc, %	Time, sec.	Acc, %	Time, sec	Acc, %	Time, sec
200	94.6	1.6	87.6	2.1	76.0	1.9
500	**95.5**	**1.9**	**90.1**	**2.5**	80.0	3.3
1000	96.0	2.1	90.7	3.1	82.7	4.4
2000	96.0	4.4	92.0	4.5	**84.8**	**6.8**
5000	96.0	4.6	92.9	8.3	86.4	12.0

For continuous speech preliminary experiments were executed in case when limit N is equal the vocabulary size. Results for different vocabulary sizes are shown in table 5.

Table 5: ELVIRCOS recognition performance

Vocabulary size	15K	95K	1987K
Accuracy, %	85.3	84.1	80.4
Time, sec	4.5	22	442

Average time reduction is not so considerable as in isolated word recognition because some phoneme sequences produce word graphs with very large sizes. It seems that introducing the limit N decrees essentially the recognition time.

7. Conclusions

This paper has presented the new approach for large vocabulary speech recognition and a preliminary experimental evaluation. The importance of information retrieval for speech recognition should be underlined. It was shown that additional information source from analysis of phoneme sequence allows to restrict the search space. These new restrictions lead to speech recognition with vocabularies cover practically all words for given language.

References

1. *S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev and P. Woodland*, "The HTK Book", Cambridge University Engineering Department, 2002.

2. *Valeriy Pylypenko*. "Information Retrieval Based Algorithm for Extra Large Vocabulary Speech Recognition", Proc. of the 11th International Conference "Speech and Computer", SPECOM'2006, St. Petersburg, Russia, 2006.

3. *Taras K. Vintsiuk*. "Generalized Automatic Phonetic Transcribing of Speech Signals", Proc. of the 5th All-Ukrainian Conference "Signal/Image Processing and Pattern Recognition", *pp. 95–98,* Kyiv, Ukraine, 2000, in Ukrainian.