LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION OF ICSI MEETING DATA

Martin KARAFIÁT, Doctoral Degree Programme (3)
Dept. of Graphics and Multimedia, FIT, BUT
E-mail: karafiat@fit.vutbr.cz

Supervised by: Dr. Jan Černocký

ABSTRACT

This paper describes a large vocabulary continuous speech recognition system for meeting data based mostly on HTK toolkit from Cambridge University. Main advantages and drawbacks in basic approaches of decoding, phone modeling are discussed. The system is designed with concentration on decreasing computational power demand without decreasing the accuracy. Combination of the stack decoder from Duisburg university and common HTK Viterbi decoder is used for this purpose.

All experiments were performed on the ICSI meeting database. Reached accuracy of recognition was bigger than 50% in 25% of pure HTK decoding time.

1 INTRODUCTION

In the field of speech-to-text translations, Large Vocabulary Continuous Speech Recognition (LVCSR) is very challenging task. In spite of growing power of computation aids, these systems still consume a lot of time due to searching in big space of hypotheses (all sentences which can be created from words in dictionary) [1].

Most of the current speech recognition systems are based on continuous Hidden Markov Models (HMMs) [4]. Word models are usually created by concatenation of smaller units, mostly phoneme models or their context dependent variants (so called triphones). Triphones can be split into the two main classes. Word internal triphones where only context dependencies inside of the words are considered and cross-word ones which are looking across word boundaries.

Time synchronous (Viterbi) decoding or Best First (Stack decoding) scheme can be used for finding the most probable word sequence which will represent an unknown speech signal. Both systems have advantages and drawbacks affecting the recognition time and accuracy. Our goal is to build a system combining these different approaches and minimizing their drawbacks.

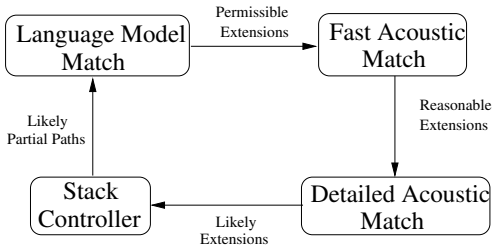

Figure 1: Basic principles of a stack decoder.

2 SEARCHING ALGORITHMS

Time synchronous decoding optimizes the best state sequence. All hypotheses (or a beam of hypotheses) are advanced frame by frame without considering their likelihoods. This decoding works on the state-level and transition between words are remembered.

The main characteristics are as follows:

- Implementation of algorithm is relatively simple.
- Decoding with cross-word dependent phonemes could be easy performed.
- A lot of computation is wasted on computing less accurate hypotheses.
- Memory and time grow almost linearly with number of required hypotheses in lattice (word networks) generation [1].

A **stack decoder** is an implementation of the *Best first* search [1]. It works on the word-level and considers only the best word sequence. Hypotheses are sorted in stack according to their partial likelihoods. The most likely hypothesis is taken off the stack and updated by one word extensions.

Space of all possible word extensions is limited by fast language and acoustic matches. Detailed acoustic match is applied and hypothesis is expanded by a list of likely extensions. New hypotheses are evaluated, sorted and pushed back into the stack [1]. This process is briefly drafted in fig. 1.

The main characteristics are:

- It is easy to implement complex language dependencies due to the expansion of hypotheses on the word level.
- It offers more types of pruning methods due to higher complexity of decoding algorithm [2].
- Generating lattices can be realized efficiently with a little computational overhead. All hypotheses on the stack can be directly linked into the lattice [2].
- Problems with the implementation of cross-word dependency because acoustic likelihoods of the new extensions depend on successive words which are not known.
- Algorithms to compare hypotheses of different length are required [1].

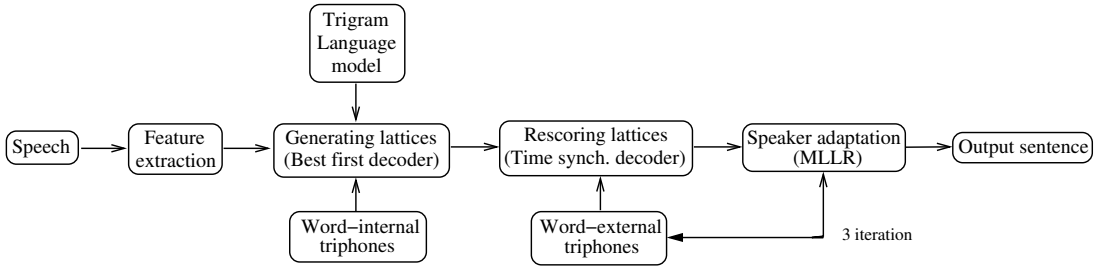

Figure 2: Recognition system.

3 RECOGNITION SYSTEM

In order to exploit the advantages of time synchronous and best first search described in section 2, we use a decoding scheme that incorporates both approaches in a two stage process. The recognition system is illustrated in figure 2.

The first stage uses faster best first decoding approach to generate a set of lattices. Since cross-word context dependent phoneme decoding is hard to implement in a best first search, this stage uses the word internal context dependent models. It significantly reduces the number of possible hypotheses.

The second stage takes the generated lattices and rescores them using a time synchronous decoder and cross-word context dependent phoneme models. Reduced number of hypotheses means that the search space is restricted, therefore the decoding is faster.

Important condition for good performance is a sufficiently broad search space given by lattices and good acoustic models in the rescoring part. Consequently, for further experiments we tried to change the size of lattices by adjusting the pruning in their generation.

Next we implemented a speaker adaptation of acoustic models. This process was performed iteratively with using decoder result from previous iteration. It was done for each speaker separately. This approach does not increase the demand for computation power significantly because only models in the rescoring (faster) part of the system were adapted.

4 EXPERIMENTS

4.1 DATABASES AND TOOLS

Acoustic models were trained on 39.4 hours selected in the ICSI meetings database [7]. Data for training language model was taken from training part of ICSI database - 53099 sentences and Switchboard database - 248581 sentences.

The system was tested on one hour from the chosen 3 meetings of ICSI meetings database [7]. Results are reported in terms of word recognition accuracy. The dictionary was created by merging the ICSI meetings dictionary with Switchboard dictionary - 36136 words.

The recognition system used a combination of packages: the HTK toolkit [4] for a parametrization of input speech, training and working with acoustic models and time

synchronous decoding; the Du-coder [5] - a stack decoder; and the SRILM toolkit [6] - a language model training tool. All experiments were running on P4 2.4GHz.

4.2 EXPERIMENTS

The "cheap" method with using of all kinds of pruning in best first decoding was used for the first experiments with lattice generation in (Tab. 1). First line contains the base line result of clean decoding without lattice generation and adaptation method for next comparison. Time of lattice generation in second line is slightly higher than the clean decoding as was expected in section 2. Further rescoring this lattices by cross-word triphones brings increase of accuracy more than 3% (see lines 1 and 3 in Tab. 1). It means, that the recognition results with using cross-word triphones could be generated in less than one day.

Line	Kind of system	Acc. [%]	Time [hour]
1	Without rescoring and adaptation	42.17	9.4
2	Lattice generation	-	9.8
3	Rescoring, without adaptation	45.68	9.8 + 2.5
4	Resc., adapt. per speaker, 1 iter.	47.18	9.8 + 4.65
5	Resc., adapt. per speaker, 2 iter.	47.21	9.8 + 9.3
6	Resc., adapt. per speaker, 3 iter.	**47.24**	9.8 + 14

Table 1: Lattice generation by trigram LM and word internal triphones with using all kind of pruning and rescoring with cross-word triphones.

Lattices with suppression of a pruning effect were generated for next experiment, only with basic beam search pruning (Tab. 2). Time of lattices generation was over 163 hours longer than previous system (see line 3 in Tab. 2 and Tab. 1) because more hypotheses were considered during the decoding.

This lattice complexity brings accuracy improvement more than 3% (see last lines in Tab. 2 and Tab. 1) with increasing of rescoring time about 150 hour.

Line	Kind of system	Acc. [%]	Time [hour]
1	Without rescoring and adaptation	44.8	165
2	Lattice generation	-	173
3	Rescoring, without adaptation	47.66	173 + 6.32
4	Resc., adapt. per speaker, 1 iter.	50.38	173 + 12
5	Resc., adapt. per speaker, 2 iter.	50.61	173 + 17.5
6	Resc., adapt. per speaker, 3 iter.	**50.73**	173 + 23

Table 2: Lattice generation by trigram LM and word internal triphones with using only beam search pruning and rescoring with cross-word triphones.

5 CONCLUSION AND DISCUSSION

This paper presented an LVCSR system based on combination of two decoders - the first (fast) one is used for limitation of search space and generates word lattices, the second (slower) one uses more powerful acoustic and language models.

Using the system with less pruning, we reached almost 51% accuracy which is a good result for this type of task (spontaneous meeting data). The decoding requires approximately 200 times real-time, which is still about 25% of the time required by the standard HTK time-synchronous decoder.

With more several pruning, we are able to limit the recognition time to only about 25 times real-time while loosing only 3% of accuracy. This might be useful for applications, where the speed of processing and availability of computer resources are limiting factors.

ACKNOWLEDGEMENT

This research was supported by EC project Multi-modal meeting manager (M4), No. IST-2001-34485 and partially by Grant Agency of Czech Republic under project No. 102/02/0124.

Jan Černocký was supported by post-doctoral grant of Grant Agency of Czech Republic No. GA102/02/D108.

REFERENCES

[1] J. J. Odell, *The Use of Context in Large Vocabulary Speech Recognition*. PhD thesis, Cambridge University Engineering Department, 1995.

[2] M. Schuster, *On supervised learning from sequential data with applications for speech recognition*. PhD thesis, Nara Institute of Science and Technology, 1999.

[3] M. Schuster, "Nozomi - a fast, memory-efficient stack decoder," in *5th International Conference on Spoken Language Processsing (ICSLP)*, (Sydney), pp. 1835–1838, 1998.

[4] S. Young, J. Jansen, J. Odell, D. Ollason, and P. Woodland, *The HTK book*. Cambridge, UK: Entropics Cambridge Research Lab., 2002.

[5] D. Willett, C. Neukirchen, and G. Rigol, "DUCODER-the duisburg university LVSCR stackdecoder," in *Proc. IEEE Int. Conf. on Acoustic, Speech and Signal processing (ICASSP)*, (Istanbul), 2000.

[6] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *Proc. International Conference on Spoken Language Processing (ICSLP)*, (Denver), pp. 901–904, 2002.

[7] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters.(2003), "The ICSI meeting corpus," in *Proc. IEEE Int. Conf. on Acoustic, Speech and Signal processing (ICASSP)*, (Hong Kong), 2003.