Overview of Japanese Dictation Toolkit
– 1999 version – *

Tatsuya Kawahara[†], Akinobu Lee (Kyoto Univ.), Tetsunori Kobayashi (Waseda Univ.),
Kazuya Takeda (Nagoya Univ.), Nobuaki Minematsu (Univ. Tokyo), Shigeki Sagayama (JAIST)
Katsunobu Itou (ETL), Akinori Ito (Yamagata Univ.), Mikio Yamamoto (Univ. Tsukuba),
Atsushi Yamada (ASTEM), Takehito Utsuro (Toyohashi Univ. Tech.),
Kiyohiro Shikano (NAIST)

http://winnie.kuis.kyoto-u.ac.jp/dictation/
E-mail: dictation-tk-request@astem.or.jp

May 8, 2000

Abstract

A sharable software repository for Japanese LVCSR (Large Vocabulary Continuous Speech Recognition) is introduced. It is designed as a baseline platform for research and developed by researchers of different academic institutes under the governmental support. The repository consists of a recognition engine, Japanese acoustic models and Japanese statistical language models as well as Japanese morphological analysis tools. We set up a variety of Japanese phone HMMs from a monophone to a triphone model of thousands of states. They are trained with ASJ (Acoustical Society of Japan) databases. A lexicon and word N-gram (2-gram and 3-gram) models are constructed with a corpus of Mainichi newspaper. A recognition engine Julius is developed for evaluation of both acoustic and language models. These modules can be easily integrated and replaced under a plug-and-play framework, which makes it possible to fairly evaluate components and to develop specific application systems. Assessment of these modules and systems in a 20000-word dictation task, which was also set up in our project, is reported. The software repository is freely available to the public.

1 Introduction

Large Vocabulary Continuous Speech Recognition (LVCSR) is a basis of speech technology applications including a voice-input word processor and automatic transcription of broadcast programs or personal audio tapes. Its component technologies can also be used in various applications such as spoken dialogue interfaces.

In order to build an LVCSR system, high-accuracy acoustic models, large-scale language models and an efficient recognition program (decoder) are essential[1][2]. Integration of these components and adaptation techniques for real-world environment are also needed. On the other hand, most of researchers are interested in specific components and try to demonstrate the effectiveness of a new method by integrating with other components. This background motivated us to develop a free sharable platform that can be used as a baseline and reference. Such a platform will also suffice for a baseline in developing application systems and for an open entry to those of other research fields or foreign countries.

It is rather easy to have agreement of a common interface and format in the LVCSR system. It realizes a plug-and-play framework for research and development. Namely, researchers can put and test a new component and system developers can replace and tune components for specific applications.

We adopted Mainichi Newspaper, one of the nationwide general newspapers in Japan, for the sharable corpus of both text and speech[3], and organized a project to develop a standard software repository that includes a recognition program together with acoustic and language models[4][5]. The three-year project (1997-2000), funded by the IPA (Information-technology Promotion Agency), Japan, was a collaboration of researchers of different academic institutes. An overview of the corpus and software is depicted in Figure 1.

*Work is sponsored by the IPA (Information-technology Promotion Agency), Japan.

†School of Informatics, Kyoto University 606-8501, Japan.

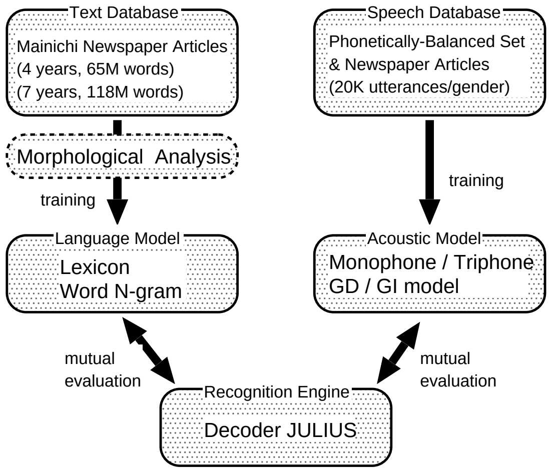

Figure 1: Platform of LVCSR

Table 1: List of Acoustic Models

model	#state	#mixture	gender
monophone	129	4, 8, 16	GD, GI
triphone 1000	1000	4, 8, 16	GD
triphone 2000	2000	4, 8, 16	GD, GI
triphone 3000	3000	4, 8, 16	GD
PTM triphone	3000/129	64	GD, GI

GD: Gender Dependent, GI: Gender Independent

Table 2: List of Japanese Phones

`a i u e o a: i: u: e: o: N w y`
`p py t k ky b by d dy g gy ts ch`
`m my n ny h hy f s sh z j r ry`
`q sp silB silE`

Specifications of the acoustic models, language models and recognition engine as well as Japanese morphological analysis tools are described in this paper. We also report evaluation of these modules under a 20000-word Japanese dictation task.

2 Specification of Models and Programs

2.1 Acoustic Model

Acoustic models are based on continuous density HMM. We basically adopt the HTK format[6] as it is an ASCII file.

We have trained several kinds of Japanese acoustic models from a context independent phone model to triphone models, as listed in Table 1. We set up both gender dependent and gender independent models. A PTM (Phonetic Tied-Mixture) model is a hybrid of the monophone and ordinary triphone, in that a mixture of Gaussian distributions is shared as in the monophone, but different weights of distributions are assigned to states of triphone contexts. Users can choose an adequate model according to the purpose. A simpler model realizes faster recognition at the expense of accuracy degradation.

The set of 43 Japanese phones are listed in Table 2. The phone notation is defined by Acoustical Society of Japan (ASJ) committee on speech database. Here, symbols `a:`∼`o:` stand for long vowels and symbol `q` for a double consonant. Three pause models, `silB`, `silE` and `sp`, are introduced for pauses at the beginning, at the end of utterances and between words, respectively. The pause models are always context independent, though `sp` model can be a context for other phones.

The acoustic models are trained with ASJ speech databases of phonetically balanced sentences (ASJ-PB) and newspaper article texts (ASJ-JNAS). In total, around 20K sentences uttered by 132 speakers are available for each gender.

The speech data were sampled at 16kHz and 16bit. Twelfth-order mel-frequency cepstral coefficients (MFCC) are computed every 10ms. Temporal difference of the coefficients (Δ MFCC) and power (Δ LogPow) are also incorporated. So the pattern vector at each frame consists of 25 (=12+12+1) variables. Cepstral mean normalization (CMN) is performed on every utterance to offset the channel mis-match.

Every phone model consists of three states excluding the initial and final states that have no distributions. The state transitions are all left-to-right, and the path from the initial state and that to the final state are limited to one.

When a triphone model is applied to continuous speech recognition, it has to cover all possible combinations of the phones that may appear in a lexicon or cross-word context. Thus, an extra file is used to define the mapping from the possible tuples (logical triphone; 21384) to the prepared models (physical triphone; 7946). In actual, there are not sufficient data for training all the triphone models. So the decision tree-based clustering is performed to build a state-tying structure that groups similar contexts and can be trained with reasonable data. By changing the threshold of clustering, we set up a variety of models whose number of the states is 1000, 2000 and 3000, respectively.

A PTM model further introduces mixture-tying. It is made up of a set of 64-mixture distributions of the monophone model (129 states in total) and a set of 3000 states defined by the triphone model. Each state of the triphone shares mixture distributions of the corresponding monophone state and has different weights to synthesize context-dependent acoustic patterns. These parameters are re-trained for optimization. Thus, the PTM model realizes an efficient triphone representation and reliable parameter estimation[7].

2.2 Morphological Analysis and Lexicon

A lexicon is a set of lexical entries specified with their notations and baseforms. It is also in the HTK format[6]. It is consistent with both acoustic model and language model. Phone symbols used in baseforms are covered with the acoustic model. For every lexical entry, its probability is given by the language model (at least 1-gram). Such words that are defined in a lexicon but not referred in a language model are collectively treated as unknown word (`UNK`) category in our decoder.

In Japanese, definition of vocabulary depends on morphological analysis system that segments undelimited texts. We adopt a morphological analyzer ChaSen that has been developed at Nara Institute of Science and Technology. Major modification has been made for speech recognition purpose.

To define lexical entries for speech recognition, the morphological analyzer has to not only segment texts into words but also perform Kanji-to-Kana (similar to grapheme-to-phoneme) transcription. The Kana transcription for dictionary words is changed from orthographic one to phonemic one. We basically follow the NHK Japanese pronunciation rules and adopt Katakana transcription, which is automatically transformed into phone symbols. For example, "*Tokyo*" is transcribed as `[t o: ky o:]` instead of `[t o u ky o u]`.

In addition, we have developed a postprocessor that handles irregular variations of pronunciation. Especially in Japanese, pronunciation of digits is dependent on adjacent words, and pronunciation of counter words is also affected by preceding numericals. For example, `[i q + p u N]` for one minute, `[n i + f u N]` for two minutes, and `[s a N + p u N]` for three minutes. These are handled by a post-processor ChaWan. Number expressions are segmented into digits and counters.

In Japanese, there are many morpheme entries that have multiple part-of-speech tags and also a lot of Kanji (Chinese character) entries that have multiple pronunciations. Generally, words of different part-of-speech tags have different tendency of possible adjacent words, even if they are same in notation. Pronunciation of some words is also dependent on adjacent words as in the above examples. In order to improve language modeling, we distinguish lexical entries by not only their notations but also their part-of-speech tags and Katakana transcriptions. Not a few words have multiple transcriptions which are not disambiguated by the morphological analysis. In such a case, one entry is allotted for the lexicon and the language model, and multiple baseforms are registered according to the transcriptions.

The vocabulary consists of the most frequent words (=morphemes) in Mainichi newspaper articles from January 1991 to September 1994 (45 months)[3]. The lexicon also includes entries of comma, period and question marks that are re-written as a pause in pronunciation. Lexical coverage of various vocabulary sizes is listed in Table 3. Lexicons of 5K, 20K and 60K vocabulary size are available. Coverage of 99% is achieved with the 60K lexicon. [1]

Table 3: Lexical Coverage

vocabulary size	coverage
5000	88.3%
20000	96.4%
24000	97.0%
53000	99.0%
60000	99.2%
101000	99.7%
154000	99.9%

2.3 Language Model

N-gram language models are constructed based on the lexicon. Specifically, word 2-gram and 3-gram models are trained using back-off smoothing. Witten-Bell discounting method is used to compute back-off coefficients. We adopt the CMU-Cambridge SLM toolkit format[8] as it is also an ASCII file.

The comma, period and question marks are also included in the statistical language models. As a result, the occurrence of short pauses between words is estimated by the probabilities of these symbols that correspond to pauses.

The cut-off threshold for baseline N-gram entries is 1 for both 2-gram and 3-gram [cutoff-1-1]. Then, elimination of N-gram entries is explored for memory efficiency. Conventionally, it has been done by setting a higher cut-off threshold. Here, we prepare a model with the cut-off threshold of 4 [cutoff-4-4]. In addition, we have introduced a new method based on the model entropy, not word occurrences[9]. The method incrementally picks out 3-gram entries so that ML estimation of the reduced model gives the smallest increase of entropy. As a result, 3-gram entries are reduced to 1/10 [compress10%].

[1] 60K lexicon is generated from newspaper articles of 75 months.

Table 4: List of 20K Language Models

	2-gram entries	3-gram entries
45month cutoff-1-1	1,238,929	4,733,916
45month cutoff-4-4	657,759	1,593,020
45month compress10%	1,238,929	473,176
75month cutoff-1-1	1,675,803	7,445,209
75month cutoff-4-4	901,475	2,629,605
75month compress10%	1,675,803	744,438

Table 5: List of 60K Language Models

	2-gram entries	3-gram entries
75month cutoff-1-1	2,420,231	8,368,507
75month compress10%	2,420,231	836,852

We have used Mainichi newspaper corpus to train the language model. Headlines and tables were removed in pre-processing. For 20K language model, we first used the training corpus of 45-month articles (01/91-09/94; 65M words), which was also used to define the lexicon. Then, training data was increased to 75-month articles (01/91-09/94, 01/95-06/97; 118M words). As for 60K model, both lexicon and language model are set up with 75-month articles, and only entropy-based reduction is performed. The list of language models are given in Table 4 and Table 5.

In our decoder, each entry occupies 18 bytes for 2-gram and 6 bytes for 3-gram. For the decoder that performs forward-backward search, the backward 3-gram model is prepared. For 60K model, the ordinary forward 3-gram model is also available.

2.4 Decoder

A recognition engine named Julius[10] has been developed to interface the acoustic and language models. It can deal with various types of the models, thus can be used for their evaluation.

It accepts not only wave files (16bit PCM) and acoustic parameter files (HTK format) but also microphone input (Sun/SGI workstation, Linux PC, via DAT-LINK/netaudio). Speech analysis is implemented only for those parameters adopted by the acoustic model of the toolkit.

Julius performs a two-pass (forward-backward) search using word 2-gram and 3-gram on the respective passes.

In the first pass, a tree-structured lexicon assigned with language model probabilities is applied with the frame-synchronous beam search algorithm. It assigns pre-computed 1-gram factoring values to the intermediate nodes, and applies 2-gram probabilities at the word-end nodes. Cross-word context dependency is handled with approximation which applies the best model for the best history [-iwcd1 option].

We assume one-best approximation rather than word-pair approximation. The degradation by the rough approximation in the first pass is recovered by the tree-trellis search in the second pass. The word-trellis index form is adopted to efficiently look up predicted word candidates and their scores. The word-trellis index is a set of survived word-end nodes in the first pass, their scores and their corresponding starting frames.

In the second pass, 3-gram language model and accurate sentence-dependent acoustic model is applied for rescoring. There is an option that applies cross-word context dependent model to word-end phones without delay for accurate decoding. We enhanced the stack-decoding search by setting a maximum number of hypotheses of every sentence length since the simple best-first search sometimes fails to get any recognition results. The search is not A*-admissible because the second pass may give better scores than the first pass. It means that the first output candidate may not be the best one. Thus, we compute 10 candidates by continuing the search and sort them for the final output.

The parameters of language model weight and insertion penalty as well as the beam width can be adjusted for the respective passes. Two default decoding options are also set up for each type of the acoustic models: Standard decoding strictly handles context dependency for accurate recognition. Efficient decoding uses a smaller beam width and terminates the search when the first candidate is obtained.

For efficient decoding with the PTM model that has a large mixture per state, Gaussian pruning is implemented. It prunes Gaussian distance (=log likelihood) computation halfway on the full vector dimension if it is not promising. Using the already computed k-best values as a threshold guarantees us to find the optimal ones but does not eliminate computation so much [safe pruning]. We implement more aggressive pruning methods by setting up a beam width in the intermediate dimensions [beam pruning] or using heuristic estimation of the yet-to-be-computed dimensions [heuristic pruning]. We perform safe pruning in the standard decoding and beam pruning in the efficient decoding.

An overview of the decoder is given in Table 6.

3 Japanese Dictation System

By integrating the modules specified in the previous section, a Japanese dictation system is realized.

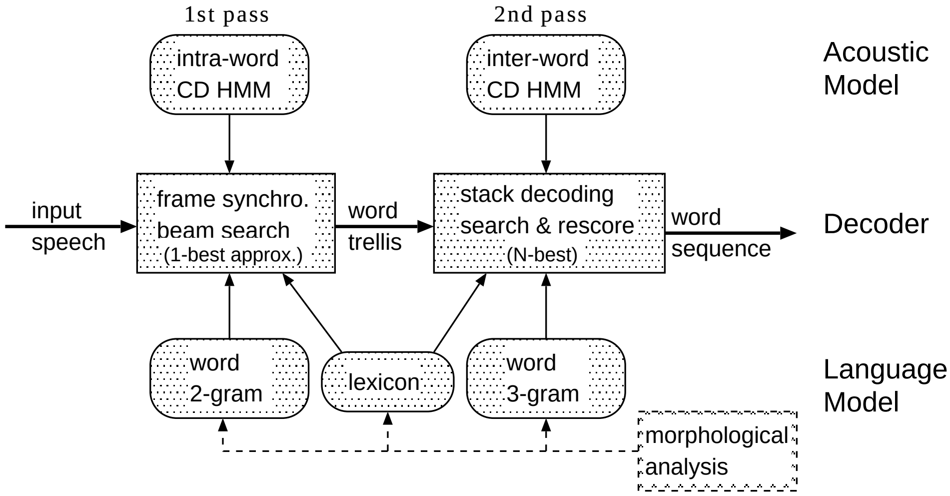

Figure 2: Block Diagram of Japanese Dictation System

Table 6: Overview of Decoder Julius

	cross-word phone model	language model	search approx.
1st pass	approximate	2-gram	1-best
2nd pass	accurate	3-gram	N-best

A block digram of the system is illustrated in Figure 2. The acoustic model and the language model are integrated based on the decoder specification. In the first pass, word 2-gram is applied and phonetic context dependency (CD) is correctly handled only within a word. Word 3-gram and inter-word context dependent model, which are more precise and computationally expensive, are incorporated in the second pass to re-score the reduced candidates.

Since there are several variations in both acoustic model and language model, we can design different systems accordingly. Setting of decoding options and parameters such as the beam width may also yield variations of the system.

A 20000-word and 60000-word dictation system has been developed. The components independently developed at different sites were successfully integrated.

4 Evaluation of Modules and Systems

The integrated system can be used to evaluate the component modules, in turn. By changing the modules under the plug-and-play framework, we can evaluate their effects with respect to the recognition accuracy and efficiency. Most of experiments are done using a 20K dictation task.

As IPA-98-Testset, [2] we have used a portion of the ASJ-JNAS speech database that were not used for training of the acoustic model. It consists of 100 samples by 23 speakers for each gender.

The sample sentences are from articles of 10/94-12/94, which are open to the language model training. The distributions of sentence length and perplexity are also taken into account. [3] The total number of words in the set is 1575, and 0.44% of them are out-of-vocabulary words by the 20K lexicon.

Word accuracy is used as the evaluation criterion. Definition of word accuracy in Japanese involves several issues. First, the unit of words depends on morphological analysis. Use of character-based accuracy is more reasonable in this sense. Yet, we use the word accuracy based on our morphological analyzer ChaSen for simplicity. Second, even if we fix the morphological analyzer, word segmentation is still non-deterministic. For example, a word "*tomoni*" may often be broken into two words "*tomo*" and "*ni*". Thus, we concatenate possible compound words before matching with answer sequences. Third, use of Kanji and Kana characters is also arbitrary for notation of many words. One solution is to transform all Kanji characters to Kana before matching. But it accepts confusion of different words of same pronunciation. Here, matching is done in Kanji-basis. These processes are automated by our tool to compute the word accuracy.

[2] www.milab.is.tsukuba.ac.jp/jnas/test-set/male/male1_LARGE.txt
www.milab.is.tsukuba.ac.jp/jnas/test-set/female/female1_LARGE.txt

[3] NORMAL:76 + LONG:24, LPP:26 + MPP:45 + HPP:29

Table 7: Evaluation of Acoustic Model (male; accuracy)

	mix.4	mix.8	mix.16
GD monophone	75.3	79.6	83.9
GI monophone	68.3	78.0	81.7
GD triphone 2000	92.0	92.6	94.3
GI triphone 2000	89.3	91.8	92.5
GD PTM 129x64 (3000)	92.4		
GI PTM 129x64 (3000)	89.5		

Table 8: Evaluation of Acoustic Model (female; accuracy)

	mix.4	mix.8	mix.16
GD monophone	75.5	80.7	88.9
GI monophone	76.0	80.8	84.7
GD triphone 2000	92.0	94.4	95.2
GI triphone 2000	92.3	93.4	94.8
GD PTM 129x64 (3000)	94.6		
GI PTM 129x64 (3000)	94.3		

4.1 Evaluation of Acoustic Models

At first, we present evaluation of a variety of acoustic models. Here, the baseline language model [75-month cutoff-1-1] and the standard decoding is adopted. Safe pruning is performed in the PTM model.

The word accuracy is listed in Table 7 for male and Table 8 for female speakers, respectively. The PTM model achieves a comparable accuracy to that of the triphone model with a significantly smaller number of parameters. In fact, recognition with the PTM model is faster by twice than that with the triphone. It is also observed that gender independent models increase the error rates to a certain extent compared with gender dependent models.

4.2 Evaluation of Language Models

Next, we present evaluation of language models. The male triphone 2000x16 model and the standard decoding is used. We have compared the baseline models [cutoff-1-1] and the reduced models [compress10%] in 20K and 60K lexicons.

The memory size and the word accuracy are shown in Table 9 for each language model. Accuracy degradation is not observed by enlarging the lexicon from 20K to 60K, though recognition time increased by 30%. As for memory-efficient models, the entropy-based compression method almost keeps the accuracy with eliminating 3-gram entries to 1/10.

Table 9: Evaluation of Language Model

	accuracy	LM size
20K 75month cutoff-1-1	94.3	79MB
20K 75month compress10%	94.3	38MB
60K 75month cutoff-1-1	93.7	100MB
60K 75month compress10%	93.5	55MB

4.3 Evaluation of Decoder

The decoding algorithms are evaluated by using the male samples and the baseline language model [75-month cutoff-1-1].

The effect of cross-word context dependency handling for accurate decoding with the triphone model is shown in Table 10. The accuracy in the 1st pass and final result is listed. It is confirmed that enhancement of the first pass drastically improves its accuracy. Together with the enhancement of the second pass, the final error rate is reduced by 25%. It turns out that the search errors are reduced to less than half.

The effect of Gaussian pruning for efficient decoding with the PTM model is shown in Table 11. The accuracy and relative computation amount of Gaussian distances are listed. In the preliminary experiment, it is found out that computing only two out of 64 mixture distributions per state does not cause any loss of accuracy. As the comparison of several pruning methods, the beam pruning method is the best and reduces the computation to about 1/5.

Table 10: Evaluation of Accurate Decoding Algorithms (triphone model)

GD triphone 2000x16	accuracy final (1st pass)
(1998 version)	92.0 (78.9)
enhanced IW-CD: 1st pass	93.0 (85.2)
enhanced IW-CD: 1st&2nd pass	94.3 (85.0)

Table 11: Evaluation of Efficient Decoding Algorithms (PTM model)

GD PTM 129x64	accuracy	Gaussian computation
No Gaussian pruning	92.8	100
safe pruning	92.4	52
heuristic pruning	90.7	36
beam pruning	91.2	21

4.4 System Assessment

Performance of the total dictation system is summarized in Table 12 for 20K system and Table 13 for 60K system, where two typical configurations are listed: efficiency-oriented and accuracy-oriented.

The accurate version adopts the triphone model and the standard decoding to achieve a word accuracy of 95%. The efficient version using the PTM model keeps the accuracy above 90% and runs almost in real-time at a standard PC. It also uses the compressed language model for memory efficiency.

The 20K system is compared to its preceding 1998 version. Both efficient and accurate versions reduce the error rate to almost 2/3 in spite of the increase of recognition time.

5 Conclusion

Key property of the software toolkit is generality and portability. As the formats and interfaces of the modules are widely acceptable, any modules can be easily replaced. Thus, the toolkit is suitable for research on individual component techniques as well as development of specific systems. Actually, the experiments in this paper are done by integrating and replacing modules that are developed at different sites. The results prove that the plug-and-play framework effectively works and our platform demonstrates reasonable performance when adequately integrated. The software repository also includes morphological analysis tools, which are useful in applying the system to various task domains.

The current version of the software (decoder) works under standard Unix platform. We are planing to port it to Windows and support standard API.

The software repository is freely available to the public.

Acknowledgement: The authors are grateful to advisory members of the project for their comments and cooperation.

References

[1] S.J.Young. A review of large-vocabulary continuous-speech recognition. *IEEE Signal Processing magazine*, 13(5):45–57, 1996.

[2] T.Matsuoka, K.Ohtsuki, T.Mori, S.Furui, and K.Shirai. Japanese large-vocabulary continuous-speech recognition using a business-newspaper corpus. In *Proc. ICSLP*, pages 22–25, 1996.

[3] K.Itou, M.Yamamoto, K.Takeda, T.Takezawa, T.Matsuoka, T.Kobayashi, K.Shikano, and S.Itahashi. The design of the newspaper-based Japanese large vocabulary continuous speech recognition corpus. In *Proc. ICSLP*, pages 3261–3264, 1998.

[4] T.Kawahara, T.Kobayashi, K.Takeda, N.Minematsu, K.Itou, M.Yamamoto, T.Utsuro, and K.Shikano. Sharable software repository for Japanese large vocabulary continuous speech recognition. In *Proc. ICSLP*, pages 3257–3260, 1998.

[5] T.Kawahara, T.Kobayashi, K.Takeda, N.Minematsu, K.Itou, M.Yamamoto, A.Yamada, T.Utsuro, and K.Shikano. Japanese dictation toolkit – plug-and-play framework for speech recognition R&D –. In *Proc. IEEE workshop on Automatic Speech Recognition and Understanding*, 1999.

[6] S.Young, J.Jansen, and J.Odell D.Ollason P.Woodland. *The HTK BOOK*, 1995.

[7] A.Lee, T.Kawahara, K.Takeda, and K.Shikano. A new phonetic tied-mixture model for efficient decoding. In *Proc. IEEE-ICASSP*, 2000.

[8] *The CMU-Cambridge Statistical Language Modeling Toolkit v2*, 1997.

[9] N.Yodo, K.Shikano, and S.Nakamura. Compression algorithm of trigram language models based on maximum likelihood estimation. In *Proc. ICSLP*, pages 1683–1686, 1998.

[10] A.Lee, T.Kawahara, and S.Doshita. An efficient two-pass search algorithm using word trellis index. In *Proc. ICSLP*, pages 1831–1834, 1998.

Table 12: Specification of 20K Systems (also comparison with 1998)

	efficient version		accurate version	
	1998	1999	1998	1999
acoustic model	monophone 16 (0.5MB)	PTM 129x64 (3.0MB)	triphone 2000x16 (8.6MB)	
language model	75month compress10% (38.0MB)		75month cutoff-1-1 (78.5MB)	
decoding	fast	fast	(1998 ver.)	standard
CPU time	1.1x RT	2.3x RT	8.4x RT	12.8x RT
Acc,Corr(male)	82.6, 83.5c	89.1, 91.1c	92.0, 93.2c	94.3, 95.4c
Acc,Corr(female)	85.7, 87.1c	91.8, 93.1c	93.2, 94.1c	95.2, 96.2c
Acc,Corr(GD ave)	84.2, 85.3c	90.5, 92.1c	92.6, 93.7c	94.8, 95.8c
Acc,Corr(GI)	81.5, 84.0c	89.7, 91.1c	90.3, 91.7c	93.7, 94.7c

RT (Real Time): 5.8sec./sample, CPU: Ultra SPARC 300MHz

Table 13: Specification of 60K Systems

	efficient version	accurate version
acoustic model	PTM 129x64 (3.0MB)	triphone 2000x16 (8.6MB)
language model	75 compress10% (54.5MB)	75 cutoff-1-1 (99.7MB)
decoding	fast	standard
CPU time	2.9x RT	16.9x RT
Acc,Corr(male)	89.1, 90.9c	93.7, 94.6c
Acc,Corr(female)	91.6, 92.7c	93.4, 94.9c
Acc,Corr(GD ave)	90.4, 91.8c	93.6, 94.8c
Acc,Corr(GI)	88.9, 90.5c	93.2, 94.2c

RT (Real Time): 5.8sec./sample, CPU: Ultra SPARC 300MHz