FREE SOFTWARE TOOLKIT FOR JAPANESE LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION *

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

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

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 a governmental support. The repository consists of a recognition engine (Julius), Japanese acoustic models and statistical language models as well as Japanese morphological analysis tools. 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 is reported. The software repository is freely available to the public.

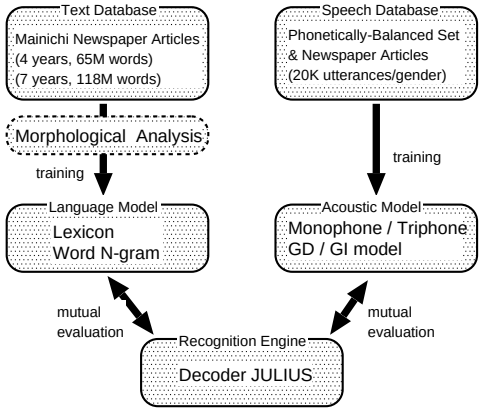

Figure 1: Platform of LVCSR

1. INTRODUCTION

Large Vocabulary Continuous Speech Recognition (LVCSR) is a basis of various speech technology applications. In order to build an LVCSR system, high-accuracy acoustic models, large-scale language models and an efficient recognition program (decoder) are essential. 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. 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[1], and organized a project to develop a standard software repository that includes a recognition program together with acoustic and language models[2]. An overview of the corpus and software is depicted in Figure 1.

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 MODULES

2.1. Acoustic Model

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

*WORK WAS SPONSORED BY THE IPA (INFORMATION-TECHNOLOGY PROMOTION AGENCY), JAPAN.

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

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

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 synthesis 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.

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. Cepstral mean normalization (CMN) is performed on every utterance.

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[3].

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.

In Japanese, definition of vocabulary depends on morphological analysis system that segments undelimited texts. We adopt a morphological analyzer ChaSen. 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. In addition, we have developed a postprocessor that handles irregular variations of pronunciation.

Table 2: Lexical Coverage

vocabulary size	coverage
5000	88.3%
20000	96.4%
60000	99.2%

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. In order to improve language modeling, we distinguish lexical entries by not only their notations but also their part-of-speech tags and Kana transcriptions. When a word has multiple transcriptions which are not disambiguated by the morphological analysis, one entry is allotted with multiple baseforms.

The vocabulary consists of the most frequent words (=morphemes) in Mainichi newspaper articles from January 1991 to September 1994 (45 months)[1]. Available lexicons are listed in Table 2. Coverage of 99% is achieved with the 60K lexicon.

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 as it is also an ASCII file. 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[4]. 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%].

We have used Mainichi newspaper corpus to train the language model. Headlines and tables were removed in pre-processing. 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). The list of language models are given in Table 3 and Table 4.

Table 3: 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 4: 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

2.4. Decoder

A recognition engine named Julius[5] 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.

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. 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[5]. In the second pass, 3-gram language model and accurate sentence-dependent acoustic model are applied for re-scoring. 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.

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[3].

An overview of the decoder is given in Table 5.

Table 5: 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

3. EVALUATION AT DICTATION TASK

By integrating the modules specified in the previous section, a Japanese dictation system is realized. 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, [1] 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 open to the language model training. Word accuracy is computed using our tool that processes compound words.

3.1. 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] is adopted. Safe pruning is performed in the PTM model.

The word accuracy is listed in Table 6 for male and Table 7 for female speakers, respectively. The PTM model achieves a comparable accuracy to that of the triphone model of much larger 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.

3.2. Evaluation of Language Models

Next, we present evaluation of language models. The male triphone 2000x16 model is used.

First, we investigated the effect of language model reduction. The memory size and the word accuracy are shown in Table 8 for various 20K models. [2] The models trained with 75-month articles consistently achieve better accuracy than those with 45-month data. As for memory-efficient models, the entropy-based compression method [compress-10%] is more effective than the simple cut-off method [cutoff-4-4].

Next, the effect of vocabulary size is examined. The performance of 20K models and 60K models for the same

[1] winnie.kuis.kyoto-u.ac.jp/pub/julius/result99/

[2] Only this experiment was carried out using the old version of the decoder, so the accuracy is worse than in other Tables.

test set is compared in Table 9. Accuracy degradation is not observed by enlarging the lexicon from 20K to 60K, though recognition time increased by 30%. The entropy-based compression method almost keeps the accuracy with eliminating 3-gram entries to 1/10.

3.3. Evaluation of Decoder

The decoding algorithms are evaluated by using the acoustic model of male triphone 2000x16 and the baseline language model.

Improvement by several enhancements is summarized in Table 10. The accuracy in the 1st pass and final result is listed. First, setting a beam in the stack decoding is effective. Then, the effect of cross-word context dependency handling for accurate decoding is confirmed. Enhancement of the first pass drastically improves its accuracy. Together with 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.

4. 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 is freely available to the public.

Table 6: 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 7: 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	

Table 8: Comparison of Language Model Reduction Methods (20K)

	accuracy	LM size
20K 45-month cutoff-1-1	89.8	54MB
20K 45-month cutoff-4-4	89.3	23MB
20K 45-month compress-10%	89.3	28MB
20K 75-month cutoff-1-1	92.0	79MB
20K 75-month cutoff-4-4	90.9	34MB
20K 75-month compress-10%	91.8	38MB

using old decoder (Julius rev.2.0)

Table 9: Evaluation of Language Model (20K and 60K)

	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

Table 10: Improvement of Decoding Algorithms

	word accuracy final (1st pass)
baseline (almost equivalent to [2])	91.2 (78.9)
+ enhanced stack decoding (=rev.2.0)	92.0 (78.9)
+ enhanced XW-CD: 1st pass	93.0 (85.2)
+ enhanced XW-CD: 2nd pass (=rev.3.0)	94.3 (85.0)

XW-CD: Cross-Word Context Dependency handling

References

[1] 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.

[2] 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.

[3] A.Lee, T.Kawahara, K.Takeda, and K.Shikano. A new phonetic tied-mixture model for efficient decoding. In *Proc. IEEE-ICASSP*, pages 1269–1272, 2000.

[4] 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.

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