RECENT PROGRESS OF OPEN-SOURCE LVCSR ENGINE JULIUS AND JAPANESE MODEL REPOSITORY – SOFTWARE OF CONTINUOUS SPEECH RECOGNITION CONSORTIUM –

*Tatsuya Kawahara** *Akinobu Lee*† *Kazuya Takeda*‡ *Katsunobu Itou*‡ *Kiyohiro Shikano*†

* Kyoto University, ACCMS, Kyoto 606-8501, Japan
† Nara Institute of Science and Technology, Takayama, 630-0192, Japan
‡ Nagoya University, School of Information Science, Nagoya 464-8603, Japan
E-mail: csrc@astem.or.jp

ABSTRACT

Continuous Speech Recognition Consortium (CSRC) was founded for further enhancement of Japanese Dictation Toolkit that had been developed by the support of a Japanese agency. Overview of its product software is reported in this paper. The open-source LVCSR (large vocabulary continuous speech recognition) engine Julius has been improved both in performance and functionality, and it is also ported to Microsoft Windows in compliance with SAPI (Speech API). The software is now used for not a few languages and plenty of applications. For plug-and-play speech recognition in various applications, we have also compiled a repository of acoustic and language models for Japanese. Especially, the set of acoustic models realizes wider coverage of user generations and speech-input environments.

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 to real-world environments 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. Standardized interface and format realize 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. Moreover, open-source property of the software provides huge flexibility in research and development.

The three year project (1997-2000) of developing Japanese Dictation Toolkit[1] [1] was so successful that the software package has been widely used in research community in Japan. The fact drove us to start Continuous Speech Recognition Consortium (CSRC), [2] which anyone can join to make contribution (either financially or as a voluntary workforce) to improvement of the software repository. Actually, we had around 70 members from both industry and universities.

The purpose of the consortium activities was summarized as follows:

1. Maintenance of the software as a research platform

   New techniques have been continuously incorporated to keep the software to the up-to-date and state-of-the-art level.

2. Wider coverage of real-world environments

   A variety of acoustic models are added to the repository to provide wider coverage on both user generations and input environments.

3. Portability to various applications

   The core speech recognition engine Julius [3] has been ported to support SAPI (Speech API) as well as languages other than Japanese.

Specifications of the latest version of the software repository are described in this paper.

2. LVCSR ENGINE Julius

Julius [2, 10] is a core engine of our open-source LVCSR toolkit. It is a two-pass decoder which applies a tree lexicon and word bigram model in the first pass, and then cross-word triphone model and word trigram model in the second

[1] http://www.ar.media.kyoto-u.ac.jp/dictation/
[2] http://www.lang.astem.or.jp/CSRC/
[3] http://julius.sourceforge.jp/

pass, for efficiency both in processing time and memory. Typically, it can realize real-time dictation of 20K-60K vocabulary with accuracy of about 90% on current PCs. The most prominent feature is its portability so that the decoder can be combined with a variety of acoustic and language models as it is carefully designed to be independent from them. Basically, it accepts most of acoustic models in the HTK format (that are trained with HTK) [4] as well as language models in the ARPA format that are generated by CMU-Cambridge SLM toolkit [5] or palmkit. [6]

The feature has made Julius a language-independent engine; it is reported to operate for not a few languages other than Japanese with satisfactory performance[3].

In the past years, tremendous amount of enhancements have been done to improve the performance and functionality. The major features are described in the following sub-sections. We have been also engaged in porting Julius to Microsoft Windows in compliance with SAPI (Speech API). [7]

2.1. Supporting Rule-based Grammars (Julian)

While the original Julius could handle only N-gram (trigram) language model, most of speech applications still use hand-crafted deterministic grammars. Therefore, we extended the decoder so that it can handle rule-based grammars. Although the generated engine for this purpose is referred to as Julian, most of the user options and source codes are shared with Julius .

Julian users first have to define a set of word categories, which is similarly used in class N-gram modeling. The mapping of word categories and actual lexical entries is defined in a vocabulary file. A grammar should be specified with a set of rewriting rules starting from a whole sentence into a sequence of word categories. Then, the grammar and vocabulary files are pre-compiled into a finite state automaton, which is used by the decoder. They can also be converted to the XML format adopted in SAPI. Tools for compiling, checking and converting grammars are included in the package.

The feature has made the software used in a greater number of applications such as a conversational robot and interfaces for home appliances.

Julius can handle multiple grammars and switch them, for example, depending on the dialogue state or the item in a form to be filled. Selective activation of grammars is also implemented via SAPI.

[4] http://htk.eng.cam.ac.uk/

[5] http://mi.eng.cam.ac.uk/~prc14/toolkit.html

[6] http://palmkit.sourceforge.net

[7] http://www.microsoft.com/speech

2.2. Supporting Class N-gram

Julius is also extended to handle a class N-gram language model, which can be regarded as an intermediate of a statistical model and a rule-based grammar when we define classes corresponding to word categories mentioned above. It is known to realize robust language modeling even with sparse training data.

The occurrence probability of words within a class is specified in the word dictionary file together with the baseform entry.

2.3. Confidence Measure Output

A confidence measure of speech recognition is necessary for back-end systems, for example, dialogue systems to control confirmation strategies or summarization systems to make compression of the text. It should be given to every word as well as the whole utterance, and normalized to range between 0 and 1 like a posterior probability.

We have devised a novel algorithm to efficiently compute the confidence measure during the stack decoding search[4].

2.4. SAPI-Compliant Version

Julius for SAPI version is also available at the Web site[10]. Most of the user options including acoustic, language models and decoding parameters can be specified through the dialog box of "Speech (Recognition)" property in "Control Panel" of Windows XP. We have also confirmed that this SAPI-compliant version works with SALT (Speech Application Language Tags)[5]. [8]

3. ACOUSTIC MODEL REPOSITORY

In the former Japanese Dictation Toolkit, we trained a baseline acoustic model using two speech databases collected under ASJ (Acoustical Society of Japan): phonetically balanced sentences (ASJ-PB) and newspaper article texts (ASJ-JNAS). They provide around 40K sentences uttered by 264 speakers. In this consortium, we made use of an even larger speech database of phonetically-balanced sentences collected at ATR (ATR/BLA)[6]. Moreover, we have added a variety of acoustic models to the repository in order to cover wider user generations including children and elderly people and typical real-world applications such as telephony and in-car interfaces.

All acoustic models are gender-independent continuous-density HMM in the HTK format for portability and compatibility with the Julius system. For most of the models, moreover, information of regression trees is attached to Gaussian components so that they can be

[8] http://www.saltforum.org

Table 1. Specification of acoustic model

sampling	16kHz & 16bit
framing	25ms long & 10ms shift
acoustic feature	12 MFCC + 12 ΔMFCC + ΔLogPow
CMN	done for every utterance
HMM structure	3 states with output probabilities without state-skipping transitions
Gaussian	diagonal covariance

adapted using the MLLR framework of HTK. This feature improves their applicability to real-world environments.

The specification of acoustic analysis, which is summarized in Table 1, and the Japanese phone set are same as those of Japanese Dictation Toolkit[7].

3.1. Basic (Adult) Acoustic Model

With incorporation of the ATR/BLA database, 260-hour speech uttered by 4130 speakers is available in total. With this training data, we constructed several kinds of acoustic models.

For efficient decoding, we have particularly adopted PTM (Phonetic Tied-Mixture) modeling[8], which is a synthesis of the monophone and ordinary triphone, in that a mixture of Gaussian distributions is shared as in the monophone, but different mixture weights are assigned to states of triphone contexts. Here, we regard a set (mixture) of Gaussians defined for each monophone state as a codebook. As there are 43 phones each having 3 states, the number of codebooks is 129.

Evaluation of the acoustic models was done using two test-sets. One is IPA-98-testset selected from ASJ-JNAS read speech corpus, which has been widely used for evaluation of Japanese LVCSR[1]. The other is the test-set (male only) from ATR dialogue speech databases (ATR/SDB & ATR/SLDB), which are different from that used for training (ATR/BLA) and were used for development of the speech translation system at ATR.

The results are summarized in Table 2. In read speech data, the accuracy is mostly determined by the model complexity or net number of Gaussian distributions, but the PTM model realizes efficient decoding. It operates actually faster by around twice than the ordinary triphone of same complexity because Gaussian pruning is more effective for a larger mixture[8]. In dialog speech which is not matched to the training data, the PTM model is more robust and achieves better performance. These accuracies are the best or close to the best figures reported for the test-sets.

Table 2. Evaluation of adult models (Word Accuracy)

	PTM	PTM	PTM	triphone	triphone
#state	3000	3000	3000	2000	5000
#codebook	129	129	129	-	-
mixture size	64	128	256	16	64
#Gaussian	8K	16K	32K	32K	320K
read speech	92.1	93.0	93.6	93.3	95.4
dialogue	83.1	84.5	84.8	82.4	83.7

Table 3. Evaluation on different user generations (Word Accuracy)

	basic model	elderly model	children model	mixed model
adult speech	93.0	-	-	91.5
elderly speech	82.1	86.7	-	86.6
children's speech	54.1	-	75.8	77.1

3.2. Model for Elderly Speech

Although speech interfaces will be useful for elderly people, a dedicated acoustic model is necessary to cope with such speakers. Therefore, we (NAIST group) compiled a speech database of 301 senior speakers (60 to 90 years old), and trained a PTM triphone model of 64 mixture components.

3.3. Model for Children's Speech

A dedicated acoustic model is also necessary for children's speech. We (Nagoya-U group) collected a speech database of 400 elementary school students (6 to 12 years old). Since it is not easy to collect long sentence utterances by children, the data size is not sufficient enough to cover triphone contexts. Therefore, we conducted MAP adaptation training from an adult female model, which is a PTM triphone model of 64 mixture components.

3.4. Mixed Model for All Generations

Moreover, we trained a universal model for all generations (children, adult and elderly speakers) by using all training data mentioned above. It is a PTM triphone model with a mixture (=codebook) size of 128. The recognition accuracy for test-sets of different generations by each model is summarized in Table 3. It is observed that mis-match of the model causes significant degradation, and the mixed model realizes comparable performance to the matched model.

3.5. Model for Telephone Speech

For telephone speech recognition purpose, we (Kyoto-U group) collected a speech database (8kHz sampling) of 517 speakers, each uttering 50 sentences via a fixed or mobile

phone. Then, we trained a shared-state triphone model of 2000 states and 16 mixture components.

3.6. Model for In-Car Speech

For advanced speech interfaces in mobile vehicles such as telematics applications, the CIAIR project of Nagoya Univ.[9] collected huge databases of speech and dialogue while driving a car. Using the database of 22000 sentences by 700 speakers, we trained a triphone model of 2000 states and 32 mixture components. The average SNR is about 20dB and speech analysis is performed with high-pass filtering above 250Hz.

4. LANGUAGE MODEL UPDATE

In Japanese Dictation Toolkit, we trained a baseline language model using the text database of Mainichi newspaper articles of 1991-1997. Since the database is increasing and vocabulary is changing year by year, we have updated the lexicon and language model.

In Japanese, definition of vocabulary depends on the morphological analysis system that segments un-delimited texts. We adopt a morphological analyzer ChaSen and define a set of lexical entries by considering part-of-speech tags and baseform entries; multiple entries are defined for a word according to these tags.

4.1. Newspaper Language Model

The current Mainichi newspaper corpus provides articles of 12 years and texts of 350M words. By deriving the most frequent words from the corpus, a lexicon of 60K entries is defined. Then, we trained a trigram language model.

The model was in the ARPA format and also converted to the binary format accepted by the Julius system.

4.2. Web Language Model

Another or even richer source of text data is World Wide Web. It is also expected that Web texts contain more spoken language expressions. Thus, we (Nagoya-U group) collected texts of 2.7G words, and trained another lexicon and language model with a vocabulary size of 60K.

5. CONCLUSION

Key property of the software repository is generality and portability. As the formats and interfaces of the modules are widely acceptable, any module can be easily replaced. Thus, the toolkit is suitable for research on individual component techniques as well as development of specific systems. Moreover, open-source property of the software provides huge flexibility in research and development.

Actually, the toolkit has been widely used in research community, especially in Japan. Since Julius supported rule-based grammars and was ported to Microsoft Windows in compliance with SAPI, it has also been used by researchers and engineers outside the speech community for various kinds of applications such as multi-modal interfaces and conversational robots. Moreover, Julius is also being ported to many languages other than Japanese, which verifies the portability.

The core engine Julius is freely downloadable[10] although the repository of acoustic models can be obtained by contacting the consortium.

Acknowledgments: Development of the software was supported by Continuous Speech Recognition Consortium (CSRC) under SIG-SLP of IPSJ. We are grateful to those members, especially committee members who volunteered for its management. The repository includes software tools contributed from Prof. A. Ito (Tohoku Univ.), Dr. H. Banno (Wakayama Univ.), Dr. T. Yamada (Tsukuba Univ.), Dr. T. Nishiura (Ritsumeikan Univ.), Mr. M. Mimura and Dr. A. Yamada (ASTEM).

REFERENCES

[1] T.Kawahara, A.Lee, T.Kobayashi, K.Takeda, N.Minematsu, S.Sagayama, K.Itou, A.Ito, M.Yamamoto, A.Yamada, T.Utsuro, and K.Shikano. Free software toolkit for Japanese large vocabulary continuous speech recognition. In *Proc. ICSLP*, volume 4, pages 476–479, 2000.

[2] A.Lee, T.Kawahara, and K.Shikano. Julius – an open source real-time large vocabulary recognition engine. In *Proc. EUROSPEECH*, pages 1691–1694, 2001.

[3] T.Rotovnik, M.S.Maucec, B.Horvat, and Z.Kacic. A comparison of HTK, ISIP and Julius in Slovenian large vocabulary continuous speech recognition. In *Proc. ICSLP*, pages 681–684, 2002.

[4] A.Lee, K.Shikano, and T.Kawahara. Real-time word confidence scoring using local posterior probabilities on tree trellis search. In *Proc. ICASSP*, volume 1, pages 793–796, 2004.

[5] K.Wang. SALT: a spoken language interface for Web-based multimodal dialog systems. In *Proc. ICSLP*, pages 2241–2244, 2002.

[6] T.Takezawa, T.Morimoto, and Y.Sagisaka. Speech and language databases for speech translation research in ATR. In *Proc. Oriental-COCOSDA workshop*, pages 148–155, 1998.

[7] 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.

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

[9] F.Itakura. Multi-media data collection for in-car speech communication. In *Proc. Workshop Hands-Free Speech Communication*, 2001.

[10] `http://julius.sourceforge.jp/`