Recent Advance of Thai Open-Vocabulary Automatic Speech Recognition

Chai Wutiwiwatchai[†a)], Vataya Chunwijitra[†], Sila Chunwijitra[†], Phuttapong Sertsi[†], Sawit Kasuriya[†], Patcharika Chootrakool[†], Kwanchiva Thangthai[†], Chanchai Junlouchai[†], Kamthorn Krairaksa[†]

National Electronics and Computer Technology Center, National Science and Technology Development Agency, Thailand

***Abstract*—We describe the recent development of the NECTEC Thai open-vocabulary automatic speech recognition system. Some of the techniques that were found beneficial over its baseline system are: hybrid word-subword language modeling to enhance the vocabulary coverage in a constraint resource; multi-conditioned noisy acoustic modeling to improve the system robustness and spoken-style language model interpolation using a newly developed large social media speech database; recent state-of-the-art speech features; and lastly, online decoding, speech compression, and Docker-based distributed computing to reduce the processing and data transmission time. These techniques result in a 29.0% word error rate on open-vocabulary noisy speech test sets which is 42.5% relatively lower than the baseline system. The overall system operates at nearly 1.2xRT which is promising for real applications.**

***Index Terms*— open-vocabulary, speech recognition, Thai language**

I. Introduction

Large vocabulary continuous speech recognition (LVCSR), automatic speech recognition (ASR) for natural speech with a large lexicon, has gained significant advance both in the research and implementation aspects in the past few years. Several well-known IT companies as well as research institutes have shown their systems running for open-vocabulary speech input. Two recent breakthroughs causing the shift of the technology are *big data*, the benefit of very large scale data obtained directly from social usage; and *deep learning*, a newly discovered learning machine capable to capture the high variation of the data. The IBM 2015 system [1] was presented to achieve 23% word recognition error rate on a natural conversational task over telephones. The system took several sophisticated neural-network based algorithms as well as huge 2,000-hours training data into account. Google English Voice Search [2] reported its performance of less than 20% word recognition error rate with 230 billion words language modeling data and more than 5,000 hours acoustic modeling data. The RWTH LVCSR system presented recently its performance on different languages including Polish, Portuguese, English, and Arabic. Significant improvements were obtained using many modern proposed techniques which made the recognition error rate from over 30% down to less than 20% [3]. These reports convincingly express the applicability of the technology in the near future.

Research on Thai LVCSR in National Electronics and Computer Technology Center (NECTEC) has been conducted since 2003 with a series of publications on Thai continuous speech corpora [4, 5, 6, 7]. A report on the system development utilizing the LOTUS corpus showed the first baseline performance at 24.4% word error rate (WER) on a quiet environment, read speech test set [8]. NECTEC has joined Universal Speech Translation Advance Research (USTAR) consortium [9] since 2007 and published a report on Thai ASR for a network-based speech translation service in travel and sport domains [10].

Research and development aiming at open-vocabulary, i.e. ASR with unlimited vocabulary, has just been focused in NECTEC since 2012. Key improvements have been reported consecutively on both the algorithm for reducing out-of-vocabulary (OOV) words and the system architecture suited for service implementation [11, 12, 13]. This paper summarizes the key improvements so far integrated in the 2015 NECTEC open-vocabulary ASR system. Comparative experiments over a baseline system along the past years regarding important issues we found on building open-vocabulary ASR and engineering the system are given.

This paper is organized as follows. The next section describes our baseline system built around the year 2012-2013. Section III presents improvement issues: hybrid word-subword language modeling, robust acoustic modeling, spoken-style language modeling, and run-time system design, respectively. Section IV shows experiments, Section V discusses on existing problems and future work, and concludes this paper.

[†] The authors are with National Electronics and Computer Technology Center, National Science and Technology Development Agency, 112 Phahon-Yothin Rd., Klong-Luang, Pathumthani, 12120, Thailand.
a) chai.wutiwiwatchai@nectec.or.th

II. Baseline System Development

a. *Structure of ASR*

Our current Thai ASR system has been developed from an open-source KALDI toolkit [14], which is based on weighted finite-state transducer (WFST). Speech parameters given by the feature extraction module are input to the decoding module which takes into account three major components, an acoustic model, a language model, and a pronunciation lexicon. To recognize a speech input, the decoding module constructs a word graph whose word links are tagged with their corresponding language model probabilities. Each word in the graph is expanded to phones according to the given pronunciation lexicon. Each phone refers to its corresponding acoustic model. The speech input travels into the word graph producing potential word paths with top cumulative probabilities. A word recognition result is finally the word path having the highest cumulative probability. In the KALDI toolkit, the acoustic model, the language model, and the pronunciation lexicon are all crafted as WFSTs. An additional context-dependent phone WFST is needed for building a context-dependent acoustic model. All these WFSTs are composited in prior to form a single big WFST used in decoding.

To achieve open-vocabulary ASR, the pronunciation lexicon as well as the language model have to largely cover words used in the language. While the larger the lexicon, the lower the recognition performance, a major problem of making the recognition vocabulary opened becomes how to trade off among the lexicon size and the recognition accuracy. Open-vocabulary ASR also implies the system capability to accept a variety of speech input from different situations, equipments, and environments. Therefore, the system robustness is also another important issue to solve. Last but not least in our open-vocabulary ASR work, an overall system response time has also been taken into account as we aim finally to make the system usable in real applications.

b. *Development Resources*

Table I summarizes resources used to develop our baseline Thai open-vocabulary ASR system. The variety of training corpora insists the openness of acceptable speech input over speakers, speaking styles and domains, microphone equipments, and environments. The baseline system utilized in total 224 hours of speech for acoustic modeling and 66.74 million-words text for language modeling.

c. *Building the Baseline System*

In acoustic modeling, conventional 13-order Mel-frequency cepstral coefficients (MFCC), their derivatives and second derivatives were extracted from all the speech data presented in the Table I. Using the KALDI toolkit, the speech features were used to train context-dependent triphone Hidden Markov Models (HMM) covering 75 phones in Thai [4] plus a silence. N-gram language models with Chen and Goodman's modified Kneser-Ney discounting were constructed from the overall text data presented in the Table I using the SRILM toolkit [15]. The number of unique words appeared in the training text was more than 140,000 which, for the baseline system, was simply included in the system pronunciation lexicon. As mentioned earlier in the KALDI platform, all these system components were constructed as WFST and recognition can thus use the WFST composition operation. Incorporate the 4-gram language model could be achieved by language model rescoring. The baseline system run-time architecture was simply designed. The ASR server stores input speech in a buffer and starts recognition when the input is completely received. Decoding starts after speech features are extracted from the overall speech. And the output text is returned to the client after the recognition process ends.

TABLE I
Summary of resources used to develop the baseline ASR system.

Component	Tool	Corpus	Detail
Acoustic model	KALDI	LOTUS	48 speakers, 55 hours of article read speech
		LOTUS-BN	147 hours of broadcast news speech
		USTAR	22 hours of the USTAR speech translation application over smart phone data channels
Language model	SRILM	BEST	7.17 million words from 12 domains
		Thai BTEC	0.83 million words in travel domain
		Thai HIT	0.60 million words in sport domain
		PANTIP	57.32 million words from 8 weblog domains
		LOTUS-BN	0.83 million words of broadcast news

III. Key Improvement Issues

Evaluations of the baseline system described above have shown limitations in many issues. In the past three years, many solutions to improve the system have been experimented. This section expresses four key issues we have attacked.

a. *Hybrid Word-Subword Language Modeling*

One of the most important issue to open-vocabulary ASR is the vocabulary coverage. Similar to other languages, new words in Thai have always been invented. Some of them are proper names, person names, and words in social networks. It is hence almost impossible to include all possible words in the system lexicon. Subword unit is one commonly used technique when modeling out-of-vocabulary (OOV) words in many languages as multiple subword units can be combined to form a new word which is not seen before in the training data. Morpheme, a smallest meaningful unit in a language, becomes a natural choice for subword unit especially for highly inflected languages [16, 17]. In Thai, *pseudo-morpheme* (PM), a syllable-like unit in a written form, has been proposed as a subword unit. The definition of the PM is a syllable and if any syllable cannot be represented by a bounded chunk of written text, that PM will span to cope with the least number of syllables whose written text can be bounded. Table II shows examples of Thai words and their corresponding PM segments. According to Thai writing rules, PM is more deterministic when compared with word and has been shown to help mitigate the word segmentation problem [18]. Given a word or a string of text, PMs can be determined quite accurately with an automatic syllable segmentation tool [19].

TABLE II
Samples of Thai words and their corresponding PM units.

Text	Meaning	Word	PM	Pronunciation
กระทรวงวัฒนธรรม	Ministry of culture	กระทรวง|วัฒนธรรม	กระ|ทรวง|วัฒน|ธรรม	k r a | s u:a ng | w a t th a n a | th a m
ราชวงศ์จักรี	Chakkri dynasty	ราชวงศ์|จักรี	ราช|วงศ์|จักรี	r a: t ch a | w o ng | c a k kr i:

Nevertheless, in recognition, small units usually suffer from acoustic confusability and also cover shorter span of context in an n-gram language model. To resolve these problems, a hybrid language model which combines both unit types, PM and word, has been proposed [11]. In training, the text data were firstly segmented into words. Frequently appeared words were kept in the text and stored in a system lexicon, and the rest words were further segmented into PMs. Only frequently occurred PMs were kept in the lexicon and the rest PMs in the text were marked as unknown units. Then the mixed-unit training text were used to train the N-gram model as usual. While the most frequently used words could be covered by the lexicon, unseen words could also been modeled by sequences of PM units. By this way, we can fully manage the size of the lexicon while keeping the OOV rate minimal.

b. *Robust Acoustic Modeling*

Although the acoustic model in our baseline system has been built from speech corpora from various speaking environments, its performance against noisy speech is still low. A major cause is that we have not yet directly taken the noisy speech data into training. Besides the data, more advanced speech features and training algorithms have been proposed and have not yet integrated in our system.

Instead of using noise-added speech data, a new speech corpus, LOTUS-SOC [7], has been developed to tackle this problem. The corpus was designed to cover quiet rooms and other 6 noisy environments including cafeteria, busy streets, cars or buses, sky or subway trains, shopping malls, and fast-food restaurants. Nearly 200 speakers were requested to naturally utter the scripts selected from Twitter to mimic the spoken style. Recording was done via a smart phone application created specifically for corpus collection. This database gains an approximately 8.24 SNR ratio in average. An improved acoustic model was constructed by multi-conditioned training, which carefully mixed noisy and clean speech training data.

Many advanced speech features have been proposed recently. In our improved system, MFCCs of a focused speech frame and its 3 surrounding frames, 91 coefficients in total, were collected. Linear Discriminant Analysis (LDA) [20] was applied to reduce the features to 40. Maximum Likelihood Linear Transformation (MLLT) [21] was then used to de-correlate among the 40-order coefficients. This technique (LDA-MLLT) has been widely used and also available in the KALDI toolkit. Discriminative training based on Maximum Mutual Information (MMI) [22] or Maximum Phone Error (MPE) [23] has been a state-of-the-art training algorithm recently as its ability to discriminate ambiguous phones often mistaken in the baseline training algorithm. These discriminative training methods have also been comparatively tested in our system.

c. *Spoken-style Language Modeling*

Naturally, ASR has been used for spoken-style speech input and, hence, building ASR to cope with a spoken-style language model is one of major challenges since a large portion of language modeling data is from written text. According to Table I, although some spoken-style text e.g. LOTUS-BN and Thai BTEC have been included for language modeling, their contribution is only up to 4% of the overall training data. Additional spoken-style training data are still required.

The LOTUS-SOC corpus has been newly developed as described in the previous sub-section. The corpus is not only designed for robust acoustic modeling, but also for spoken-style language modeling by using Twitter text as reading scripts. Instead of combining the LOTUS-SOC text data into the overall language modeling set, language model interpolation shown in Equation (1) is conducted.

$$\log P(W) = \lambda \log P_{Baseline}(W) + (1-\lambda) \log P_{SOC}\,(W) \qquad (1)$$

$P(W)$ is an interpolated language model probability, $P_{Baseline}(W)$ and $P_{SOC}(W)$ are probabilities of the baseline and the LOTUS-SOC language models, and λ is an interpolation weight. Language model interpolation allows the system to bias its language modeling to either the baseline or the LOTUS-SOC one.

d. *Run-time System Architecture*

The baseline system architecture has to be improved when operating as a real service. Four features have been integrated as illustrated in Fig. 1. First, a Voice Activity Detection (VAD) module was used to segment an input speech at the client side so that the client can gradually send small speech chunks to the ASR server during segmenting. A more complicated VAD module was integrated also in the server side to improve the system robustness against background noise. Second, Speex, an open-source speech codec [24], was incorporated to encode the speech chunk at the client side before data transmission. Speex not only compresses the data, it also suppresses background noise. Since the Speex is a lossy compression, the ASR acoustic model has to be rebuilt from Speex decoded speech to eliminate the mismatch among training and run-time data. Third, data streaming was introduced among client/server transmission to reduce the waiting time required for input buffering. Using the released KALDI online decoder, all ASR models are preloaded before providing a service. The language model rescoring part described in the Section II.c is minimized to allow real-time processing. This of course degrades the overall system accuracy but tradeoffs for a better real-time factor. Almost all modules in the run-time architecture were developed in a multi-thread concept. Therefore, all the modules can function simultaneously to save the overall processing time.

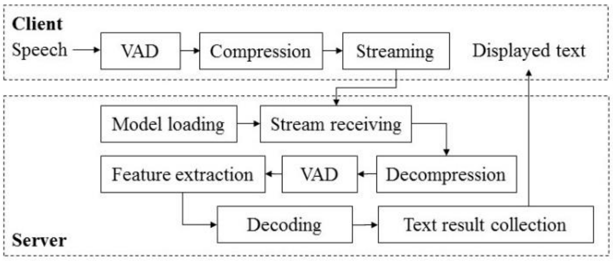

Fig. 1. An improved run-time system architecture.

Lastly, our system has been prepared for scalability by introducing a load balancing service that can receive multiple speech inputs and distribute to available multiple ASR engines running in different machines. Conventional distributed ASR employs a simple load balancing and queue management service. With an amount of ASR engines available in multiple machines, the load balancing service can share concurrent inputs to different machines in order to accelerate the recognition process.

For a higher flexibility, the *Docker* platform [25] has been adopted. Docker is an open source software allowing multiple applications, worker tasks, and other processes to run autonomously on a single physical machine or across multiple virtual machines. The ASR server part in Fig. 1 is split into a front-end service, which takes care audio streaming and load balancing; and a speech recognition engine, staring from feature extraction to text output. One Docker *container* is allocated for the front-end service and another set of containers is prepared for multiple speech recognition engines. The number of running speech recognition containers can be varied by the traffic of service requests. This Docker-based service is more resource efficient than the conventional distributed service as a new speech recognition container can be created on-the-fly whereas the number of speech recognition engines has to be fixed in the conventional one.

IV. Experiments

In this section, experiments aiming to show the performance improvement obtained by using each of the techniques described above. Subsection a. summarizes the experiments on hybrid word-subword language modeling, Subsection b. on robust acoustic modeling including both recent speech features, multi-conditioned and discriminative training, Subsection c. on spoken-style language modeling, and Subsection d. on the recent system architecture.

a. Experiments on Hybrid Word-Subword Language Modeling

To express the effectiveness of hybrid language modeling, a test set was obtained from three subsets: 2,200 utterances of 10 speakers from the LOTUS-BN, 300 utterances recorded by 3 speakers in office environment covering 5 genres (newspaper, law, novel, social media and web board), and 2,000 utterances from the U-STAR speech translation mobile application.

Fig. 2 illustrates comparative results between the hybrid language model system and the baseline system. It is clear that at a much smaller size of the system lexicon, the proposed hybrid technique can even lower the OOV rate, reduce the test set perplexity (which means easier recognition), and preserve the overall PM recognition error. Having this proposed technique, we later included more training text and our final run-time system contains 59,835 lexical units in which 11,035 are PMs.

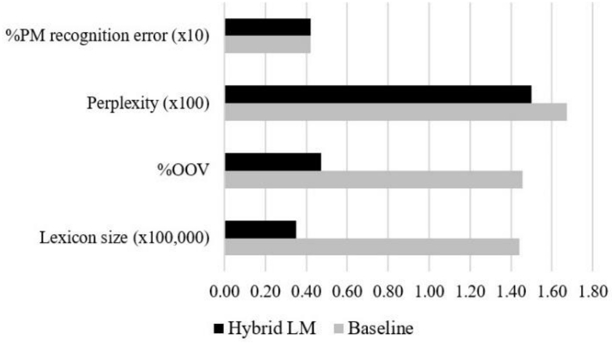

Fig. 2. Experimental results of the hybrid LM system against the baseline system.

b. Experiments on Robust Acoustic Modeling

To evaluate the system robustness against noisy speech, another test set was constructed. It contained 3,140 utterances from 3 speakers in the LOTUS-BN; 1,916 utterances from the U-STAR speech translation mobile application, and 5,586 utterances from 14 speakers in 7 noisy environments taken from the LOTUS-SOC.

TABLE III
Word error rate (%) results of the systems using LDA-MLLT, discriminative and multi-conditioned training.

Speech features		Training condition	
		Normal	Multi-conditioned
Baseline		50.4	34.9
LDA-MLLT	MPE	63.8	43.3
	MMI	49.1	32.4

Table III shows evaluation results of the baseline system and improved systems using the LDA-MLLT features, the two discriminative training methods (MPE and MMI), and the multi-conditioned noisy speech training, described in the Section III.b LDA-MLLT and MMI discriminative training slightly help reducing the Word Error Rate

(WER). Multi-conditioned training clearly shows its effectiveness to noise robust. The best setting is the LDA-MLLT, MMI, and multi-conditioned trained system which achieved a 32.4% WER on this open-vocabulary noisy test set, which is 35.7% relatively lower than the baseline WER result.

c. Experiments on Spoken-style Language Modeling

To further improve the language model for spoken-style speech input, Twitter scripts used in LOTUS-SOC were used to build another n-gram language model. The LOTUS-SOC language model was interpolated to the baseline model as described in Section III.c. In this experiment, the best acoustic model using LDA-MLLT and discriminative training described in the robust acoustic modeling experiment was conducted.

The LOTUS-SOC scripts contain 3.2 million words with 17,000 unique words. Trigram language model was constructed for both the baseline and the LOTUS-SOC data. The interpolation weight λ was empirically tuned to 0.68. Fig. 3 illustrates comparative results. The best result is obtained by language model interpolation and MMI-based acoustic modeling, which achieves over 10% relative reduction of WER.

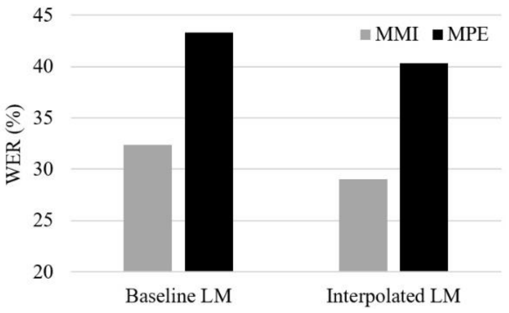

Fig. 3. Word error rate (%) results of the baseline and interpolated language models.

d. Experiments on the Run-time Performance

According to the newly designed run-time system explained in the Section III.c, we expect the system could operate in a much lower Real-time Factor (RTF); calculated by the total time required from the start of voice recording until the text output is shown, divided by the length of speech input. Since our ASR engine has been designed for scalability, its service can be multiplied and run in parallel to serve concurrent requests. We also expect our system to be acceptably fast in both broadband (WiFi) and narrow-band (3G) conditions. We simulated run-time usage by varying the number of concurrent inputs, the number of available ASR services, and a network condition. The broadband is set 30 Mbps uploading and 30 Mbps downloading, whereas the narrow-band is set 500 Kbps uploading and 1 Mbps downloading.

Fig. 4 presents RTF results from both the baseline architecture (denoted as "B" in the graph) and the improved architecture in the Fig. 1 (denoted as "I" in the graph). The results obviously show that at only one ASR service in the narrow-band condition, the new architecture can reduce the RTF down from 3.0xRT to about 1.2xRT, closed to that produced in the broadband network. One run-time service supports up to 5 concurrent requests at about 3.5xRT response time. The system footprint per service is recommended at least 6.2 GB RAM, 3 GB HDD, 2.6 GHz CPU, and 100 Mbps speed network adapter.

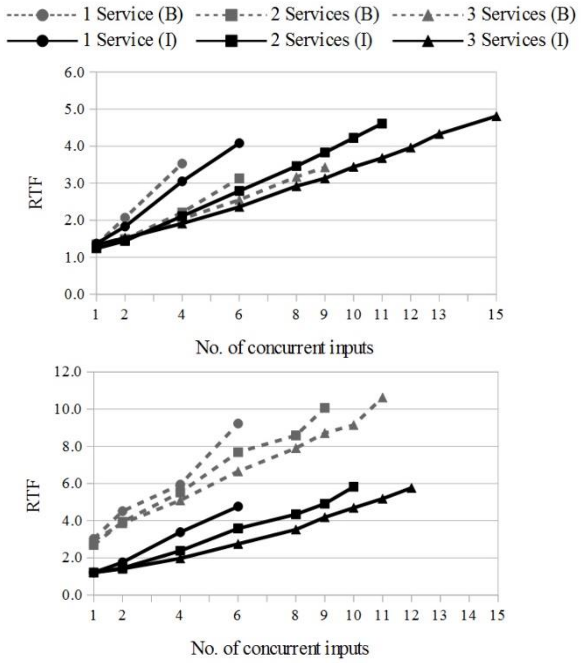

Fig. 4. RTF results of the systems running in a broadband (WiFi) condition (top), and a narrow-band (3G) condition (bottom), B and I denote the baseline and the improved systems.

TABLE IV
RTF results of the baseline (B), improved (I), and Docker-based improved (D) systems under a broadband (WiFi) condition.

System		No. of concurrent inputs				
		1	2	4	6	8
Baseline (B)	1 service	1.34	2.07	3.53		
	2 services	1.35	1.49	2.21	3.13	
	4 services	1.22	1.42	1.63	2.11	2.47
Improved (I)	1 service	1.37	1.83	3.05	4.08	
	2 services	1.24	1.44	2.11	2.79	3.46
	4 services	**1.37**	**1.41**	**1.61**	**2.07**	**2.43**
Docker (D)	2 services	1.23	1.60	2.18	2.81	3.46
	4 services	**1.21**	**1.26**	**1.60**	**2.01**	**2.35**

According to Section III.c, the run-time performance is expected to further improve by using the Docker platform. To evaluate the idea, a 16-core 2.6 GHz Intel Xeon CPU, 64 GB RAM, 1.5 TB HDD, and 1 Gb Ethernet adapter server was prepared. The improved system architecture described above, denoted "Improved (I)", and a new system deployed on the Docker platform, denoted "Docker (I+D)", were compared with the "Baseline (B)" system on the same server. In this experiment, the server and client were only connected via a broadband network (WiFi). Table IV summarizes comparative results. According to Table IV, the Docker-based system shows its efficiency obviously when a higher number of speech recognition services is available. The Docker-based system is not only benefit in reducing the processing RTF, its resources required are also much more efficient by the resource sharable feature. For example, the improved system requires about 10 GB for each speech recognition service whereas the Docker-based system consumes only about 1 GB each. More details are given in Chunwijitra et al. [26].

V. Conclusion and Discussion

This paper aimed at summarizing the key research and development issues, and showed the recent performance of a Thai open-vocabulary ASR system at NECTEC. Following the advanced algorithms on robust speech feature extraction, discriminative training, and multi-conditioned noisy speech training clearly raised the overall recognition accuracy on our real-noisy speech evaluation data. The novel hybrid word-subword language modeling method was shown to be highly efficient for making the system largely covers Thai lexical words at a small resource required. The system architecture was well designed to be ready for service deployment. The developed run-time system produced an acceptable response time and is scalable up on the user requirement.

Limitations of the system are of course existing. One major issue is the coverage of proper names always created every day. Although the system has been built with open-vocabulary in mind, real applications are often domain and vocabulary specific. Enlarging the system lexicon is not always right but with the current highly covered lexicon, a method to rapidly adapt the system to cope with such specific set of vocabulary is more attractive. Another issue is the 4-gram rescoring part, which can clearly increase the overall recognition accuracy, has been skipped in our run-time system to preserve a low RTF. There might be a better way to take the larger n-gram into account. In real applications, the system robustness against a variety of background noise is still open for research. Background music and speaker separation is needed to make the system usable.

Deep neural network (DNN) has been in the recent trend of modern ASR as it naturally handles the large variation of input speech. The values of DNN hidden layers have also been proven to be an efficient features for further conventional processing. Recurrent neural network (RNN) and Long short-term memory (LSTM) has also been investigated for modern language modeling as their better properties to capture long dependency than the conventional n-gram model. Our current research focuses on such DNN-based algorithms. The RNN language model has also been experimented for our future ASR system [12].

Acknowledgement

We would like to thank Dr. Atiwong Suchato from Chulalongkorn University and his team for research collaboration under this project, and Dr. Chiori Hori from National Institute of Information and Communications Technology (NICT) Japan for her help in visiting research.

References

1. Saon, G., Kuo, H. J., Rennie, S., Picheny, M.: The IBM 2015 English conversational telephone speech recognition system. In: Proc. INTERSPEECH 2015, Dresden, Germany (2015)
2. Schalkwyk, J., Beeferman, D., Beaufays, F., Byrne, B., Chelba, C., Cohen, M., Garrett, M., Strope, B.: Google search by voice: a case study. In: Advances in Speech Recognition: Mobile Environments, Call Centers and Clinics, Springer, pp. 61-90 (2010)
3. Shaik, M., Tüske, Z., Tahir, M., Nussbaum-Thom, M., Schlüter, R., Ney, N.: Improvements in RWTH LVCSR evaluation systems for Polish, Portuguese, English, Urdu, and Arabic. In: INTERSPEECH 2015, Dresden, Germany, pp. 3154-3157 (2015)
4. Kasuriya, S., Sornlertlamvanich, V., Cotsomrong, P., Kanokphara, S., Thatphithakkul, N.: Thai speech corpus for speech recognition. In: Oriental COCOSDA 2003, Singapore (2003)
5. Saykham, K., Chotimongkol, A., Wutiwiwatchai, C.: Online temporal language model adaptation for a Thai broadcast news transcription system. In: LREC 2010, Valletta, Malta (2010)
6. Chotimongkol, A., Thatphithakkul, N., Purodakananda, S., Wutiwiwatchai, C., Chootrakool, P., Hansakunbuntheung, C., Suchato, A., Boonpramuk, P.: The development of a large Thai telephone speech corpus: LOTUS-Cell 2.0. In: Oriental COCOSDA 2010, Kathmandu, Nepal (2010)
7. Chotimongkol, A., Chunwijitra, V., Thatphithakkul, S., Kurpukdee, N., Wutiwiwatchai, C.: Elicit spoken-style data from social media through a style classifier. In: Oriental COCOSDA 2015, Shanghai, China (2015)
8. Chotimingkol, A., Saykham, K., Thatphithakkul, N., Wutiwiwatchai, C.: Toward benchmarking a general-domain Thai LVCSR system, In: ECTI-CON 2010, Thailand (2010)
9. Universal Speech Translation Advanced Research (U-STAR) consortium, `http://www.ustar-consortium.com/`
10. Wutiwiwatchai, C., Thangthai, K., Sertsi, P.: Thai ASR development for network-based speech translation. In: Oriental COCOSDA 2012, Macau, China (2012)
11. Thangthai, K., Chotimongkol, A., Wutiwiwatchai, C.: A hybrid language model for open-vocabulary Thai LVCSR. In: INTERSPEECH 2013, Lyon, France (2013)
12. Chunwijitra, V., Chotimongkol, A., Wutiwiwatchai, C.: Combining multiple-type input units using recurrent neural network for LVCSR language modeling. In: INTERSPEECH 2015, Dresden, Germany (2015)
13. Kurpukdee, N., Sertsi, P., Chunwijitra, S., Chunwijitra, V., Chotimongkol, A., Wutiwiwatchai, C.: Enhance run-time performance with a collaborative distributed speech recognition framework. In: ICSEC 2015, Thailand (2015)
14. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K.: The Kaldi speech recognition toolkit. In: ASRU 2011, Hawaii, US (2011)
15. Stolcke, A.: SRILM – an extensible language modeling toolkit. In: ICSLP 2002, Colorado, US (2002)
16. El-Desoky, A., Gollan, C., Rybach, D., Schlüter, R., and Ney, H.: Investigating the use of morphological decomposition and diacritization for improving Arabic LVCSR. In: INTERSPEECH 2009, Brighton, UK, pp. 2679 – 2682 (2009)

17. Kwon, O. W., Park, J.: Korean large vocabulary continuous speech recognition with morpheme-based recognition units. Speech Communication, 39(3):287-300 (2003)
18. Jongtaveesataporn, M., Thienlikit, I., Wutiwiwatchai, C., Furui, S.: Lexical units for Thai LVCSR. Speech Communication, 51(4): 379-389 (2009)
19. Aroonmanakul, W.: Collocation and Thai word segmentation. In: SNLP-Oriental COCOSDA 2002, Prachuapkirikhan, Thailand, pp. 68-75 (2002)
20. Haeb-Umbach, R., Ney, H.: Linear discriminant analysis for improved large vocabulary continuous speech recognition. In: ICASSP 1992, pp. 13–16 (1992)
21. Gopinath, R.: Maximum likelihood modeling with Gaussian distributions for classification. In ICASSP 1998, vol. 2, pp. 661– 664 (1998)
22. Bahl, L., Brown, P., de Souza, P., Mercer, R.: Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In: ICASSP 1986, vol. 1, pp. 49-52 (1986)
23. Povey, D., Woodland, P.: Minimum phone error and i-smoothing for improved discriminative training. In: ICASSP, Kyoto, Japan (2012)
24. Speex: a free codec for free speech, `http://www.speex.org/`
25. Bernstein, D.: Containers and cloud: From lxc to docker to kubernetes. IEEE Cloud Computing, vol.1, no.3, pp.81–84, Sept 2014.
26. Chunwijitra, S., Junlouchai, C., Krairaksa, K., Chunwijitra, V., Wutiwiwatchai, C.: A cloud-based framework for Thai large vocabulary speech recognition. In: ECTI-CON 2016, Chianmai, Thailand (2016).

Chai Wutiwiwatchai received Ph.D. in Computer Science from Tokyo Institute of Technology in 2004. He is now the Director of Intelligent Informatics Research Unit, National Electronics and Computer Technology Center (NECTEC), Thailand. His research interests include speech processing, natural language processing, and human-machine interaction. His research work includes several international collaborative projects in a wide area of speech and language processing as well as nation-wide e-Learning. He is now a member of the International Speech Communication Association (ISCA) and the Institute of Electronics, Information and Communication Engineers (IEICE).

Vataya Chunwijitra received the B.Sc. degree in computer science and M.Sc. degree in computer technology from King Mongkut's Institute of Technology North Bangkok, Thailand, in 2000 and 2005 respectively, and Dr.Eng. degree (2013) in information processing from Tokyo Institute of Technology, Tokyo, Japan. She is currently a researcher at Speech and Audio Technology Laboratory, NECTEC, Thailand. Her research interests include speech synthesis and speech recognition.

Sila Chunwijitra graduated with a Ph.D. in Informatics from The Graduate University for Advanced Studies, Japan in 2012. He is a researcher at National Electronics and Computer Technology Center (NECTEC), Thailand. His researches include e-Learning system, video conferencing, web applications and open source technologies. He is also an engineering working on a speech recognition system for Thai language.

Phuttapong Sertsi graduated with Bachelor Degrees in Computer Science from Khon kaen University, Thailand in 2010.He currently works at Speech and Audio Technology Laboratory, NECTEC, where he conducts several research works on natural language processing and speech recognition especially on language modeling.

Sawit Kasuriya earned his M.Eng in Electrical Engineering (Digital Signal Processing) from Chulalongkorn University, Thailand. He is interested in speech technology and natural language processing, and working as a researcher at National Electronics and Computer Technology Center (NECTEC).

Patcharika Chootrakool graduated Master of Science in Informatics Technology from King Mongkut's University of Technology Thonburi (KMUTT), in 2004. She received bachelor degree in Linguistics from Thammasat University, in 1999. She is a researcher at National Electronics and Computer Technology Center (NECTEC), Thailand. She is working on a Thai speech recognition system with her expertise in Thai Linguistics.

Kwanchiva Thangthai is currently a PhD student in Speech, Language and Audio Laboratory, Department of Computing Science, University of East Anglia, UK. She obtained her B.Sc. (Computer Science) from Naresuan University, Thailand, in 2006. She has been working as an assistant researcher at National Electronic and Computer Technology Center (NECTEC), Thailand, since 2006. Her research interests are Multimodal Speech Processing and Speech Recognition.

Chanchai Junlouchai received his B.Sc. degree in Computer Science from Burapha University, Thailand in 2004. He is a research assistant, for National Electronic and Computer Technology Center since 2003. His researches focus on open-source software, Linux operation system, cloud computing, API and open services.

Kamthorn Krairaksa graduated with a B.Sc. degree in Computer Science from Khon Kaen University, Thailand in 2000. He is a research assistant at National Electronics and Computer Technology Center (NECTEC), Thailand. His researches include Service Innovation and Cloud Technology.