19th International Computer Science and Engineering Conference (ICSEC)
Chiang Mai, Thailand, 23-26 November, 2015

Enhance Run-time Performance with a Collaborative Distributed Speech Recognition Framework

Nattapong Kurpukdee*, Phuttapong Sertsi*, Sila Chunwijitra*,
Vataya Chunwijitra*, Ananlada Chotimongkol* and Chai Wutiwiwatchai*
* NECTEC, National Science and Technology Development Agency (NSTDA),
112 Pahonyothin Road, Pathumthani, 12120, Thailand
Email: {nattapong.kurpukdee, phuttapong.sertsi, sila.chunwijitra,
vataya.chunwijitra, ananlada.chotimongkol, chai.wutiwiwatchai} @nectec.or.th

***Abstract*—This paper presents an improvement of a distributed Thai speech recognizer, aiming to enhance system response time as measured by a real-time factor (RTF) for a better user experience. The system is designed based on a collaborative multi-agents and task workers concept. A *Streaming Agent* is introduced to manage speech signal transfer while a *Recognition Agent* is applied to manage speech recognition task distribution. The speech recognition task is distributed to an available pipeline of *task workers*, which contain speech recognition core engines. A concept of task worker is introduced to provide light-weight management for each individual task in the pipeline. Both multi-agents and task workers are designed to work synchronously in order to minimize the overall processing time, especially in narrow-band or unstable network environment. The proposed improved system is compared with a traditional system in terms of their recognition word error rate (WER) and RTF. The results show that the implementation of speech codec, multi-agents and task workers in the proposed framework can substantially reduce the computational cost in terms of RTF by 42.7% on average in a narrow-band mobile network. In addition, there is no significant difference in WER between the proposed and baseline systems.**

Keywords*—Distributed speech recognition, Thai speech recognizer, Multi agents, Task workers.

I. Introduction

Voice-controlled interface becomes more important in the Internet of Things (IoT) generation as it has been applied in mobile devices, entertainment systems, automobiles, etc. A number of people who speak to their smart phones – asking them to send e-mail and text messages, search for directions, or find information on the web – is now growing. Speech recognition (SR) technology which can convert speech into text makes it easier to both create and access information on a small device where a keyboard maybe small or even not available. Speech recognition performance is primarily relied on the quality of its acoustic and language models; higher requirements along these dimensions typically implies increasing complexity of the models. This, thus, increases both computation and memory requirements of the system. A large vocabulary continuous speech recognition (LVCSR) system which can handle speech input from unrestricted domains and vocabulary typically requires complex acoustic and language models. However, mobile devices and other embedded systems tend to have limited resources, i.e. lack of computational power, memory and storage capacities, which have to be shared among multiple tasks.

Since mobile devices are equipped with wireless connectivity, it is therefore considered more resource efficient to perform a recognition task on a remote server. A distributed speech recognition (DSR) architecture, where a speech utterance is obtained on a mobile device and transmitted via a wireless network to a remote recognition engine on a server, enables a low resource device to have speech recognition capability. With explosive growth of wireless communication and speech technology, a DSR system plays an important role in speech applications on mobile devices, smart phones, and tablet computers and also allows data access via speech even when moving across wireless networks.

In the DSR framework, transmitting speech data over a wireless network is another challenging task as the quality of wireless communication is still quite poor in developing countries. In Thailand, there are many problems with wireless communication in a rural area, e.g. lack of high-speed Internet, a large number of clients per mobile repeater, and unstable network condition. In some areas only EDGE or 3G network is provided and shared among many local nodes. Moreover, the actual speed of the available network does not reach the theoretical maximum speed due to the aforementioned problems. For instance, the bandwidth of the network in a major city can be as high as up to 2 Mbps, while in a rural area the bandwidth is quite limit at 500 Kbps or less. This is the typical example of 3G situation in Thailand. Since a wave file, without any compression, is quite large, transmitting speech signal over a narrow-band environment is quite slow. The DSR, therefore, would not perform at an acceptable response time especially in the narrow-band environment.

Various methods of DSR have been proposed to reduce the total response time consumed by the complex processing and the network traffic. The speech signal is transmitted to a server through a wireless connection by using a single speech utterance to reduce heavy processing on the client side [1]. To reduce a heavy burden on the network, a speech coding technique was used to compress speech data before transmitting over the Internet [2]. In [3], [4], the front-end processing was done at the client side, then the speech features were transmitted to a remote server where speech decoding was performed. These strategies, however, were based on a client-server model where high computational processes were place on a server side, but there was no inter-process management to streamline the processes to further reduce the computational cost.

The aim of this work is to apply an inter-operation of multi-agents and task workers for gaining up the system performance in distributed speech recognition. Agents are utilized to manage data transfer between a server host and client nodes, and to manipulate the workload of SR engines with load balancing. A concept of *task worker* is introduced to provide light-weight management for each individual task in a pipeline of processes. In our proposed collaborative DSR framework, task workers are applied to an SR engine where the recognition process is divided into a set of small tasks; each task is individually handled by a task worker. The proposed method also enhances the scalability of the distributed Thai LVCSR system. As the demand of an application increases, a new SR engine can be easily added to the framework, and the agents will operate on the new SR engine immediately. In addition, speech compression is also deployed to decrease the response time of the LVCSR system when a client is in a narrow-band environment. The proposed improvement is mainly intended to be used for applications distributed over a narrow-band network because it helps reduce the amount of data transferring over the network and also streamline the processes on the server side to reduce the wait time.

This manuscript is organized as follows: The response time problem of an LVCSR system is described in section 2. Section 3 explains the architecture of a conventional DSR system. Section 4 then describes the proposed DSR system including system architecture, integration of functionalities, and system implementation. Section 5 discusses the experiments and evaluation results of the proposed DSR architecture. Finally, the summary of the proposed framework is given in Section 6.

II. The response time problem

At the moment, the performance and accuracy of Automatic Speech Recognition (ASR) systems are quite far from perfect. One key performance issue is the speed or response time which usually measured in terms of real-time factor (RTF). The RTF is defined as the total recognition processing time divided by the total time of that speech utterance being uttered, i.e. the length of the input wave file. For efficient human-computer interaction, the RTF should be small; otherwise, users will have to wait for a long time before the system could react to their speech input. There are many factors, e.g. noise, language richness, and variation in speaking styles [5], [6], [7], that can intensify computational complexity of an ASR system, and thus increase RTF. For example, noises are the classic unwanted information that have to be filtered out from the input signal before feeding to the recognition process; voice characteristics of speakers also have an effect on speech signal e.g. male and female speakers have different voices and pitch ranges. Hence, we need more complex algorithms and methods to do the pre-processing, decoding, post-processing processes to deal with these factors. Typically, more resources are required to handle complex ASR algorithms and also to reduce the RTF.

For a distributed speech recognition (DSR) system, there are also other factors that have influence on the system performance. Fig 1 shows a conceptual framework of a DSR system that consists of multiple clients and an ASR server connected through the Internet. Based on this figure, each client sends captured audio signal to the ASR server. A

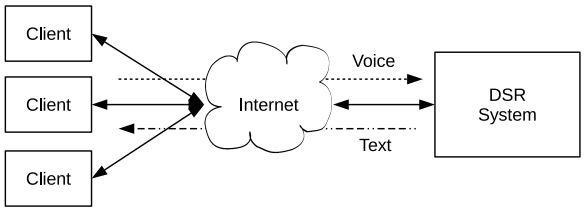

Fig. 1. A basic framework of a distributed speech recognition (DSR) system.

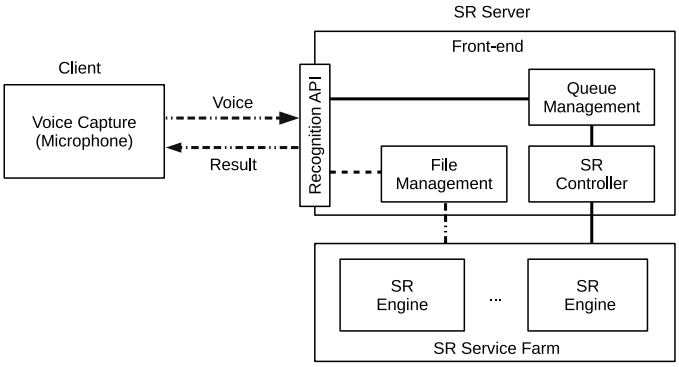

Fig. 2. The design of a traditional Thai DSR architecture.

speech recognition process is then executed on the server which sends back a text result to the requesting client after the recognition process is finished. The ASR server can be utilized to serve multiple clients according to the one-to-many concept. As the architecture of a DSR system is designed for supporting a recognition process via network infrastructure, the communication between clients and a server is another important factor that affects the ASR performance. For a DSR system, the response speed from the client point of view relies on the performance of the network environment. This is an important issue that could increase the RTF of the system [8], [9], [10]. Several factors in the network environment such as the bandwidth of the network, the usage traffic at the moment, network stability between client and server nodes, etc., can affect overall network performance and thus the DSR system. One common problem is a bottleneck stage that causes the entire process to slow down or even terminate. Not only those identified factors but also the size of data transmission that contributes to a high RTF value. A big chunk of data requires longer time to transmit than a small chunk in the same network bandwidth [11], and hence increases the wait time of succeeding processes. For this reason, the RTF could be decreased if we reduce the amount of data transferring between a requesting node and a responding node.

III. A Traditional Thai DSR system

The architecture of a traditional Thai DSR system is shown in Fig 2. The system is designed to support scalability. The number of SR engines can be increased to support more concurrent connections from clients which help increase the performance of the overall system. Speech clients can be on different platforms e.g. personal computer and mobile phone. A speech recognition process starts from the client side. When a user starts to speak, the voice signal is captured by a

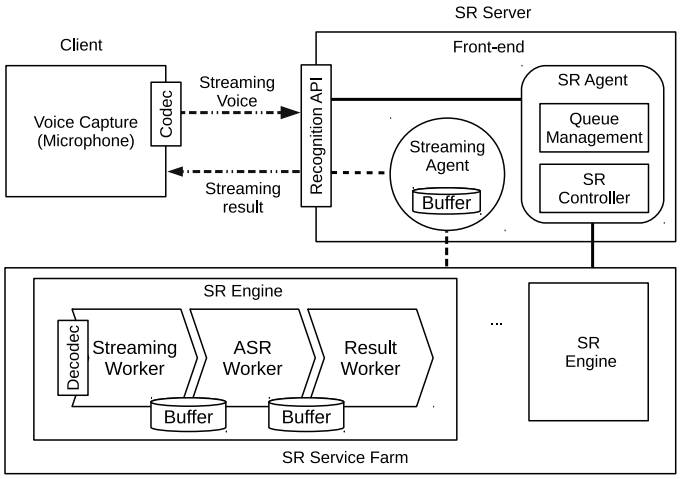

Fig. 3. The design of a collaborative DSR architecture with multi-agents and SR workers pipeline.

microphone on a computer or a phone, and then transmitted to the server directly. This means that there is no pre-processing step on the client side and the raw voice data are transmitted. At the server side, there are two main parts: *SR front end* and *SR service farm*. The SR front end is a portal or gateway of the system directly connecting to the clients, while the SR service farm is a pool of working SR engines that provide distributed recognition services. The *queue management module* works collaboratively with the *SR controller* which is used to manage speech recognition engines. When the system gets a newly connected client, the SR controller chooses an available SR engine, i.e. the one with the shortest queue, to serve that client automatically. At the same time, the *file management module* receives audio data from the connected client and saves the data as a local file. After that, the saved file is fed to a selected SR engine for recognition. Finally, the result is transferred back to the requesting client after the recognition process is finished.

There are some response time problems when using a traditional DSR system in a narrow-band or unstable network environment. We investigated the causal factors to identify the root causes of the problems. We found that the data transferring rate is a primary cause affecting to the RTF. It takes time to transfer a raw speech signal from a client to a server. The long transfer time also causes other problems, for instance, 1) the processing time of an SR engine is increased since an SR engine has to wait for the entire audio data before it can begin the recognition process, 2) no recognition result is returned to the client node due to the network timeout termination, and 3) an SR engine is locked to the paired client and cannot serve other requests. To solve these problems, some improvements on the DSR architecture are required.

IV. A Collaborative DSR Framework

This section explains the system design, data structure and implementation of the proposed collaborative DSR framework which aimed at boosting the system efficiency of a speech recognition system in terms of the response speed from a client point of view.

Regarding the causal factors that affect the system RTF in the traditional framework, we re-design the system architecture to solve those causes. The enhanced system architecture is shown in Fig 3. In this design, the key improvement is to alleviate the data transferring problem both between a client and a server and within the recognition process. To do so, we utilize the agents [12], [13] and pipeline of workers concepts to help streamline the recognition process. We also apply an audio compression method by using a speech encoder/decoder to reduce the size of the speech signal. The Speex audio compression, also known as speech codec, is utilized to encode and decode the transmitting speech data [14]. All improvements are targeted for a DSR system in a narrow-band or unstable network environment. Although, the proposed collaborative framework of a DSR system looks quite similar to the conventional framework at the conceptual level, they are different in the methodologies. The particular details are described below;

A. Intelligent Multi-Agents

The agent-based computing technology has been proposed for interacting intelligently to solve the problem that is beyond the capability of each individual problem solver. One key property of the software agent is that it is encapsulated, i.e. self-contained, and can migrate among different machines without effecting the execution. The use of the agent technology in a distributed system has several advantages on the system performance. The most important one would be a system run time which is the focus of this paper. As each agent can operate dependently from each other, this can reduce the wait time between processes and thus improve the system response time [15]. In this research, agents are added to the front-end of the server side in order to remove data dependency between a client node and an SR engine. Two type of agents, *Streaming Agent* and *SR Agent*, are designed to replace existing modules in the traditional DSR framework. The Streaming Agent, which is responsible for speech signal handling, is introduced to replace the file management module. The SR Agent, which composes of two components – queue management and SR controller, is utilized to control SR engines instead of the old modules. Both agents independently operate their responsible tasks with the clients and SR engines.

When a new client connects to the DSR system, agents are arranged to operate on the requesting client automatically. Instead of raw speech data, in the new framework, a client is also asked to compress the input voice signal with codec before transferring the data to the server. The Streaming Agent is the one that communicates directly with the requesting client. The streaming voice input from the client is immediately stored in the data buffer of the Streaming Agent awaiting to be forwarded to a selected SR engine when it is ready. Meanwhile, the SR Agent is taking the responsibility to manage a processing queue and find an available SR engine to take up the request from the client. Besides, not only the input speech signal but also the recognition result is organized by the Streaming Agent. A Streaming Agent automatically retrieves the result generated from an SR engine and saves it the result buffer. The text result is returned to the requesting client as a stream of word.

B. Task Workers for Speech Recognition

In the traditional framework, the SR engine is designed as a series of recognition processes. Each process is operated dependently on the previous process and resource. For example, a decoding process is not operated while an SR engine still receiving input speech data; likewise, the recognition result is returned to the requesting client only after the decoding processing generates the final output. This inter-process dependency can increase the system RTF if the system operates on an unstable or narrow-band network.

A concept of *task worker* is utilized to provide light-weight management for each individual task. A task that can be managed by a task worker is defined as a small identifiable and essential piece of a job that serves as a unit of work. In the proposed framework, to enhance the system operation, we divide and arrange the processes in the SR engine into a pipeline of task workers. The collaborative DSR architecture consists of three task workers, *Streaming Worker*, *ASR Worker*, and *Result Worker* as shown in Fig 3. Each worker can independently complete its task without waiting for each other. As a result, the proposed collaborative framework can provide a faster response which enhances user experience and also maintain system stability in unstable network environment. Common resources such as shared buffers are also prepared for data sharing among workers. In the collaborative DSR architecture, there are two shared buffers placing between the related task workers. The detail of each worker is described in more details as following;

1) Streaming Worker: The Streaming Worker is tasked to operate on the encoded input streaming from the client. In fact, this worker directly communicates with the Streaming Agent, not the client node. The speech signal is fed from the Streaming Agent buffer. The worker executes the speech decoding process to decompress the encoded input signal and then store it to the shared buffer. It will repeat the processes until the end of the input streaming.

2) ASR Worker: The ASR Worker is utilized to recognize the input speech signal and generate the output text. This worker executes all important speech recognition processes. The worker is on-the-fly started when speech data appears in the input shared buffer. The worker retrieves a chunk of speech data and activates the recognition processes to generate the results. At the end, the recognition result is delivered to the shared buffer of the output data.

3) Result Worker: The Result Worker is the last worker in charging of returning the recognition result from the output shared buffer to the Streaming Agent. The recognition output is transmitted between the worker and the agent as a continuous stream of text output.

V. Experiments

To verify our scheme, we evaluate the system performance on both recognition accuracy and computational efficiency training and test data are described in Sect. V-A. The performance of the proposed collaborative DSR system is examined with two metrics: word error rate (WER) and real-time factor (RTF) as discussed in Sect. V-C and V-B respectively.

A. Experimental settings

Our acoustic model was trained from 224 hours of speech data from LOTUS [16], LOTUS-BN [17] and VoiceTra4U-M. VoiceTra4U-M is a speech translation application in travel and sport domains created by Universal Speech Translation Advanced Research (U-STAR) project [18]. 22 hours of speech were recorded on mobile devices in a real usage environment. Kaldi Speech Recognition Toolkit [19] was used to train a conventional GMM-based acoustic model. We also applied the Minimum Phone Error (MPE) discriminative training technique [20] to the trained acoustic model. Speech features are in the form of a 39 dimensional feature vector composes of 12 MFCCs augmented with log energy their first and second derivatives. The features were extracted from 25 ms window frame of speech data shifted by 10 ms each time. Features from a context window of 3 frames to the left and right were also included. A Linear Discriminate Analysis (LDA) was also applied to the feature space to reduce feature dimensions to 40.

Language model training data contain 67M words from four text sources – BEST [21], LOTUS-BN [17], HIT-BTEC [22], and Thai web blogs. These texts cover variety of domains, e.g. law, drama, news, life-style and travel, with the vocabulary size of 66K. We used SRILM toolkit [23] with modified KneserNey discounting to a train trigram hybrid language model described in [24].

The evaluation data are obtained from two different recognition tasks: VoiceTra4U-M and Dictation. The VoiceTra4U-M test set consists of 1,916 utterances with multiple speakers recorded from mobile devices in a real usage environment. The VoiceTra4U-M test set is not included in the training data. The Dictation test set is a set of prepared speech of 300 utterances recorded by 3 speakers in an office environment. The test utterances cover 5 genres: newspaper, law, novel, social media and web board.

B. Computational efficiency

To investigate the run time efficiency of all recognition processes when multi-agents and task workers were implemented using the proposed collaborative framework, we compared the computational time of the proposed framework (P) with the computational time of the baseline traditional framework (B). The RTF is used to measure the computational efficiency of both DSR systems. RTF is defined as total recognition processing time, starting from the time the client transmits the input signal to the time the client receives the output text, on a 2.6GHz Intel Xeon with 64 GB memory, divided by the total time of the speech utterances being uttered. Ten-second long speech data were used to evaluate the system RTF in two network bandwidth specifications: 3G narrow-band mobile network and wireless broadband network. To simulate the 3G mobile bandwidth, we specifically set the average download and upload speeds of the data transfer to 1 Mbps and 500 Kbps, respectively. In the wireless broadband environment, the average speed is up to 34.85 Mbps for downloading and 33.59 Mbps for uploading. In addition, we also examined the computational performance of both DSR frameworks in processing multiple concurrences of input data. For each DSR framework, up to three SR services are experimented to process up to fifteen input concurrences.

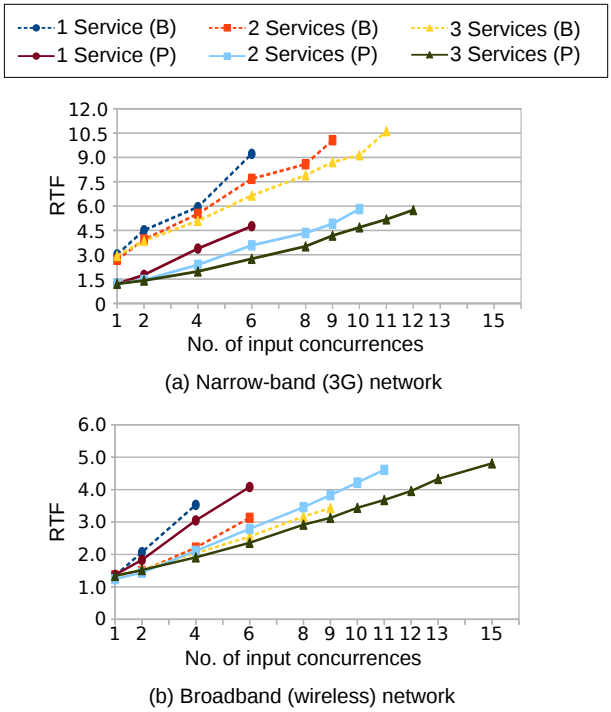

Fig. 4. RTF of DSR frameworks on broadband and narrow-band networks.

Fig. 4 shows the RTF results examined under two network conditions (a) narrow-band (3G) network and (b) broadband (wireless) network. The baseline traditional DSR framework (B) is represented by dashed lines while the proposed collaborative framework (P) is represented by solid lines. From the results shown in Fig. 4 (a), the narrow-band network, our collaborative architecture can considerably reduce the computational cost from 3.0 RTF in the traditional system to 1.2 RTF when one SR service was used to serve one input concurrence. On average, of all numbers of input concurrences, the RTF can be decreased by 2.9, 3.0 and 3.5 absolute compared to the traditional system when 1, 2 and 3 SR services were utilized respectively. This demonstrates that the implementation of speech codec, multi-agents and task workers in the proposed collaborative architecture can substantially reduce the computational cost in the narrow-band mobile network with the average relative RTF reduction of 42.7% in all experiment scenarios. In the broadband (wireless) case shown in Fig. 4 (b), it can be seen that the proposed framework has slightly low RTF than the baseline traditional one in all cases. The proposed collaborative framework of multi-agents and task workers may not substantially reduce the system RTF in the broadband situation as in the narrow-band network; however, it can support more input concurrences. Especially in the case of 3 SR services, the collaborative framework can support almost twice the number of input concurrences without increasing the RTF. As expected, the narrow-band network can handle fewer simultaneous inputs than the broadband one.

C. Speech recognition accuracy

To demonstrate that the proposed collaborative architecture dose not affect recognition accuracy, the word error rates of both DSR systems were compared. We evaluated recognition results on two different tasks as described in Sect. V-A. Note that, the same acoustic and language model were used in both the baseline traditional framework and the proposed collaborative framework to fairly compare recognition performances. Word Error Rates (WERs) of both systems reported in Table I.

TABLE I. RECOGNITION ACCURACY RESULTS

Test set	WER (%)	
	Baseline framework	Proposed framework
VoiceTra4U-M	20.65	20.59
Dictation	26.24	26.04
Average	23.45	23.32

In the proposed framework, speech codec was applied to compress speech signal before transmitting the data to the server. The data compression could reduce the signal quality and thus affect the recognition process. From the results, we can clearly see that the recognition results of our proposed collaborative DSR system is comparable to those of the traditional system. In fact, the WER of the proposed framework is slightly higher than that of the baseline framework, but with no significant difference. The results demonstrate that there is no negative impact from our proposed architecture and the use of speech codec on the recognition accuracy.

VI. CONCLUSION

In this paper, we compared two distributed speech recognition architectures – the traditional system and the proposed collaborative multi-agents and task workers system – for Thai speech-to-text processing. The proposed framework aims primarily at enhancing system response time as measured by a real-time factor (RTF) for a better user experience on a narrow-band mobile network. We utilized agent-based technology to take charge of the speech recognition front-end processes to remove data dependency between a client node and an speech recognition engine. Task workers are applied to a speech recognition engine where the recognition process is divided into a set of small tasks to streamline the process and reduce the wait time. We tested both DSR systems on two Thai LVCSR tasks: speech-to-speech translation and dictation. The performance of both systems is compared in terms of run-time efficiency and recognition accuracy. The results show that the traditional DSR system which sends the raw speech file directly to the server without any agent or worker implementation requires much more computation. The computational complexity of the proposed framework can be reduced by using the speech codec, multi-agent and worker implementation. The computational cost is substantially reduced by 42.7% in terms of RTF on average in a narrow-band mobile network. In a broad-band wireless network, the proposed framework can support more input concurrences than the traditional one without increasing the RTF. In terms of recognition accuracy, there is no significant difference in WER between the two systems. With the proposed architecture, the recognition time can be saved without affecting recognition accuracy. This advantage is important to the DSR system, where the processing power and memory available are limited.

References

[1] D. Vaufreydaz, J. Rouillard, and M. Akbar, "A network architecture for building applications that use speech recognition and/or synthesis," in *Eurospeech'99*, Budapest, Hungary, Sep. 1999, pp. pp. 2159–2162.

[2] Z. Tu and P. C. Loizou, "Speech recognition over the internet using java," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999.

[3] V. Digalakis, L. Neumeyer, and M. Perakakis, "Quantization of cepstral parameters for speech recognition over the world wide web," *Selected Areas in Communications, IEEE Journal on*, vol. 17, no. 1, pp. 82–90, Jan 1999.

[4] B. Raj, J. Migdal, and R. Singh, "Distributed speech recognition with codec parameters," in *Automatic Speech Recognition and Understanding, 2001. ASRU '01. IEEE Workshop on*, 2001, pp. 127–130.

[5] M. Forsberg, "Why is speech recognition difficult?" 2003. [Online]. Available: http://www.speech.kth.se/~rolf/gslt_papers/MarkusForsberg.pdf

[6] U. Shrawankar and V. Thakare, "Adverse conditions and asr techniques for robust speech user interface," *JCSI International Journal of Computer Science Issues*, vol. 8, no. 3, pp. 440–449, Sep 2011.

[7] T. Fulcher, "What's the deal with automatic speech recognition?" 2012. [Online]. Available: http://www.cs.unm.edu/~pdevineni/papers/Fulcher.pdf

[8] D. Pearce, "Enabling new speech driven services for mobile devices: An overview of the etsi standards activities for distributed speech recognition front-ends," *AVIOS 2000: The Speech Applications Conference*, vol. 2000, pp. 261–264, May 2000.

[9] B. Delaney, N. Jayant, and T. Simunic, "Energy-aware distributed speech recognition for wireless mobile devices," *Design Test of Computers, IEEE*, vol. 22, no. 1, pp. 39–49, Jan 2005.

[10] A. Gomez, A. Peinado, V. Sanchez, and A. Rubio, "Recognition of coded speech transmitted over wireless channels," *Wireless Communications, IEEE Transactions on*, vol. 5, no. 9, pp. 2555–2562, September 2006.

[11] D. Vlaj, B. Kotnik, B. Horvat, and Z. Kačič, "A computationally efficient mel-filter bank vad algorithm for distributed speech recognition systems," *EURASIP J. Appl. Signal Process.*, vol. 2005, pp. 487–497, Jan. 2005.

[12] A. Ahmad, A. Nordin, E. Saaim, D. F. bin Samaon, and M. Ibrahim, "An architecture design of the intelligent agent for speech recognition and translation," in *TENCON 2004. 2004 IEEE Region 10 Conference*, vol. B, Nov 2004, pp. 255–258 Vol. 2.

[13] E. H. M. Saaim, M. A. Alias, A. M. Ahmad, and J. N. Ahmad, "Applying mobile agent for internet-based distributed speech recognition," in *The International Conference on Computer Applications in Shipbuilding, ICCAS 2005*, Jun 2005, pp. 134–138.

[14] Speex, "Speex: A free codec for free speech." [Online]. Available: http://www.speex.org

[15] E. H. M. Saaim, A. M. Ahmad, M. A. Alias, and M. F. Othman, "Mobile agent: Agent design pattern for distributed speech recognition system," Jun 2006, pp. 430–434.

[16] S. Kasuriya, V. Sornlertlamvanich, P. Cotsomrong, S. Kanokphara, and N. Thatphithakkul, "Thai speech corpus for Thai speech recognition," in *Proc. Oriental COCOSDA 2003*, Jun. 2003, pp. 54–61.

[17] A. Chotimongkol, K. Saykhum, P. Chootrakool, N. Thatphithakkul, and C. Wutiwiwatchai, "LOTUS-BN: A thai broadcast news corpus and its research applications," in *Speech Database and Assessments, 2009 Oriental COCOSDA International Conference on*, Aug 2009, pp. 44–50.

[18] U-STAR, "Universal speech translation advanced research (u-star)." [Online]. Available: http://www.ustar-consortium.com/

[19] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011.

[20] D. Povey and P. Woodland, "Minimum phone error and i-smoothing for improved discriminative training," in *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, vol. 1, May 2002, pp. I–105–I–108.

[21] K. Kosawat, M. Boriboon, P. Chootrakool, A. Chotimongkol, S. Klaithin, S. Kongyoung, K. Kriengket, S. Phaholphinyo, S. Purodakananda, T. Thanakulwarapas, and C. Wutiwiwatchai, "Best 2009 : Thai word segmentation software contest," in *Natural Language Processing, 2009. SNLP '09. Eighth International Symposium on*, Oct 2009, pp. 83–88.

[22] G. Kikui, E. Sumita, T. Takezawa, and S. Yamamoto, "Creating corpora for speech-to-speech translation," in *8th European Conference on Speech Communication and Technology, EUROSPEECH 2003 - INTERSPEECH 2003, Geneva, Switzerland, September 1-4, 2003*, 2003.

[23] A. Stolcke *et al.*, "Srilm-an extensible language modeling toolkit." in *INTERSPEECH*, 2002.

[24] K. Thangthai, A. Chotimongkol, and C. Wutiwiwatchai, "A hybrid language model for open-vocabulary thai lvcsr." in *INTERSPEECH*, 2013, pp. 2207–2211.