Journal of Computer Science and Cybernetics, V.34, N.4 (2018), 335–348

DOI 10.15625/1813-9663/34/4/13165

DEVELOPMENT OF HIGH-PERFORMANCE AND LARGE-SCALE VIETNAMESE AUTOMATIC SPEECH RECOGNITION SYSTEMS

QUOC TRUONG DO[1], PHAM NGOC PHUONG[1], HOANG TUNG TRAN [2], CHI MAI LUONG[1]

[1] *Vietnamese Artificial Intelligent Systems; research@vais.vn*
[2] *ICTLab, USTH, Vietnam*

Abstract. Automatic Speech Recognition (ASR) systems convert human speech into corresponding transcription automatically. They have a wide range of applications such as controlling robots, call center analytic, voice chatbot. Recent studies on ASR for English have achieved the performance that surpass human ability. The systems were trained on a large amount of training data and performed well under many environments. With regards to Vietnamese, there have been many studies on improving the performance of existing ASR systems, however, many of them are conducted on a small-scaled data, which does not reflect realistic scenarios. Although the corpora used to train the system were carefully design to maintain phonetic balance properties, efforts in collecting them at a large-scale is still limited. Specifically, only a certain accent of Vietnam was evaluated in existing works. In this paper, we first describe our efforts in collecting a large data set that covers all 3 major accents of Vietnam located in the Northern, Center, and Southern regions. Then, we detail our ASR system development procedure utilizing the collected data set and evaluating different model architectures to find the best structure for Vietnamese. In the VLSP 2018 challenge, our system achieved the best performance with 6.5% WER and on our internal test set with more than 10 hours of speech collected real environments, the system also performs well with 11% WER.

Keywords. ASR; Automatic speech recognition; Vietnamese corpora; Vietnamese Speech recognition.

1. INTRODUCTION

The researches on automatic speech recognition (ASR) of Vietnamese have made significant progresses since it was first introduced more than twenty years ago. However, this ASR of Vietnamese is just at its experimental stage and yet to reach the performance level required to be widely used in real-life applications.

The work by Vu et al. [15] was the first attempt constructing a large vocabulary Vietnamese ASR system. The authors conducted an intensive study on Vietnamese phonetic structure and also built a phoneme set used for Vietnamese ASR. The works in [9, 10] made a number of improvements on optimizing the pronunciation dictionary and input features specifically for Vietnamese. Most of existing works are, however, still either based on HMM-GMM, which does not perform well under noisy environment, or does not leverage the use of large data sets.

To address the limitations, in this paper, we first introduced our data collection method that can collect a large amount of data in a short period of time and also maintains the phonetic balance property. The total amount of data is 1200 hours with large variations of speakers and speaking conditions. Second, we proposed our ASR development that can take the most advantages of the collected large-scale data to train a model that is highly optimized for Vietnamese and is also robust to all major accents of Vietnam including Northern, Center, and Southern regions. In the VLSP 2018 challenge, our systems have achieved the first-place system with 6.5% WER, marking the new state-of-the-art system for Vietnamese ASR.

2. OVERALL SYSTEM

In this section, we describe the acoustic, language model training, and also the system combination strategy. An overview of our system is illustrated in Figure 1.

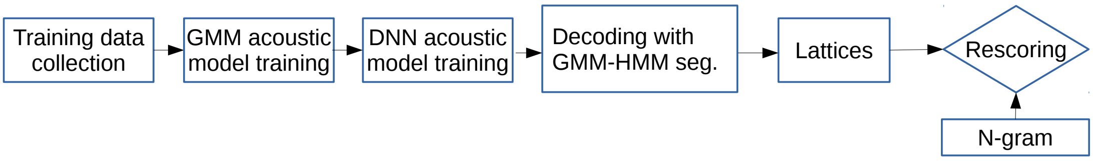

Training data collection
GMM acoustic model training
DNN acoustic model training
Decoding with GMM-HMM seg.
Lattices
Rescoring
N-gram

Figure 1. System overview

2.1. Acoustic model training

We tested several acoustic model training strategies during development. Deep neural network (DNN) acoustic models trained with 2 different types of input features were adopted, which are either standard Mel-Frequency Cepstral Coefficients (MFCC) + i-vector or FBANK + i-vector features. The DNN with standard MFCC input features is trained by the cross-entropy criterion, whereas the DNN with standard feature (MFCC, FBANK) + i-vector features is a p-norm deep neural network [16], which is trained with both cross-entropy (CE) and state-level minimum Bayes risk (sMBR) training criterion [5, 12]. The full training procedure is illustrated in Figure 2.

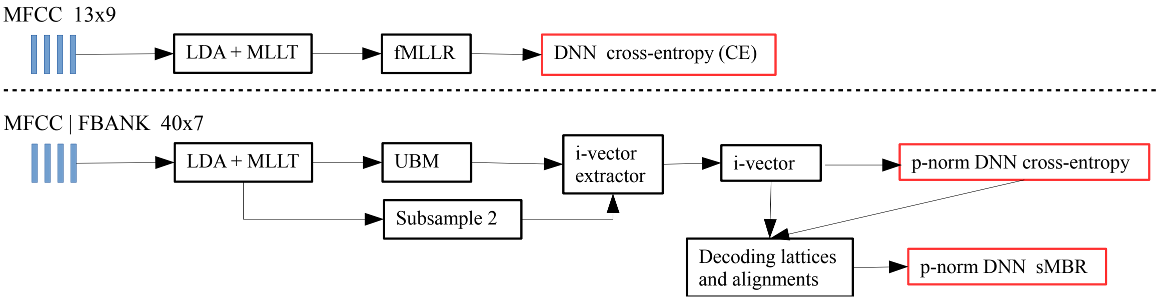


Figure 2. Acoustic model training procedure

2.1.1. Acoustic features

We utilized a combination of acoustic features including Mel-Frequency Cepstral Coefficients (MFCC) [3], pitch, and i-vector. While the MFCC is standard for speech recognition tasks, pitch is a particularly important feature for tonal languages, such as Vietnamese and Chinese. The i-vector was initially introduced for speaker recognition tasks [4], and recently has drawn researcher attention in the field of speech recognition. The i-vector $\mathbf{w}$ is defined in the context of the following term,

$$\mathbf{M} = \mathbf{m} + \mathbf{T}\mathbf{w}, \tag{1}$$

where $\mathbf{M}$ is the utterance supervector which depends on speaker and channel dependent components [8], $\mathbf{m}$ is the mean supervector of a universal background model (UBM), $\mathbf{T}$ is low rank rectangular total variability matrix, and $\mathbf{w}$ is the i-vector following the standard normal distribution $N(0, I)$.

Given input feature frames $\mathbf{Y}$, the i-vector $\mathbf{w}$ can be defined by the mean of the posterior distribution $P(\mathbf{w}|\mathbf{Y})$, where this posterior distribution is a Gaussian distribution [7].

The matrix $\mathbf{T}$, which refers to the i-vector extractor, is trained for every second feature frame to speed up the training. The i-vector extractor and the UBM models used for the experiments described in this paper were trained using 338 hours of training data including 48,587 speakers. The UBM model has 512 Gaussian components. As mentioned before, two different types of speech features (MFCC and FBANK) were utilized.

By combining several features including pitch and i-vector, the acoustic model becomes more robust against different environments and it also models better for Vietnamese, which is a tonal language.

2.1.2. DNN cross-entropy

The first DNN model can be considered a standard DNN acoustic model with 6 hidden layers, where each layer consists of 2048 nodes. The non-linear sigmoid activation function is applied in each hidden layer, and the softmax function is applied in the output layer. The input features are LDA + MLLT + fMLLR on top of MFCC. The feature frames are also spliced with 5 preceding and 5 succeeding frames, resulting in the final 440 dimensional DNN input feature vector covering 11 frames of context.

First, we performed the pre-training with deep belief network (RBM) [6]. After that, the DNN was trained using the back-propagation algorithm and stochastic gradient descent with frame cross-entropy (CE) criterion as implemented by the Kaldi speech recognition toolkit [11].

2.1.3. P-norm DNN

As a second type of model, the p-norm deep neural network [16] was adopted. The p-norm is a “dimension-reducing” non-linearity that is inspired by maxout

$$y = ||\mathbf{x}||_p = \left(\sum_i |x_i|^p\right)^{1/p}, \tag{2}$$

where the vector $\mathbf{x}$ represents a bundled set of 10 feature vectors, p is the normalized parameter and is set to 2 as it showed the best performance as described in [16]. The number of hidden layers is 6. The 40 dimensional MFCC or FBANK feature vectors and the 100 dimensional i-vectors are stacked to form a 140 dimensional DNN input feature.

The parameters are trained by using either CE or sMBR criterion as implemented in Kaldi. For each type of input features, two DNN models are trained, one by the CE criterion, the other by the sMBR criterion. After decoding, the decoding lattices of both systems are combined to produce the final decoding lattices.

Using the p-norm DNN model, we can greatly reduce the input dimension while maintaining the accuracy of the acoustic model.

2.1.4. I-vector extraction

The i-vector was initially introduced for speaker recognition tasks [4], and recently has drawn researcher attention in the field of speech recognition. The i-vector $\mathbf{w}$ is defined in the context of the following term,

$$\mathbf{M} = \mathbf{m} + \mathbf{T}\mathbf{w}, \tag{3}$$

where $\mathbf{M}$ is the utterance supervector which depends on speaker and channel dependent components [8], $\mathbf{m}$ is the mean supervector of a universal background model (UBM), $\mathbf{T}$ is low rank rectangular total variability matrix, and $\mathbf{w}$ is the i-vector following the standard normal distribution $N(0, I)$.

Given input feature frames $\mathbf{Y}$, the i-vector $\mathbf{w}$ can be defined by the mean of the posterior distribution $P(\mathbf{w}|\mathbf{Y})$, where this posterior distribution is a Gaussian distribution [7].

The matrix $\mathbf{T}$, which refers to the i-vector extractor, is trained for every second feature frame to speed up the training. The i-vector extractor and the UBM models used for the experiments described in this paper were trained using 338 hours of training data including 48,587 speakers. The UBM model has 512 Gaussian components.

2.1.5. Pronunciation dictionary

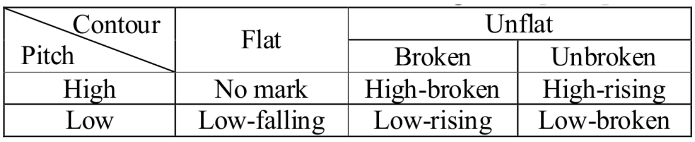
To train an acoustic model, we need to define pronunciations of all Vietnamese words. Vietnamese language is a complex language compared with other languages because it is a monosyllable language with tones, every syllable always carries a certain tone [2, 14].

<table>
<tr><td colspan="4">TONE</td></tr>
<tr><td rowspan="2">Initial</td><td colspan="3">FINAL</td></tr>
<tr><td>Onset</td><td>Nucleus</td><td>Coda</td></tr>
</table>

Figure 3. Structure of Vietnamese syllables

The structure of Vietnamese syllables is showed in Table 3. There are 22 initials and 16 finals phonemes in Vietnamese. There are 6 tones in Vietnamese. Five tones are represented by different diacritical marks such as low- falling tone, high- broken tone, low-rising tone, high-rising tone, low-broken tone. The tone called mid tone is not represented by a mark. Tones are differentiated in Table 4.

The total number of unique syllables in Vietnamese is 19000 but there are only 6500 syllables used in practice.



Contour / Pitch	Flat	Unflat	
		Broken	Unbroken
High	No mark	High-broken	High-rising
Low	Low-falling	Low-rising	Low-broken

Figure 4. Structure of Vietnamese tone

2.2. Language model

2.2.1. *N*-gram

N-grams have long been a standard language modeling technique for ASR, where $N-1$ words are used as context to predict the next word. The larger the context, the more data is required to avoid the data sparsity problem. During the experiments described here, two N-gram language models were trained with Kneser-Ney smoothing [1] implemented in SRILM language modeling [13], a 3-gram LM pruned with probability 10^{-8} for decoding purposes, and a full 4-gram model for rescoring in a second pass. All available textual training data was used for the training of these models.

Recently, Recurrent Neural Network-based Language Models (RNNLM) have been proposed and delivered promising performance over n-gram models. However, the performance improvement of RNNLM comes at a cost of computation. Another limitation of RNNLM is that it can only be used for rescoring decoding lattices. These limitations make it not possible to utilize RNNLM in a large-scale ASR system.

3. DATA COLLECTION

- To build a high-quality ASR system that can handle speech in different environment and context, it is important to understand Vietnamese phonetic structure and to train the system on a large-scale dataset.

- Vietnamese is a complex language because of its characteristics such as isolating morphology and phonemically distinctive tones. To understand Vietnamese phonetic structure, we need to collect a high quality speech corpus and analyze speech characteristics of Vietnamese from it.

- There are not many public large-scale datasets for Vietnamese, some of them are known as MICA VNSpeechCorpus, AIlab VIVOS, VOV (Radio broadcast resources). However, those corpora have neither large-scale nor high quality sound. Besides, it is hard to manually collect high quality speech corpus because of time-consuming and cost, so we need to collect automatically to reduce effort.

3.1. High quality speech collection

To have a wide variety of context, we have collected sentences from most popular electronic newspapers. However, since the data from newspapers is typically noisy(raw text input), we need to put it through a chain of data processing phases as illustrated in Figure 5. Then, we select the smallest subset of data that maintains the phonetic balance criterion.

3.1.1. Design text data

The text recording needs to be designed to meet the criteria that it is not too large but it must ensure the phonetic balance. We first collect a large amount of text from electronic newspapers. And then, process it in a chain of processing phases.

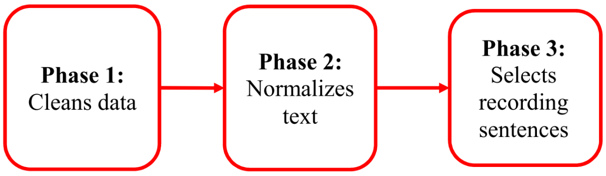


Figure 5. The process of refining and processing the recording texts

Phase 1. We downloaded 10GB raw text from electronic newspapers. All text data have the main content stored in the ⟨content⟩ tags, other information outside the tag is metadata. To reduce noises, we only select texts from the ⟨content⟩ tag. Next we cut them into small sentences based on ending punctuation such as ".", "?", "!". To further reduce noises, we remove all lines containing post time, shortcuts, address, as well as arbitrary strings (asterisks, special characters, punctuation marks, author names, annotations, quoted source names etc).

Phase 2. In this phase we have 4,000,000 sentences, the main task is to normalize the text which includes many non standard words according to the standards words in Vietnamese. Non standard words include digit sequences; numbers; abbreviations; units of measurement; roman numerals; foreign proper names and place names... We analyzed text and used the technique to transforming (or expanding) a sequence of words into a common orthographic transcription. The process is done by 2 steps:

Step 1. To reduce pronunciation ambiguity, all numbers, date, time and measure units are spelled out with the following rules:

- Number format: Numbers are transcribed in code by assigning them to arrays and transcribing them into corresponding strings (e.g. 1235 → một nghìn hai trăm ba mươi năm). Then exceptions are replaced with standard words (e.g. không mươi → lẻ, mươi năm → mươi lăm, mươi một → mươi mốt).

- Time format: Format dd/mm/yyyy is automatically transcribed into day...month...year. Format (dd/mm, dd-mm-yyyy and dd-mm) with the word 'day' standing in front is transcribed as "day", "month". Format hh:mm:ss is understood as hour, minute, second. Format hh:mm with "at" standing in front is transcribed into "hour", "minute".

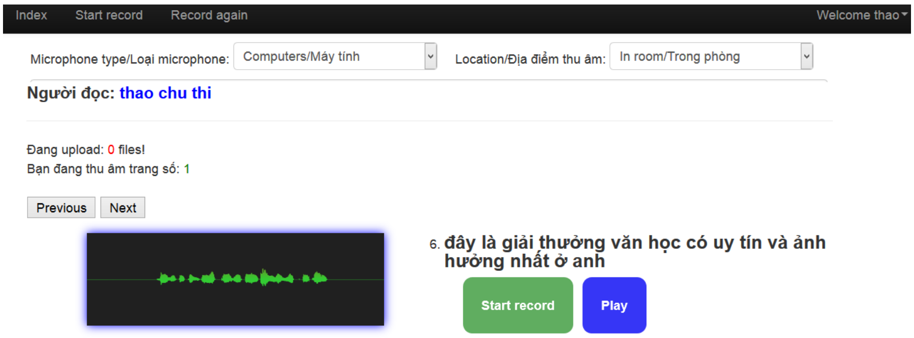
- Units of measurement: Separate alphanumeric characters with spaces (e.g. 10Kg → 10kg, 10m → 10m, 11hz → 11hz, 8/10 → 8/10, 90).Then, replace words with transcribing digits for signs or measure units (e.g. 10kg → ten kilograms, 10meters → ten meters, 11hz eleven hertz, 8/10 eight per ten, 90).

Step 2. Transcribe abbreviated acronyms or proper names with self-defined dictionaries (e.g. TP - city, HCM - Ho Chi Minh City, VND - Vietnam dong, Paris - Pa ri, Samsung - Sam Sung). After normalizing the text, we split them into small sentences and only keep ones containing minimum 40 and maximum 90 syllables. This is an appropriate length for speech recording.

Phase 3. The final step is to select a good amount of sentences for audio recording. The recording sentences should maintain the phonetic balance property and be small to reduce recording cost. We adopt text selection based on greedy search to find the optimal sentences. This step is repeated until a certain number of sentences is selected.

3.1.2. Recording

To speed up the recording, as well as make it easier and less prone to human error, we designed a recording application that runs on Firefox browser, web-based recording interface as shown in Figure 6. The speaker can listen to his/her recorded audio and can also see the audio signal to ensure that there is 1 second of silence at the beginning and ending of utterances and there is no audio clipping occurred. During a recording session, if there is any sentence which does not meet the requirement, the speaker will only need to record the sentence again. The recording process is supervised by the administrator to ensure that the recordings meet the quality requirements. With unsatisfactory sentences the administrator will judge “bad” in the “verify” section, then the recorder will open the “Recording again” window to record the unsatisfactory passages that have been filtered. Completed recording gives us the recording data that encapsulates each record into a single *.wav file and an *.info text file containing the corresponding information.



Microphone type/Loại microphone: Computers/Máy tính

Location/Địa điểm thu âm: In room/Trong phòng

Người đọc: thao chu thi

Đang upload: 0 files!

Bạn đang thu âm trang số: 1

6. đây là giải thưởng văn học có uy tín và ảnh hưởng nhất ở anh

Figure 6. Web-based recording application

The audio is recorded with a high-quality TakStar PC-K600 microphone. The audio sampling rate is 48Hhz, 16-bits PCM, and mono channel.

3.2. Large scale speech collection

The section above has described our approaches to collect a phoneme balanced corpus which is important to study about Vietnamese phoneme characteristics. However, building an ASR system based on only that data is insufficient, it does not capture real-condition environment such as low quality microphones, accents, genders, and noisy environment. Collecting a data set that reflects all above conditions with the existing approach is impractical due to cost and time limitation.

In this section, we describe another approach that allows us to collect a massive amount of data in a short period with a minimal cost and human resources. We first collect a large amount of untranscripted audio and then build a semi-automation transcription system to make the transcript for those audio.

3.2.1. Audio acquisition

To have a large variety of audio environment, we collected untranscripted audio from various resources including movies, youtube movies, and online newspaper. An automated system is constructed to assemble the resources automatically. The audio is then re-formatted to raw PCM 16 bits, sampling at 16Khz, mono channel.

3.2.2. Hybrid text transcription

Manually transcribing the large amount of un-transcripted audio above is a difficult and time-consuming task. A study conducted by https://www.livechatinc.com has showed that the average person's typing speed is 38-40 words per minute. Therefore, it is not feasible to manually transcribe the whole data set. Instead, we have built a hybrid system that allows us to automatically transcribe the audio and verify the transcription by human.

The idea is simple, we first build an ASR system with an existing data set, which we already have transcribed. Then, we generate text for the un-transcripted audio, the generated text is full of errors because the audio is recorded in different environments. To minimize the errors, manual verification and revision is required. To help the verification task easier, we adopt our web-base recording (as illustrated in Figure 6) by removing the record button and allow workers to edit the text. By doing this, we can transcribe the audio quickly and accurately with minimal amount of resources needed.

4. EXPERIMENTS

4.1. Corpora

In total, we have collected 3 corpora with the total amount of speech is approximately 900 hours including 2 corpora recorded in a control and clean environment and 1 collected at large scale from various resources over the Internet. The detail of those corpora is showed in Table 1.

Table 1. Statistics of text sentences for recording

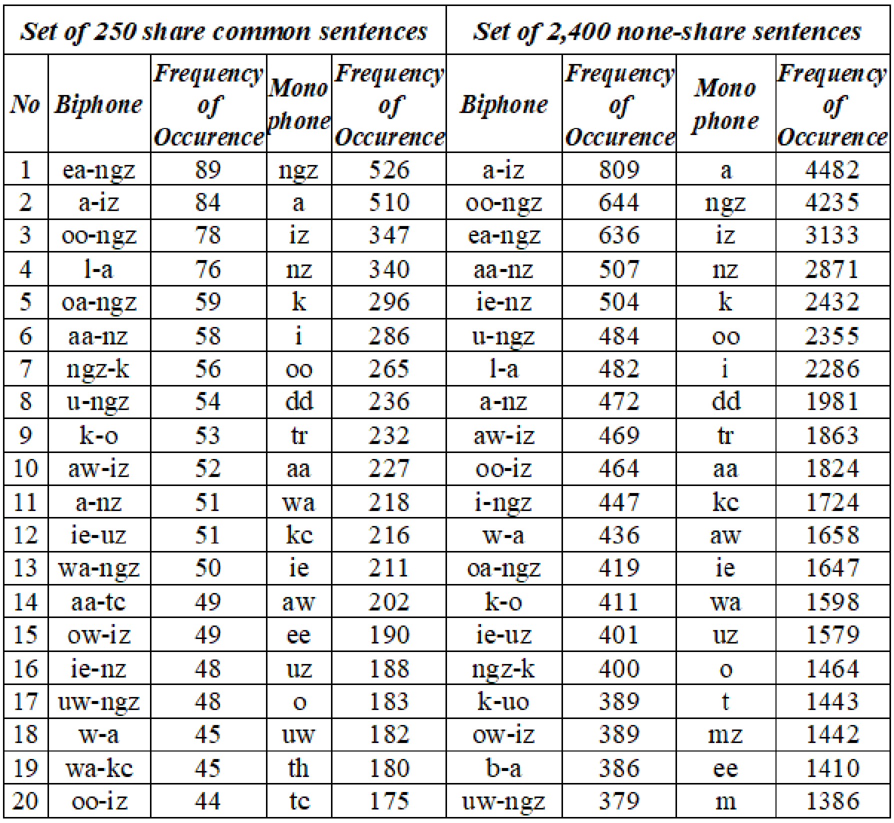
Data set	**Number of hours**
5400 sentences	6
6000 sentences	6.5
1.3M sentences	900

4.2. Data analyses

In this experiment, we analyze various aspects of Vietnamese speech characteristics based on our data. To eliminate external effects such as environment, microphone quality, we only assert the 6000 sentences corpus which is recorded in a clean environment with a high quality microphone.

4.2.1. Phonemes statistic

To evaluate data corpus, we use several modules to count two text data sets based on occurrence frequency and deference of phonemes, syllables and words. The results are shown in Table 7.



Set of 250 share common sentences					*Set of 2,400 none-share sentences*			
No	***Biphone***	***Frequency of Occurence***	***Mono phone***	***Frequency of Occurence***	***Biphone***	***Frequency of Occurence***	***Mono phone***	***Frequency of Occurence***
1	ea-ngz	89	ngz	526	a-iz	809	a	4482
2	a-iz	84	a	510	oo-ngz	644	ngz	4235
3	oo-ngz	78	iz	347	ea-ngz	636	iz	3133
4	l-a	76	nz	340	aa-nz	507	nz	2871
5	oa-ngz	59	k	296	ie-nz	504	k	2432
6	aa-nz	58	i	286	u-ngz	484	oo	2355
7	ngz-k	56	oo	265	l-a	482	i	2286
8	u-ngz	54	dd	236	a-nz	472	dd	1981
9	k-o	53	tr	232	aw-iz	469	tr	1863
10	aw-iz	52	aa	227	oo-iz	464	aa	1824
11	a-nz	51	wa	218	i-ngz	447	kc	1724
12	ie-uz	51	kc	216	w-a	436	aw	1658
13	wa-ngz	50	ie	211	oa-ngz	419	ie	1647
14	aa-tc	49	aw	202	k-o	411	wa	1598
15	ow-iz	49	ee	190	ie-uz	401	uz	1579
16	ie-nz	48	uz	188	ngz-k	400	o	1464
17	uw-ngz	48	o	183	k-uo	389	t	1443
18	w-a	45	uw	182	ow-iz	389	mz	1442
19	wa-kc	45	th	180	b-a	386	ee	1410
20	oo-iz	44	tc	175	uw-ngz	379	m	1386

Figure 7. Statistics of 20 most popular phonemes in 2 data set (without sil)

4.2.2. Sound quality analysis

The sound quality was analyzed by Praat v6.0 software to evaluate the characteristics of sound waves, spectra, pitch and sound intensity. The following example in Figure 8 will analyze the waveforms and spectrograms of a random utterance recorded.

Data evaluation was done through the assessment of the recording environment, the noise ratio [14]. Through analysis and evaluation of all data, the recording was evaluated to be of

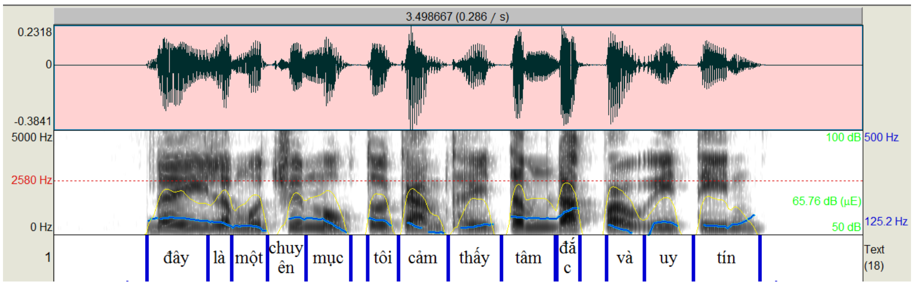


Figure 8. Waveform and spectrogram of the female voice

good quality with clear sound and little noise.

4.2.3. Duration analysis

In this experiment, we are interested in the difference between genders and ages in terms of duration of words. To obtain word duration, we build an automatic speech recognition (ASR) using Kaldi toolkit. The training for the ASR system is the same data used for the decoding process so that we can have accurate audio alignment results. State duration of each HMM is modeled by a multivariate Gaussian estimated from histograms of state duration which were obtained by the Viterbi segmentation of training data.

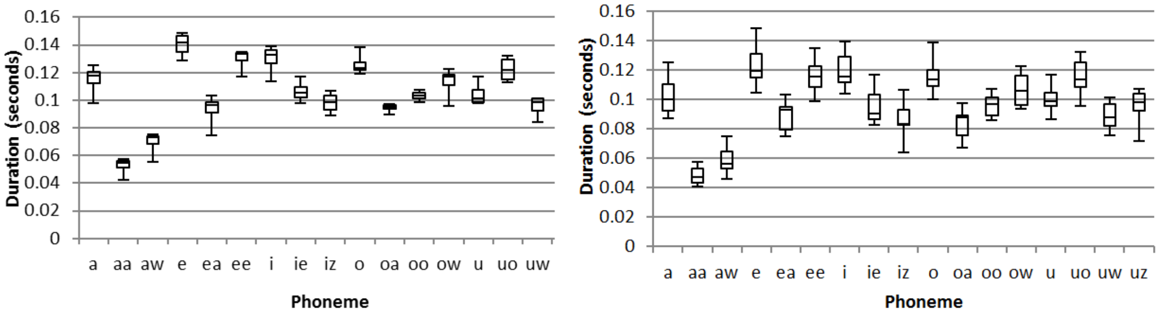


Figure 9. Duration distributions of vowels spoken by female voices at the same age and by people with wide range of age and different gender

The left Figure 9 shows vowel duration distributions of all female speakers at the same age. As we can see, the ranges of all distributions are quite small, indicating that females at the same age tend to have similar reading speed. On the other hand, the right Figure 9 shows the duration distributions of speakers with different genders and wide range of age. We can clearly see that the distribution is larger than it is in the left one.

4.3. Language model evaluation

The training data for language modeling is collected from various sources including websites and movie subtitles. A total of 6.5 million sentences have been collected. We extracted a 10000 held-out sentences for evaluation and the rest is used for training the models.

We used SRILM toolkit for modeling training and used perplexity for evaluating the performance. Various n-gram models have been conducted including 3-gram, 4-gram, and their pruned versions. The results are showed in Table 2.

Generally speaking, the higher the n gram, the better perplexity we have. From the table, we can see that the 4-gram has the best perplexity, however, its size is almost doubled compared with the 3-gram model. On the other hand, the 4-gram pruned has the size of only 18 MB, with relatively good perplexity.

In real application, it is a good practice to use as small language model as possible to limit the memory usage of the ASR engine. For this reason, we use the 4-gram pruned model for online decoding and then, we rescore the output lattices with the 4-gram language model to achieve the best performance while reduce the memory usage.

Table 2. Language model evaluation

Model	**Perplexity**	Size (MB)
3-gram	82.47	43 MB
4-gram	56.29	99 MB
3-gram pruned	114.93	12 MB
4-gram pruned	93.23	18 MB

4.4. Acoustic model evaluation

We built several systems for the evaluation, featuring various front-ends, acoustic model types and training criteria. The training data for acoustic modeling was fixed on one subset of the provided material prior to system development. Likewise, all systems are based on the same phoneme set and language model training data.

Two test sets were used through out the evaluation, one is clean and one is noisy data. Each test set has approximately 10.000 sentences.

4.4.1. The effect of pitch on Vietnamese ASR system

In this experiment, we evaluate the effect of pitch on Vietnamese ASR. We train 2 systems with the exact configuration except one with pitch and one without pitch. To reduce the training time for this experiment, we used a subset of the training data of 200 hours. The result is showed in Table 3.

Table 3. ASR performance comparison between systems trained with and without pitch features

Pitch	**WER on clean test set**	**WER on noisy test set**
no	11.42	35.64
yes	10.38	32.51

As we can see, pitch features help to improve the performance with significant margin. WER reduced by 1% on the clean data set and 3% on the noisy data set.

4.4.2. Full system evaluation

After training a model to evaluate the effect of pitch, we build a complete system that utilizes all data techniques described above. To have a system that is robust against different speaking rate and to reduce over-fitting of the model, we triple the data into 3 parts including slow, normal, and fast speaking rate. And then, adding some noise into them so that the model does not get over-fitted.

In addition, we also apply CMVN to normalize the input features so that it is more robust against different ages and speaking styles. Usually, CMVN is only applied to GMM-HMM model and for offline decoding. In our system, to make it possible for online decoding, we apply sliding windows CMVN where statistic information is calculated with a sliding windows of 600 frames.

The result is showed in Table 4. As we can see by increasing the training data we can reduce WER by approximately 1% (the 1st row) on the clean dataset. We hypothesize that with 200 hours of training data used in the previous experiment, the model is able to capture most of acoustic information in clean environments, so that increasing the data does not help too much. However, the WER on the noisy set get a huge improvement by 14% absolute. By tripling the data set and utilizing CMVN normalization, we were able to reduce the WER by 1% on the clean and 3% on the noisy set. Moreover, rescoring the best system with the 4-gram language model reduced WER by 1.5%, yielding the best system with WER of 7%.

To compare our system with other teams, we participated in VLSP challenge 2018[1] to evaluate our system on a 2-hour test set collected with various accents and genders. Our system outperforms all other systems to yield the new state-of-the-art Vietnamese ASR performance with 6.29% of WER.

Table 4. ASR performance

Speed perturbation	**Pitch**	**CMVN**	**WER on clean set**	**WER on noisy set**	**WER on VLSP 2018**
no	yes	no	10.62	21.34	-
yes	yes	no	9.60	18.42	-
yes	yes	yes	**9.51**	**17.39**	6.29

5. CONCLUSIONS

This paper described the structure and development of VAIS's Vietnamese ASR systems along with collection of large speech corpora. We have collected three corpora with different recording environments and devices. The corpora are used for two purposes including analyzing Vietnamese speech properties and constructing Vietnamese ASR models.

With regards to data analysis, we found that (1) phonemes in Vietnamese are spoken with a relatively short duration, ranging from 0.04s to 0.15s. This is an important observation to help us deciding the number of HMM states. And (2), we also observed that people in the same age and gender tend to have similar reading speed.

[1] http://vlsp.org.vn/

Regarding the construction Vietnamese ASR models, we have applied various techniques including pitch, speed perturbation, n-gram language modeling and deep neural network models to construct the most state-of-the-art ASR system for Vietnamese. Our system performs significantly better than other competitors in the VLSP 2018 challenges with WER of only 6.29%.

Future works will include building ASR models that are robust on different accents of Vietnam and optimizing the pronunciation dictionary.

REFERENCES

[1] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Proceedings of ACL*, 1996, pp. 310–318.

[2] M. Chu, *Fundamentals of Linguistics and Vietnamese language* (in Vietnamese). Education Publishing House in Hanoi, 1997.

[3] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE*, pp. 357–366, 1980.

[4] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front end factor analysis for speaker verification," *IEEE*, 2010.

[5] M. Gibson and T. Hain, "Hypothesis spaces for minimum bayes risk training in large vocabulary speech recognition," in *Proceedings of INTERSPEECH*, 2006.

[6] G. Hinton, "A practical guide to training restricted boltzmann machines," *Momentum*, vol. 9, no. 1, p. 926, 2010.

[7] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *IEEE*, vol. 13, no. 3, pp. 345–354, May 2005.

[8] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE*, vol. 15, no. 4, pp. 1435–1447, May 2007.

[9] Q. Nguyen, T. Vu, and C. Luong, "Improving acoustic model for vietnamese large vocabulary continuous speech recognition system using tonal feature as input of deep neural network," *Journal of Computer Science and Cybernetics*, vol. 30, pp. 28–38, 2014.

[10] V. Nguyen, C. Luong, T. Vu, and Q. Do, "Vietnamese recognition using tonal phoneme based on multi space distribution," *Journal of Computer Science and Cybernetics*, vol. 30, pp. 28–38, 2014.

[11] D. Povey, A. Ghoshal, G. Boulianne, N. Goel, M. Hannemann, Y. Qian, P. Schwarz, and G. Stemmer, "The Kaldi speech recognition toolkit," in *Proceedings of IEEE workshop*, 2011.

[12] D. Povey and B. Kingsbury, "Evaluation of proposed modifications to MPE for large scale discriminative training," in *Proceedings of ICASSP*, vol. 4, April 2007, pp. IV–321–IV–324.

[13] A. Stolcke, "SRILM – an extensible language modeling toolkit," in *Proceedings of ICSLP*, vol. 2, Denver, USA, 2002, pp. 901–904.

[14] V. Thang, L. C. Mai, and S. Nakamura, "An HMM-based vietnamese speech synthesis system," in *Proceedings of O-COCOSDA*, 2009.

[15] T. Vu, T. Nguyen, C. Luong, and J. Hosom, "Vietnamese large vocabulary continuous speech recognition," in *Interspeech*, 2005.

[16] X. Z., J. Trmal, D. Povey, and S. Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *Proceedings of ICASSP*, May 2014, pp. 215–219.

Received on October 04, 2018
Revised on December 28, 2018