International Journal of Multidisciplinary Research and Growth Evaluation

International Journal of Multidisciplinary Research and Growth Evaluation
ISSN: 2582-7138
Received: 28-05-2021; Accepted: 17-06-2021
www.allmultidisciplinaryjournal.com
Volume 2; Issue 4; July-August 2021; Page No. 152-156

A method for speech Vietnamese recognition based on deep learning

Thanh T Nguyen [1*], Binh A Nguyen [2], Manh Hoang [3], Tung V Nguyen [4], Giao N Pham [5]
[1*] Ha Noi University of Science and Technology, Hanoi, Vietnam
[2-4] ICT Department, FPT University, Hanoi, Vietnam
[5] Department of Computing Fundamentals, FPT University, Hanoi, Vietnam

Corresponding Author: **Thanh T Nguyen**

Abstract

Speech recognition has been increasingly applied in various fields such as automatic switchboards, security, searching by voice and so on. However, the quality of recognition is the problem of utmost concern. This paper describes the Speech Vietnamese recognition system built on Kaldi toolkit. The paper also evaluates the quality of system based on evaluating the ratio of the WER on acoustic models. The proposed system achieved superior results compared to previous toolkits on the Vietnamese speech.

Keywords: Speech recognition; speech Vietnamese; Kaldi toolkit; Acoustic Modelling, Pronounce dictionary

1. Introduction

In this paper, The Kaldi toolkit has selected because the main advantages are modern, flexible code, clearly structed code. And moreover, Kaldi gives higher quality on the recognition of other toolkit such as HTK, Sphinx or Alize. Christian Gaida et al. [1] proposed a large scale evaluation of open-source speech recognition toolkits. Authors adjusted the systems and test on the German and English. Laboratory results showed that Kaldi outperforms all the other recognition toolkits, providing training and decoding pipelines including the most advanced techniques. This conveniently enables the best result in short time.

The time spend on setting up, preparing, running and optimizing the toolkits was most for HTK, less Sphinx and least Kaldi. The toolkit of the Sphinx family comes up with a training tool also, not containing all techniques of Kaldi leading to less accuracy. HTK is the most difficult toolkit, although the obtained results are similar to Sphinx, however; system settings need to take time. Compared to the other recognition toolkits, Kaldi's outstanding performance is seen as a revolution in speech recognition technologies open-source.

Currently there are some researches on Vietnamese speech recognition, however almost uses HTK toolkit only [2]. Therefore, research objective of this paper are building speech Vietnamese recognition toolkit used Kaldi toolkit, testing advanced techniques in Kaldi to evaluate Kaldi's ability to Vietnamese. The next section of the paper will introduce the speech recognition Kaldi toolkit, Part III describes building methods speech Vietnamese recognition using Kaldi toolkit and optimization solutions for system. Part IV are the conclusion and subsequent development.

2. Introduction the speech recognition Kaldi toolkit

A. Introduction the speech recognition Kaldi toolkit

Kaldi is an open-source toolkit for speech recognition written in C++ and licensed under the Apache License v2.0 [3]. Kaldi is designed for speech recognition researchers. Compared to other speech recognition toolkits, Kaldi is similar of aims and scope to HTK. The goal is to have modern and flexible code, written in C++, that is easy to modify and extend. Important features include: code-level integration with Finite State Transducers (FSTs); extensive linear algebra suppport include a matrix library that wraps standard BLAS and LAPACK routines; extensible design; the decoder could work from any suitable source of scores, such as a neural net; open license allows convenient use.

B. Kaldi toolkit structure

Kaldi include a library, the command-line programs and scripts for acoustic model. Kaldi deploy multiple decoders to evaluate the acoustic models, using the Viterbi training for estimating the acoustic models. Only in special cases speaker adaptive discriminative training extended using Baum-Welsh algorithm. The architectures of Kaldi toolkit can be separated into Kaldi library and training scripts.

These scripts access the Kaldi library's function through the command-line program. Kaldi C++ library is based on the library OpenFST [4]. These functions are relevant to each other and often grouped in a domain in C++ code, which corresponds to one directory on file system. The examples of the namespaces or directories can be seen in Figure 2.

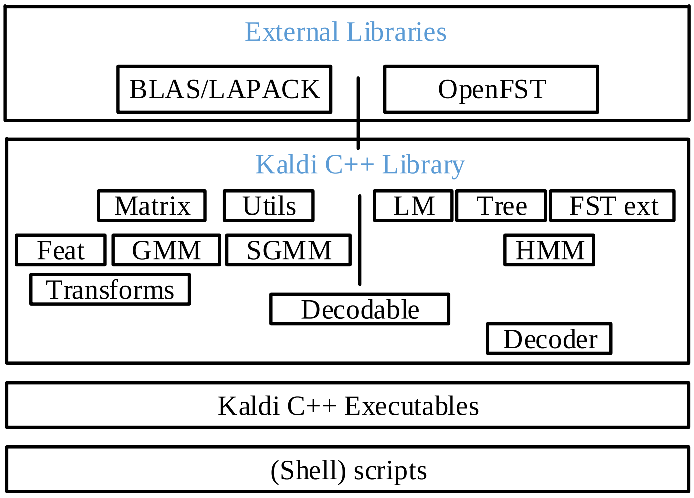

Fig 2: Kaldi toolkit architecture

3. Speech Vietnamese recognition using Kaldi toolkit

A. Speech Vietnamese recognition model used Kaldi toolkit

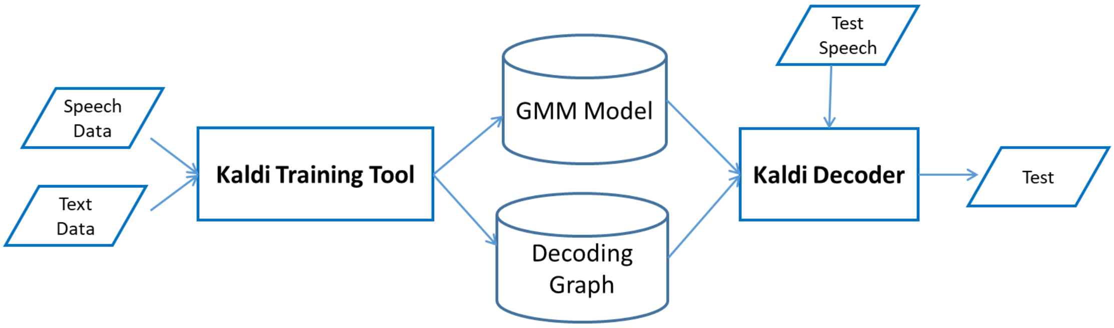

Fig 3: Speech Vietnamese recognition model used Kaldi toolkit

Overview schematic of Speech Vietnamese recognition model use Kaldi toolkit is described in figure 3. In this model, acoustic modelling (AM) is arguably the heart of speech recognition. The AM estimates the probability P*(a/w;θ),* this value is used in the speech recognition by the equation (1).

$$W^* = argmax_w \left\{ \frac{P(a|W) * P(W)}{P(a)} \right\} \\ = argmax_w \{P(a|W) * P(W\} \qquad (1)$$

Acoustic model has only partial information available for training AM parameters θ because the corresponding textual transcription is time-unaligned. The hidden information of the word (time) alignment in a utterance makes acoustic model training more challenging. Modern speech recognition toolkits use Hidden Markov Model for modelling uncertainty between acoustic features and the corresponding transcription.

B. Speech Vietnamese data

Data is recorded by 35 people (16 men and 19 women) aged 17-29 years old. Data were recorded on topic including: life, business, science, automobile-motorcycle and laws. Voices were recorded in the form of reading, were recorded in the normal working environment, recorded at the sampling frequency 16KHZ, 16 bits per sample, mono mode. The data were divided into two parts: one part to the training and the secon part to test. Details about the data decribed in Table 1.

Table 1: Speech Vietnamese Data

Data	Gender speaker		Record (hour)	Total sentence
	Male	Female		
Training	12	15	6	3.375
Test	4	4	2	1.000
Total	16	19	8	4.375

C. Text corpus

The text corpus is used to create statistical language models. The file consists of 4 million sentences with 90 million syllables collected from Vietnamese electronic documents. The characters are converted to Bach-Khoa Text Code (BKTC) [2]. The perplexity of bigram LM and trigram LM is 108.57, and 62.43 respectively. We used SRILM toolkit on the text corpus to create language models in ARPA format. The bigram language model contains 8925 unigrams and 3,742,980 bigrams. The trigram language model has all grams in bigram LM and 11,593,319 trigrams. The files are used to create LM in FST file format.

D. Acoustic model scripts

The recordings and their transcriptions from training dataset are used for acoustic modelling. The estimated AMs are evaluated on the test set. The decoding of the test utterances is performed always with the same parameters, so that different AMs can be compared. The Table 2 lists all acoustic models trained in scripts. An advanced AM is always initiated by audio alignments (respectively acoustic features alignments) using a simpler AM.

The used methods are listed in Figure 4 together with their hierarchy. The hierarchy shows that a more advanced method typically reuses initial values from previously trained simpler AM.

At first, a mono-phone model is trained from flat start using the MFCCs (Mel Frequency Cepstral Coefficient), Δ and ΔΔ features. The feature vectors are aligned to HMM states using utterance's transcriptions. Secondly, we retrain the tri-phone AM (*tri1*). One branch of experiments finishes by training MFCC Δ + ΔΔ tri-phone AM (*tri2a*). On the other hand, the second branch instead of Δ + ΔΔ transformation uses LDA+MLLT to train AM (*tri2b*). Using the AM *tri2b* three AMs are discriminatively trained and use LDA+MLLT+SAT to train tri3, using the following objective functions:

- MMI (Maximum Mutual Information). [6]
- BMMI (Boosted Maximum Mutual Information). [7]
- MPE (Minimum Phone Error). [8]
- SAT (speaker adaptive training). [9]

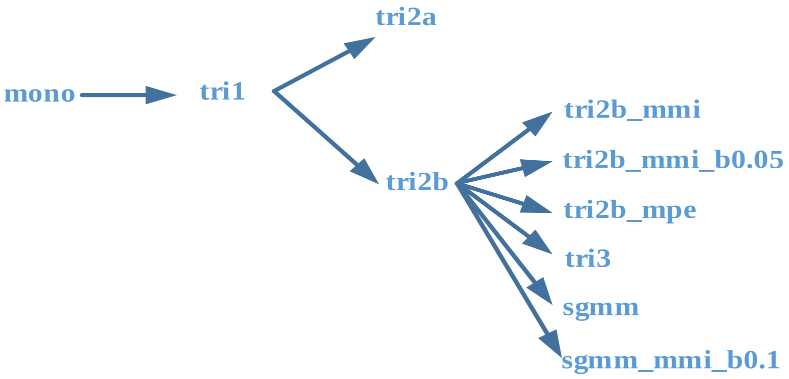

Fig 4: Hierarchy of acoustic model training

Table 2: The training models of system

Train models	Description
Mono phone	Mono
Triphone	Tri1
Δ + ΔΔ	Tri2a
LDA + MLLT	Tri2b
LDA + MLLT + MMI	Tri2b_mmi
LDA + MLLT + bMMI	Tri2b_mmi_b0.05
MPE	Tri2b_mpe
LDA + MLLT + SAT	Tri3
SGMM	Sgmm
SGMM+bMMI	Sgmm_mmi_b0.1

E. GMM models

Kaldi support GMMs [10] with diagonal and full convariance structures. Rather than representing individual Gaussian densities separately, Kaldi directly implement a GMM class that is parametrized by the *natural parameters*. The GMM classes also store the *constant* term in likelihood computation, which consist of all the terms that do not depend on the data vector. Such an implementation is suitable for efficient log-likelihood computation with simple dot-products.

A Gaussian mixture model (GMM) represents features as the weighted sum of multiple Gaussian distributions. Each Gaussian state i has a: Mean (µ*i*), Covariance (*Σi*), Weight (*Wi*). During the training process, system learn about datas which it uses to make decisions. A set of parameters that is collected from a speaker (or language or dialect).

Instead of training speaker model on only speaker data, adapt the UBM to that speaker takes advantage of all the data, MAP adaptation: new mean of each Gaussian is a weighted mix of the UBM and the speaker, Weigh the speaker more if we have more data: $\mu_i = \alpha E_i(x) + (1-\alpha)\ \mu_i$, α=n/(n+16).

Features are normal MFCC can use more dimensions (20 + deltas). UBM background model: 512–2048 mixtures, speaker's GMM: 64–256 mixtures, often combined with other classifiers in mixture-of-experts

F. Building the decoding graph

A decoding graph is a graph represented as an OpenFst object. It stores all language model information and part of information for acoustic modelling. The decoding graph is necessary for decoding with Kaldi decoders [11]. In paper, building the *HCLG* graph using standard OpenFst operations which are implemented in Kaldi utilities. Designed scripts so they automatically update newly built AMs and LMs and create all files necessary for decoding.

The HCLG build script requires:

- Language Model
- Acoustic Model
- Acoustic phonetic decision tree
- Phonetic dictionary

In addition to building HCLG, the script also copies necessary files for decoding from AM and the HCLG graph to one directory. To sum up, following files are necessary for decoding with Kaldi decoders:

- Decoding graph HCLG,
- Acoustic Model,
- A matrix which defines feature transformations,
- A configuration file for speech parameterization and feature transformations with the same settings as used for AM training,
- A Word Symbol Table (WST)-a file containing mapping between integer labels.

G. Kaldi decoder

In the Kaldi toolkit [12] there is no single "canonical" decoder, or a fixed interface that decoders must satisfy. There are currently two decoders available: *Simple Decoder* and *Faster Decoder;* and there are also lattice-generating versions of these. By "decoder" we mean the internal code of the decoder; there are command-line programs that wrap these decoders so that they can decode particular types of model (e.g. GMMs), or with particular special conditions. Examples of command-line programs that decode are gmm-decode-simple, gmm-decode-faster, gmm-decode-kaldi, and gmm-decode-faster-fmllr.

H. Decoding setup

First, the Δ + ΔΔ triples the number of 13 MFCC features by computing the first and the second derivatives from MFCC coefficients. The computation of MFCC coefficients with the derivatives produce 39 features per frame in total.

Second, the combination of LDA and MLLT is computed from 9 spliced frames consisting of 13 MFCC features. The default context window of 9 frames takes current frame, four frames from the left context and four frames from the right context. The LDA and MLLT feature transformation gains substantial improvement over Δ + ΔΔ transformation. See Figure 5.

Using the trained AMs described above for decoding the utterances from the test dataset. For each trained AM we use the same speech parametrization and feature transformation method as was used for the given AM at training time. We experiment with all trained AMs with both zero gram and bigram LM.

The default bigram and zero gram LMs for are built from orthographic transcriptions. The bigram LM is estimated from the training data transcriptions. Consequently, in a test set appear unknown words, so called Out of Vocabulary zero gram Word. The zero gram is extracted from a test set transcriptions. The zero gram is a list of words with probabilities uniformly distributed, so it helps decoding just by limiting the vocabulary size. The bigram LM contains 1075 unigrams and 3517 bigrams for Vietnamese. The zero gram
language model is limited to 1076 words for Vietnamese.

The speech recognition parameters are set to default values; the exceptions are decoding parameters: beam=12.0, lattice-beam=6.0, max-active-states=14000 and Language Model Weight. The LMW parameter sets the weight of a LM, i.e., it regulates how much the LM is used to help AM in decoding. The LMW value is estimated on the development set and the best value is used for decoding on the test dataset.

The *gmm-latgen-faster* decoder is used for the evaluation on testing data. It generates a word level lattice for each utterance and the one-best hypothesis is extracted from the decoded lattice and evaluated by WER (Word Error Rate) and SER (Sentence Error Rate).

4. Experimental Results

AMs mono, tri1, tri2a, tri2b are trained generative. The models tri2b_mmi, tri2b_mmi_b0.05, tri2b_mpe, tri3, sgmm, sgmm_mmi_b0.1 are trained discriminatively in four iterations. The discriminative models yield better results than generative models, see Figure 5.

A. The results of implementing the training models

This section presents the result of testing speech Vietnamese recognition system with different acoustic modelling. Table 3 shows the results of the AMs. The chart represents the WER through training models shown in figure 5. The result showed that the discriminative training methods clearly outperformed the generative AMs, and also the LDA+MLLT is more effective feature transformation than using Δ + ΔΔ features. On the other hand, there are subtle differences among the three discriminatively trained AM (tri3, sgmm, sgmm_mmi_b0.1) in terms of performance.

Table 3: WER and SER for training methods

Model	% WER	% SER
mono	4.34	53.4
tri1	1.95	37.4
tri2a	1.98	37.6
tri2b	1.89	36.2
tri2b_mmi	1.76	34
tri2b_mmi_b0.05	1.75	33.8
tri2b_mpe	1.83	35.5
tri3	1.48	30.4
sgmm	1.1	23.7
sgmm_mmi_b0.1	1.09	23.5

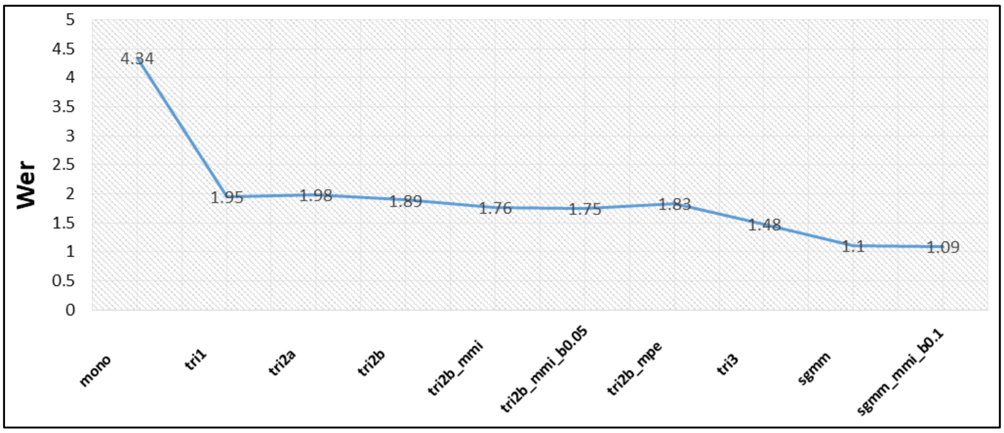

Fig 5: The WER chart reflect training models

B. The results of implementing the language model weight (LMW)

Tested with LMW respectively by 9,10 and 15. The results are described in table 4 and figure 6. The results show that the feature LMW = 15 result outperformed LMW = 9. Thus, choosing s suitable weight for the language model is also one of the important features of speech Vietnamese recognition system.

Table 4: The table results with LMWS.

Models	WER (LMW=9)	WER (LMW=10)	WER (LMW=15)
mono	68.84	8.09	4.34
tri1	42.49	3.42	1.95
tri2a	42.76	3.55	1.98
tri2b	31.55	3.14	1.89
tri2b_mmi	33.51	2.87	1.76
tri2b_mmi_b0.05	32.92	2.81	1.75
tri2b_mpe	30.1	2.96	1.83
tri3	19.07	2.22	1.48
sgmm2	13.4	1.44	1.16
sgmm2_mmi_b0.1	11.94	1.35	1.15

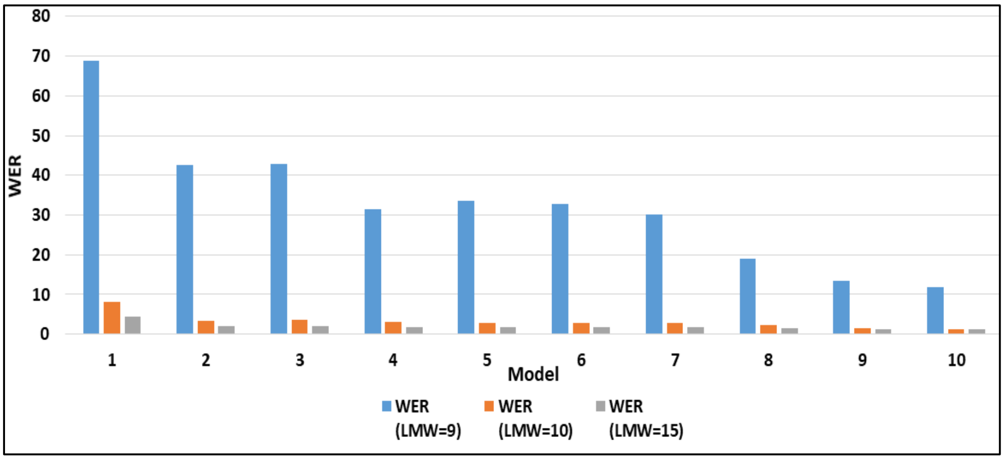

Fig 6: The WER chart with LMWs

5. Conclusions

This paper describes building methods speech Vietnamese recognition system using Kaldi toolkit. We have tested the different training methods supported by Kaldi. The language model weights are also considered and evaluated. The test showed that Kaldi toolkit for recognition results very well with Vietnamese. In addition, the weight of the language models is an important parameter when building system.

6. References

1. Christian Gaida, Patrick Lange, Rico Petrick, Patrick Proba, Ahmed Malatawy, David Suendermann-Oeft, Comparing Open-Source Speech Recognition Toolkits.
2. Nguyen Hong Quang, Trinh Van Loan, Le The Dat, Automatic Speech Recognition for Vietnamese using HTK system, IEEE-RIVF 2010, Ha noi, 2010.
3. Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, *et al*. The Kaldi Speech Recognition Toolkit.
4. KyleGorman http://www.openfst.org/twiki/bin/view/FST/WebHome, 2016.
5. Daniel Povey, Brian Kingsbury, Lidia Mangu, George Saon, Hagen Soltau, Geoffrey Zweig, fMPE: Discriminatively Trained Features for Speech Recognition, ICASSP, 2005.
6. Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahrmani, Vimal Manohar, Xingyu Na, *et al*. Purely sequence-trained neural networks for ASR based on lattice-free MMI, Inter speech, 2016.
7. Daniel Povey, Dimitri Kanevsky, Brian Kingsbury, Bhuvana Ramabhadran, George Saon & Karthik Visweswariah, Boosted MMI for Model and Feature Space Discriminative Training, ICASSP, 2008.
8. Daniel Povey, Brian Kingsbury. Evaluation of Proposed Modifications to MPE for Large Scale Discriminative Training, ICASSP, 2007.
9. Yajie Miao, Hao Zhang, Florian Metze Language Technologies Institute, Towards Speaker Adaptive Training of Deep Neural Network Acoustic Models, School of Computer Science, Carnegie Mellon University Pittsburgh, PA, USA.
10. Daniel Povey, Lukas Burget, Mohit Agarwal, Pinar Akyazi, Kai Feng, Arnab Ghoshal, *et al*. Subspace gaussian mixture models for speech recognition.
11. Daniel Povey, Partner http://kaldi-asr.org/doc/graph.html” Generated on Wed Aug 10 2016 for Kaldi by Doxygen 1. 8. 1. 2.
12. Daniel Povey and Partner http://kaldi-asr.org/doc/decoders.html Generated on Wed Aug 10 2016 for Kaldi by Doxygen 1.8.1.2 .