AN EMPIRICAL STUDY OF MULTIPASS DECODING FOR VIETNAMESE LVCSR

Khoa TRINH, Ha NGUYEN, Duc DUONG, Quan VU

Faculty of Information Technology, University of Science at VNU-HCM
227 Nguyen Van Cu, Dist 5., HoChiMinh city, Vietnam
email: vhquan@fit.hcmuns.edu.vn

ABSTRACT

In this paper, we represent an empirical study of multipass decoding for Vietnamese LVCSR. We report our experiments with N-best, lattice and consensus decoding on the VNBN data. Results from this study indicate that our acoustic model for Vietnamese was precise. The results could be investigated in further steps to improve the performance of our system.

Index Terms Vietnamese, Acoustic Model, Language Model, N-best, Word Lattice, Confusion Network.

1. INTRODUCTION

Large vocabulary continuous speech recognition (LVCSR) is both a pattern recognition and search problem. As described in [4], the acoustic and language models are built upon a statistical pattern recognition framework. In speech recognition, making a search decision is also referred to as decoding. The decoding process should take into account all available knowledge sources when hypothesising an utterance. Besides the speech signal, and the models of the recognition units, also knowledge about syntax, semantic, and other properties of the natural language might be used when searching for the most likely word sequence. One way to include these knowledge sources in the search process is to use them simultaneously to constrain a single search. Since many of the natural language knowledge sources contain "long-distance" effects, the search can become quite complex. Furthermore, the common left-to-right search strategy requires that also all knowledge sources are formulated in a predictive, left-to-right manner, which restricts the type of knowledge that can be used. One way to solve these problems is to apply sources not simultaneously but sequentially so that the search for the most likely hypothesis is constrained progressively. Thus the advantages provided by a knowledge source can be traded-off against the costs of applying it. First, the most powerful and cheapest knowledge sources are applied to generate a list of the top *N* hypotheses or word lattices (or just lattice in this paper). Then, these hypotheses are evaluated by means of the other more expensive knowledge source so that the list of hypotheses can be reordered according to a more refined likelihood score. More recently, a new class of normalized lattices have been proposed. These lattices (a.k.a. word confusion networks, (WCN)) are more efficient than traditional lattices, in terms of size and structure, without compromising recognition accuracy. They also provide an alignment for all the strings in the lattices.

In this paper, we report our experimental results with multipass search for Vietnamese LVCSR on the VNBN corpus [1-3], including N-best list, lattice and confusion network decoding. Those results could be further used in post-processing phrases to improve the performance of the system, such as rescoring with long-span language models, N-best re-ranking or even with spoken language translation. We organize the paper as follows. In Section 2, we briefly review some aspects of the multipass decoding, which are typically exploited in the current LVCSR systems. In section 3, we give a short introduction to the Vietnamese LVCSR system, which has been developed at the AI-LAB, VNU-HCM. The details of this part can be found in [1-3]. All the experiments with multipass decoding are carefully represented in Section 4. Finally, Section 5 ends up with some notes and remarks.

2. MULTIPASS DECODING IN AN LVCSR SYSTEM

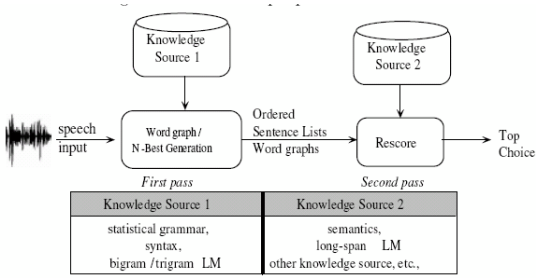

Figure 1 The multipass decoding framework

Fig.1 illustrated the multipass search paradigm. In the initial pass, the most discriminant and computationally affordable knowledge sources (KS) are used to reduce the number of hypotheses. In subsequent passes, progressively reduced sets of hypotheses are examined, and more powerful and expensive KSs are then used until the optimal solution is found. In this section we will review N-best, lattice and confusion network algorithms.

2.1. N-Best Algorithms

Different algorithms for finding the N-best sentence hypotheses have been proposed in [5]. Some of these algorithms are exact while others use different approximations to reduce computational requirements. Basically, the Viterbi algorithm typically used in an HMM-based speech recognizer only finds the best word sequence (corresponding to the state sequence with the highest likelihood score). To obtain not only the first best hypothesis but the list of the best N-hypotheses, several modifications of the Viterbi algorithm are necessary. In this study, we used the algorithm for finding the N-best hypotheses, proposed by [6], which is based on the lattice. The principle of the approach is based on the following considerations: When several paths lead to the same node in the lattice, according to the Viterbi criterion, only the best scored path is expanded. The remaining paths are not considered for further expansions. Assuming that the first best sentence hypothesis was found by the Viterbi decoding through a given lattice, the second best path is the path which competed with the best one but was recombined at some node of the best path. Thus in order to find the second best sentence hypothesis, we have to consider all possible partial paths in the lattice which reach some node of the best path and might share the remaining section with the best path. By applying this procedure repeatedly, N-best sentence hypotheses can be successively extracted from a given lattice. Thus it is important to point out that this algorithm performs a full search within the lattice and delivers exact results as defined by the lattice structure.

2.2. Word Lattice Algorithms

The main obstacle of N-best method is that the number of sentences needed to include the correct hypothesis grows exponentially with the length of the utterance. In order to find a way to compactly represent the alternative hypotheses, word lattices were introduced [7]. The main idea of lattice is to represent word alternatives in regions of the speech signal where the ambiguity in the acoustic recognition is high. The advantage is that the acoustic recognition is decoupled from the application of the language model, in particular a long-span language model, which can be applied in a subsequent post-processing step. The number of word alternatives should be adapted to the level of ambiguity in the acoustic recognition. In the following, we present the lattice generation which was proposed by Odell [7]. According to Odell, only a few simple modifications are needed to modify the one-pass time-synchronous decoding to generate a lattice of hypotheses. Rather than discarding all but the most likely partial path when these merge at word ends, it is possible to retain information about them to allow lattice traceback. When only the most likely hypothesis is required, the language model likelihoods are added to the word ending state and the traceback information updated. The states from equivalent partial paths are recombined and only the most likely survives to propagate into the following network. When a lattice of hypotheses is needed, the less likely word ending states are not discarded but are linked to the most likely one and the combined structure propagates into the following network. The calculation in the remainder of the network is only performed on the most likely word ending state but when traceback occurs at the end of the sentence, all of the word ending states are used to construct a lattice of hypotheses. At the end of each utterance, traceback proceeds separately through each of the linked word ending states and a lattice of alternative hypotheses is constructed. Each node in the generated lattice has an associated time. Each arc has an associated word identity, the acoustic likelihood and the language model likelihood, and forms a link between two nodes which define the start and end times of the word hypothesis

2.3. Word Confusion Networks

With lattice decoding, we have to face with a computation problem. In general, the number of paths through a lattice is exponential in the number of links. These paths correspond to different segmentation hypotheses (i.e. word sequence plus boundary times) of utterances. A method to overcome these problems has been suggested in [8]. The lattice is first transformed into a special form in which the calculation of the expected word error rate becomes trivial. This special form of a lattice is called a confusion network. A confusion network itself is a lattice. Each edge is labeled with a word and a probability. The most important feature of these lattices is that they are linear, in the sense that every path from the start to the end node has to pass through all nodes. A consequence of this (combined with the acyclicity) is that all paths between two nodes have the same length. Thus the confusion network naturally defines an alignment for all pairs of paths. This alignment is used as the basis for the word error rate calculation. An approach to actually construct a confusion network from a lattice is also presented in [8]. The task is treated as a clustering problem, where the edges from the original lattice have to be clustered into groups according to criteria such the overlapping in time between edges and the phonetic similarity between words.

2.4. Pruning

Since the lattice produced by using hypotheses directly output by the decoder can be very large, it is essential to use pruning methods for generating compact lattices. In this study, two different pruning techniques were considered:

- The forward-pruning. At each time frame, only the most promising hypotheses are retained in one-pass search. This pruning technique is usually named “beam search".
- The forward-backward pruning. It employs a beam search with respect to the forward and backward

scores of a specific hypothesis. Strictly speaking, for every lattice arc representing a word hypothesis, we compute the overall score of the best path passing through this specific arc. Word arcs whose scores are relatively close to the global score of the best path are kept in the word lattice, while the others are pruned.

2.5. Evaluation Criterion

In order to evaluate the quality of lattices, it usually relies on two criteria: the size of the generated lattice and the lattice error rate (LER). This section will cover both criteria.

2.5.1. Lattice size

There are some different measures about the quality of the lattice, with respect to its size. Aubert used the word lattice density, the node lattice density and the boundary lattice density as measures [9]. Those criteria have been widely used in most lattice evaluation systems. They are informally defined as follows.

- The word lattice density (WGD) is defined as the total number of lattice arcs divided by the number of actually spoken words.
- The node lattice density (NGD) is defined as the total number of different words ending at each time frame divided by the number of actually spoken words.
- The boundary lattice density (BGD) denotes the number of word boundaries, i.e. different start times, respectively, per spoken word.

2.5.2. Lattice word error rate

The lattice word error rate (LER) is computed by determining which sentence through the lattice best matches the actually spoken sentence. The match criterion is defined in terms of word substitutions (SUB), deletions (DEL) and insertions (INS). This measure provides a lower bound of the word error rate for this lattice. The algorithm, for its computation, is very similar to the one that computes the string edit distance. Obviously, the LER can be used for word confusion network. Similarly, The N-best word error rate is calculated by choosing the sentence with the minimum number of errors among the top N-best sentence hypotheses. The N-best word error rate when N = ∞ is interpreted as the LER and when N=1 is interpreted as word error rate (WER). Therefore, the WER is the edit distance between the best hypothesis and its given utterance.

3. THE VIETNAMESE LVCSR SYSTEM

In this section, we present the Vietnamese LVCSR system, which has been developed at AILAB, VNU-HCM since 2005.

3.1. Training and Testing Data

Table 1. Data for training and testing

	Training		*Testing*	
Dialect	Length (hours)	#Sentence	Length (hours)	#Sentence
Hanoi	18.0	17502	1.0	1021
Saigon	2.0	1994	-	-
Total	20.0	19496	1.0	1021

3.1.1. Acoustic training and testing data

We used the VNBN for training and testing [1]. The acoustic training data was collected from July to August 2005 from VOV – the national radio broadcaster (mostly in Hanoi and Saigon dialects), which consists of a total of 20 hours. The recording was manually transcribed and segmented into sentences, which resulted in a total of 19,496 sentences and a vocabulary size of 3174 word unit (WU). The corpus was further divided into two sets: training and testing, as shown in Table 1. The speech was sampled at 16kHz and 16 bits. They were further parameterized into 12 dimensional MFCC, energy, and their delta and acceleration (39 length front-end parameters).

3.1.2. Language training data

The language model training data comes from newspaper text resource. In particular, an 100M WU collection of the national wide newspaper, VOV, was employed, which included all issues between 1998-2005. Numeric expression and abbreviated words occurring in the text were replaced by suitable labels. In addition, the transcription of the acoustic training data was also added into the corpus.

3.2. The acoustic models

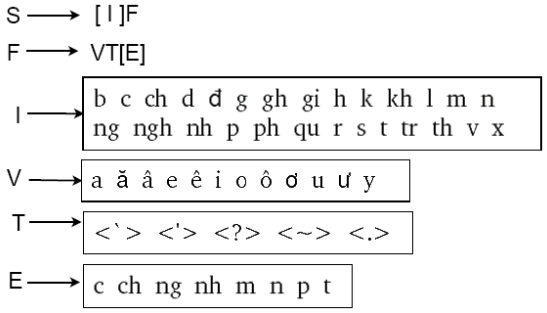

Figure 2 Initial-Final Units for Vietnamese

As depicted in Fig 2, we follow the usual approach as for Mandarin acoustic modeling [10] in which each syllable is decomposed into initial and final parts. While most of Vietnamese syllables consist of an initial and a final part, some of them have only the final. The initial always corresponds to a consonant. The final includes main sound plus tone and an optional ending sound. This decomposition results in a total number of 44 phones, as shown in Fig 2.

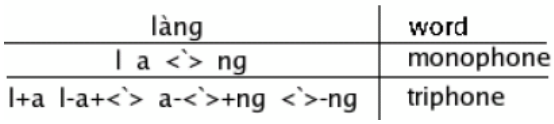

Figure 3 Constructing of triphones

There is an interesting point in our decomposed scheme here, which is related to a given tone in a syllable. Specifically, we treat the tones as a distinct phoneme and it follows immediately after the main sound. With this approach, the context-dependent model could be built straightforwardly. Fig. 3 illustrates the process of making triphones from a syllable.

3.2 Language Model

Both the bigram and trigram WU-based LM were trained on the text corpus mentioned above using the SRILM toolkit [11] with Kneser-Ney smoothing. For this, a lexicon with the 5K most frequent WUs was used. This lexicon gives an 1.8% OOV rate on the newspaper corpus and about 1.0% on the VNBN data. The perplexity of bigram and trigram language, measured on the transcription of the 1021 testing sentences was 188.6 and 136.2 respectively.

4. EXPERIMENTAL RESULTS

This section presents all the experimental results of the algorithms mentioned in Section 2, including the lattice, the N-best list and consensus decoding. Moreover we also reported the influences of the pruning methods on the LER.

4.1 The Decoding Experiments

Table 2. WER with different decoding methods.

Decoder	Lattice Decoding		Consensus Decoding	
	bigram	trigram	bigram	trigram
19.1	20.8	19.1	20.2	18.8

Let us begin with the decoding experiments. Three decoding methods, namely the decoder, the 1-best lattice decoding and the consensus decoding, are evaluated in terms of WER. The difference between the three decoding methods is in the way the best hypothesis is chosen. With the decoder, the best output hypothesis is taken directly from the recognizer, while with the 1-best lattice decoding, the best hypothesis is the best path in the lattice. Finally, in consensus decoding, the best hypothesis is the consensus hypothesis, which has the highest posterior probability among all possible hypotheses taken from the confusion network. This evaluation should fulfill two important requirements:

- correctness: for the implementation of the word lattice generation algorithm.
- improvement: for the consensus decoding.

Table 2 reports the WER for the VNBN datasets with respect to the three different decoding methods. Specifically, while the WER of the decoder and the 1-best lattice decoding give the same value of 19.1% with trigram-case and 20.8 with the bigram case, the WERs of the consensus decoding for both bigram and trigram cases are smaller (20.2 and 18.8 respectively). These results satisfy both the correctness and the improvement requirements.

4.2 The Pruning Experiments

In this section, we report our experiments on two different pruning methods. The first pruning technique is the beam search with different beam-width. The second one is based on the forward-backward pruning algorithm.

4.2.1.Imfact of beam-width

Table 3. Costs of the decoder with different thresholds

Threshold	Time	Active State	Active Transition	Active Model
1.40^{-40}	158.45	162946	198558	177056
1.10^{-50}	173.36	444816	818468	389770
1.10^{-60}	242.02	1225273	2151878	757241
1.10^{-70}	335.03	2911172	4872872	1301022
1.10^{-80}	612.41	6012061	9595802	2043120

Table 3 shows the costs of the recognition system for a part of the VNBN dataset, with respect to different beam-widths. By decreasing the threshold value from 10^{-40} to 10^{-80}, the ratio of time that the system needs to complete the recognition task is approximately 1:3.4 and the ratio of active states is 1:13.37 respectively. This means that not only the time but also the used memory increase as the beam-width value is set larger. The same result holds for the lattice generation. However, there is a trade-off here. Specifically, when the beam-width is large, the density of the generated lattices will be high. Consequently, LER of these lattices will be low. The choice of the beam-width is very important. It should satisfy both the time and the LER constraints. Keeping this in mind and looking at the experimental results, we have chosen the threshold value of 10^{-50} for all experiments.

4.2.2 Forward-Backward based Pruning for Lattices

Table 4. Bigram lattice LER with different beam-width

B-W	WGD	NGD	BGD	LER
Inf	2363.2	389.8	34.6	5.1
300	2200.5	337.2	34.0	5.1
250	2014.7	317.2	33.4	5.1
200	1635.8	278.1	31.8	5.1
150	991.3	134.6	22.2	5.2
100	298.7	87.8	18.0	5.5
50	27.3	13.1	4.1	7.1

Table 5. Trigram lattice LER with different beam-width

B-W	WGD	NGD	BGD	LER
Inf	139.2	75.5	8.7	14.2
300	135.8	73.9	8.7	14.2
250	132.2	72.4	8.6	14.2
200	126.0	69.8	8.6	14.2
150	114.1	63.3	8.5	14.2
100	71.9	42.4	7.5	14.3
50	11.0	7.4	2.3	14.8

Table 4 and 5 show the impact of the forward-backward pruning algorithm on the lattice quality for the VNBN data set. In the first line of the tables, the beam-width value of *Inf* means that no pruning is performed. The LERs, in this initial case, were 5,1 and 14.2, for bigram-based and trigram-based lattices respectively. When using the threshold value of 100, we obtained an absolute increase in GER of 0.4% but the reduction in WGD is significant, as the ratio of 1:8 are obtained with respect to the initial case for bigram-based lattices. The results for the trigram lattices were consistent with the bigram ones as shown in Table 5.

4.2.3 Forward-Backward based Pruning for WFN

Table 6. Bigram CFN LER with different beam-width

B-W	WGD	NGD	BGD	LER
1400	103.7	16.1	8.8	3.6
900	80.8	14.3	7.6	3.9
500	49.3	12.3	6.1	4.2
300	35.2	9.7	5.1	4.6
200	26.8	8.1	4.4	5.0
100	16.8	6.0	3.4	5.7
50	10.9	4.6	2.6	6.7

Table 7. Trigram CFN LER with different beam-width

B-W	WGD	NGD	BGD	LER
1400	35.79	6.0	3.7	12.1
900	27.6	5.3	3.2	12.2
500	18.3	4.5	2.7	12.2
300	13.5	3.9	2.3	12.3
200	10.8	3.5	2.0	12.4
100	7.5	3.0	1.7	12.7
50	5.7	2.7	1.5	12.8

We applied the same procedure for pruning lattices, before passing them to the confusion network construction. As mentioned in Section 2, the confusion network construction stage takes a lattice as its input and produces a compact representation, namely the confusion network, by grouping edges which have similar pronunciation words and overlap in time.

Table 6, 7 show the impact of the pruning threshold values on the quality of confusion networks for the VNBN dataset. From the numbers on these tables, two comments arise:

- Firstly, it is important to notice that the LER for confusion networks is always smaller than the LER of lattices. At the beam-width value of 1400, that ban be considered equivalent to the *inf* value of lattices, the LER for for bigram-based confusion networks is 3.6, while the LER for lattices, as reported in Table 4 was 5.1.
- Secondly, confusion network sizes are always smaller than lattice sizes, at the same LER values.

In summary, the following conclusions can be drawn for the forward-backward based pruning algorithm:

- In the best case, the lower-bound of the error is 3:6%, with respect to the LER of bigram-based confusion network for the VNBN dataset. It is a very positive result.
- In most cases, the lattice sizes can be at least halved, with a relatively small effecting to their LERs.
- In all cases, the LERs of confusion networks are always smaller than the LER of the corresponding lattices, at the same pruning condition. The same results are obtained for the WER

4.3 The N-Best Experiments

Table 8 The N-Best experiments

N	bigram case	trigram case
∞	5.1	14.2
1000	10.2	14.4
500	10.4	14.4
300	10.5	14.5
100	11.1	14.8
50	11.7	14.9
10	14.2	16.1
5	16.4	17.0

Table 8 presents the experimental results of N-best list on the VNBN dataset. The top-most line in the Table shows the WER with N=∞. In fact its actually the LER of trigram and bigram lattices with respect to VNBN dataset. As we can see, just with the first top 50 best sentences, we can get a WER of 14.9, compared to the 19.1 WER of the 1-best lattice decoding.

5. CONCLUSIONS

In this paper, we report an empirical study of the multipass decoding for Vietnamese LVCSR. The study consists of N-best, lattice and consensus network decoding. Two pruning methods were also investigated on the quality of lattices and confusion networks. All the experiments were carried out on the VNBN corpus. Results from the study could be used not only for improving the performance of the Vietnamese LVCSR system but also for translating Vietnamese to other languages.

6. REFERENCES

[1] Ha Nguyen, Quan Vu, “Selection of Basic Units for Vietnamese Large Vocabulary Continuous Speech Recognition”, The 4th IEEE International Conference on Computer Science - Research, Innovation and Vision of the Future, HoChiMinh City, Vietnam, Feb 2006.

[2] Ha Nguyen, Quan Vu, “Progress in Transcription of Vietnamese Broadcast News”, in Proceedings of HUT-ICCE2006, Hanoi, Vietnam.

[3] Quan Vu et al, “Vietnamese Automatic Speech Recognition: the FLaVoR approach”, The 5th International Symposium, ISCSLP 2006, Singapore.

[4] R. de Mori el al, “Spoken Dialogues with Computers", Academic Press, San Diego, CA, USA, 1998.

[5] R. Schwartz and S. Austin, ”A comparison of several approximate algorithms for _nding multiple (N-best) sentence hypotheses", in Proceedings of ICASSP’1991, Toronto, Canada, May 1991.

[6] B. H. Tran, F. Seide, V. Steinbiss, “A word graph based N-Best search in continuous speech recognition", in Proceedings of ICSLP 1996.

[7] J. J. Odell, “The Use of Context in Large Vocabulary Speech Recognition", Ph.D. thesis, University of Cambridge, Cambridge, UK, 1995.

[8] L. Mangu, E. Brill, and A. Stolcke, “Finding consensus among words: Lattice-based word error minimization", in Proceedings ISCA European Conference on Speech Communication and Technology 1999, Budapest, Hungary, September 1999.

[9] X. Aubert and H. Ney, “Large vocabulary continuous speech recognition using word graphs", in Proceedings of ICASSP’1995, Detroit, MI, USA, 1995.

[10] J.J.Wu et al., “Modeling Context-dependent Phonetic Units in a Continuous Speech Recognition System for Mandarin Chinese”, ICSLP 96, Philadelphia, USA, 1996.

[11] A. Stolcke, “SRILM - an extensible language modeling toolkit", ICSLP 2002.