國 立 清 華 大 學

碩 士 論 文

題目: Viterbi 搜尋的最佳化以及多語系辨識

Viterbi Beam Search Optimization and

Multilingual Speech Recognition

系別 資訊工程學系 組別

學號姓名 894342 林玄松 (Shiuan-Sung Lin)

指導教授 張智星博士 (Jyh-Shing Roger Jang)

呂仁園博士 (Ren-Yuan Ly)

江永進博士 (Yuang-Chin Chiang)

中華民國九十一年六月

Abstract

Most successful speech recognition systems are based on Hidden Markov Models (HMM), which rely on computation-intensive Viterbi search during recognition. The first part of this thesis focuses on the optimization of Vierbi beam search in HMM decoding for isolated-word speech recognition. The proposed data-driven method can effectively identify a near-optimal beam search ranking curve that can reduce the computation time to an acceptable amount while minimizing the reduction in recognition rate based on a set of sample data. Experimental results based on the most famous 300 poems in Tang Dynasty of China demonstrate the feasibility of the proposed approach.

In the second part of this thesis, we applied the proposed approach to a multilingual speech recognition system that can deal with Mandarin Chinese, Taiwanese, English, and their combinations. The system employs different acoustic models for different languages, and hence possesses a high degree of flexibility and modularity. We also described how to treat similar phone models as an equivalence class in order to reduce the total number of phone models for a limited speech corpus. Experimental results demonstrated its feasibility for automatic speech recognition for fixed-domain vocabularies.

Acknowledgement

I would like to express my deeply gratitude to my advisor Prof. Jyh-Shing Roger Jang whose suggestions and stimulating encouragements help me in all the time of research.

I have furthermore to thank Prof. Ren-Yuan Lyu from Chang-Gung University and Prof. Yuang-Chin Chiang from Institute of Statistics, National Tsing Hua University. They give me many valuable experiences and advices in building a speech recognition system.

My lab mates from National Tsing Hua University and Chang-Gung University supported me in my research work. I want to thank them for their assistance.

I would like to give special thanks to my family whose support enabled me to complete this work.

Contents

Index

Part1: Optimization of Viterbi Beam Search

Introduction

HMM (hidden Markov models) has been used for speech recognition with satisfactory results for the past few decades (Rabiner L. and Juang B.-W. 1993, Huang X., Acero A., and Hon H.-W. 2001). However, the intensive computation associated with Viterbi search in HMM decoding has posed a computation barrier for large-vocabulary ASR (automatic speech recognition) applications, or medium-vocabulary ASR applications on platforms with limited computing power, such as PDA or microprocessor based embedded applications. Conventionally, a full Viterbi search is replaced with a straightforward Viterbi beam search (Huang X., Acero A., and Hon H.-W. 2001) such that a balance between speed and accuracy is achieved. This paper goes one step further and presents an innovative data-driven approach to the optimisation of beam search in HMM decoding. The proposed method identifies an appropriate beam search ranking curve that can effectively reduce Viterbi beam search time while keep the reduction in recognition rate to a minimum. We have applied the proposed approach to the recognition of 3211 sentences from the most famous 300 poems of Tang Dynasty in China. The experimental results demonstrate that our approach can effectively reduce the search time with no or little reduction in recognition rate, which compares favourably with the conventional simple-minded approach in beam search. The rest of this paper is organized as follows. The next section introduces the conventional Viterbi beam

search used for HMM decoding. Section 3 explains the proposed approach that allows a better tradeoff between speed and accuracy. Section 4 demonstrates the experimental results based on poems in Tang Dynasty of China. Conclusions and future work are given in Section

Viterbi Beam Search in HMM Decoding

It is well known that a complete Viterbi search is computation intensive and most large-vocabulary ASR systems would rely on beam search (Lowerre 1976, Huang X., Acero A., and Hon H.-W. 2001) to cut down the computation cost. In other words, we do not need to search the entire Viterbi trellis to find the optimal path. Instead, we limit the number of branch-out candidates (which is proportional to the computation cost) according to a certain heuristics. Common heuristics in beam search can be divided into the following categories:

1. Rank based: At each time frame, only candidates with probabilities in top-n (n=1000, for instance) ranking are kept for branching out in the next time frame.
2. Probability based: At each time frame, only candidates with probabilities no less than a threshold (d in log probability, for instance) from the maximal probability of this frame are kept for branching out in the next time frame.

3. Combination of the above two: Only candidates that satisfy the above two conditions simultaneously are kept for branching in the next time frame.

Some other pruning strategies can be applied in addition to the above rules. For instance, we can make the parameters or a function of time frame indices. The use of frame-index-based pruning parameters can further increase the ratio between recognition rate and computation time, which have been supported in HTK, or Hidden Markov Model Toolkit (HTK 2002). However, the determination of these time-dependent parameters becomes another issue that calls for further investigation. The goal of this paper is to propose a systematic approach that can determine the rank-based beam search parameter as a function of the frame index. For the ease of discussion, we shall refer to the curve with respect to the time index as BSRC (beam search ranking curve). The BSRC is first expressed as a parameterised curve of the sum of two exponential functions, and then is optimised by the use of a sample data set according to an appropriately selected object function. Details of the proposed method are covered in the following section.

Proposed Approach to BSRC Optimization

Given a set of sample data (a set of recordings with corresponding correct answers), our goal is to identify the best BSRC that can achieve an optimal ratio between recognition rate and computation time. Since the “optimal ratio” is based on

a data set, we need to be very careful regarding the evaluation of our approach. Basically, the evaluation involves the following steps:

1. Divide the sample data set into training and test data sets.
2. Use our design method on the training set to find the best BSRC.
3. Use the test set to evaluate the ASR system with the identified BSRC.

To avoid the dependency on a specific way of partitioning the sample data set, the above steps have to be repeated many times to get an unbiased result.

The basic idea behind our design method is based on the observation of each recording's ranking curves of the training set during HMM decoding. Once the ranking curves of all recordings in the training data are collected, we can simply find the maximum of them at each time frame to form a maximal ranking curve. If we simply use the maximal ranking curve as BSRC, then we would obtain 100% recognition rate on the training set. However, we still need to take care of the following two issues:

1. The maximal ranking curve is usually rugged, which would lead to poor performance on the unseen test data set. Hence we need to use a smooth curve as the BSRC that can "cover" the maximal ranking curve to get a 100% recognition rate on the training set.
2. Even with a smooth BSRC, the computation time may still be too long, especially on slow platforms such as PDA or embedded systems. Hence we need to further

identify a optimal curve that relates recognition rate with respective to computation time.

Our observation on the maximal ranking curve suggests that the smooth BSRC should have the format :

$$n(x) = c_1 e^{-\lambda_1 x} + c_2 e^{-\lambda_2 x},$$

where x is the frame index and c_1, c_2, λ_1, λ_2 are adjustable parameters. To find a set of parameters for BSRC that can "cover" the maximal ranking curve, our approach involves the following steps:

1. Use least-squares criterion to fit the BSRC to the maximal ranking curve. In particular, c_1 and c_2 are linear parameters can be found with least-squares estimate. On the other hand, λ_1 and λ_2 are nonlinear parameters and can be found with downhill Simlex method. This hybrid data fitting approach can effectively find the optimal parameters (Jang, Sun and Mizutani 1997).
2. Use the parameter set identified above as the initial guess for a second-step optimisation based on a different objective function that aimed to "cover" the maximal ranking curve.

The initial parameters identified in the first step can greatly increase the optimization results in the second step. The objective function used in the second step will be described in the next section.

If the identified BSRC does not fulfil the requested response time, then we need to

sacrifice the recognition rate for shorter response time. This is achieved by an exhaustive manner which repeats the following steps for $i = 1$ to the number of training recordings:

1. Eliminate recording i in computing the maximal ranking curve.
2. Find the corresponding BSRC and compute its area (which is proportional to the actual computation time).

The BSRC with the smallest area is then picked as the one with minimal response time under a given smaller recognition rate. This procedure is repeated and we can obtain an optimal curve of recognition rate with respect to computation time that can help the design strategy on platforms with less computing power.

Some of the above descriptions should become more clear in the following section which exaplins how simulation is performed to obtain the experimental results.

Experimental Results

The ASR task used in this paper is the recognition of 3211 sentences of the most famous 300 poems from the Tang Dynasty in China. The HMM training is based on a balanced corpus recorded by 70 subjects in Taiwan. The distributions of lengths of these sentences are

1. 3 characters ===> 35 sentences
2. 4 characters ===> 13 sentences

3. 5 characters ===> 1401 sentences
4. 6 characters ===> 9 sentences
5. 7 characters ===> 1755 sentences
6. 8 characters ===> 1 sentence
7. 9 characters ===> 9 sentences

To collect the sample data set for the optimisation of beam search, we have 10 subjects (5 males and 5 females) to record about 300 sentences each. If the number of candidate at each frame is set as 1200, we can have a perfect recognition rate of 100% and the ranking for each recording can be seen in the following plot:

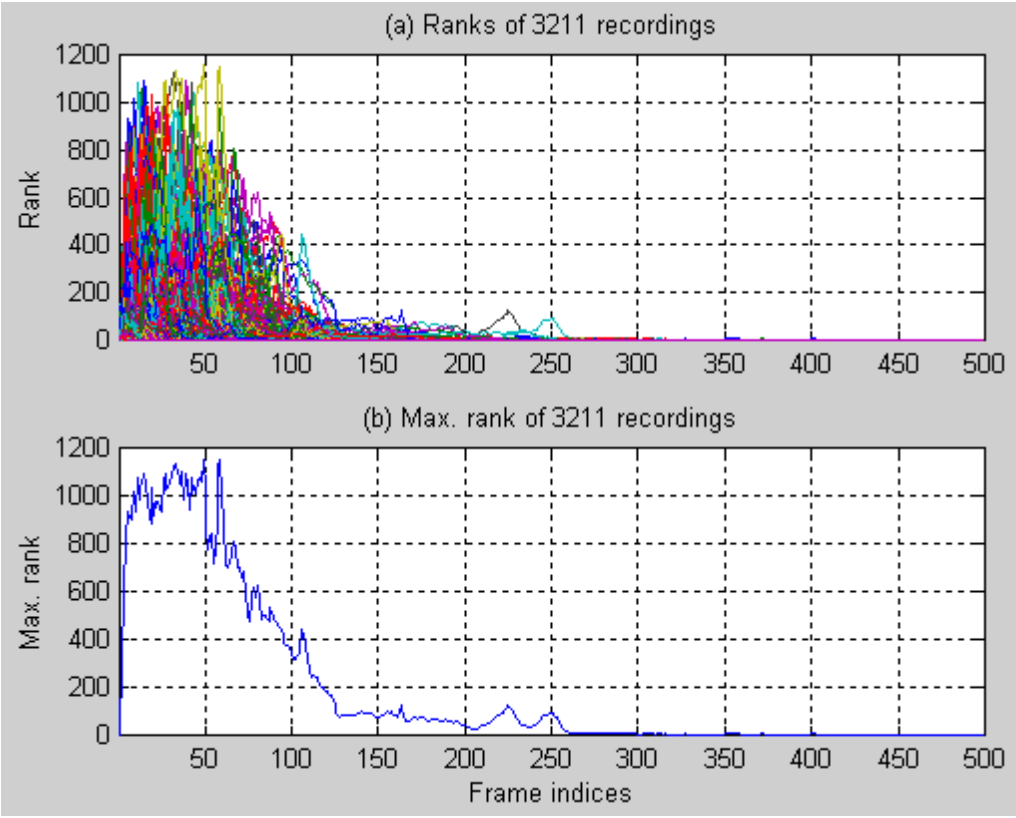

Figure 1-1: ranks of 3211 recordings and maximal rank of 3211 recordings

Obviously, if the BSRC (beam search ranking curve, that is, number of branching-out candidates versus frame index) is set as a constant of 1200, then we can get a recognition rate of 100%. However, a constant BSRC is time consuming since the computation time is proportional to the area below the BSRC. On the other hand, we can set the BSRC as the maximal rank in (b) of the above plot. Unfortunately, this is data dependent and the rugged curve obviously cannot have good performance with other recordings. Therefore our goal is to find a smooth BSRC that can

1. Fulfil the constraint of required computation time.
2. Minimize the reduction in recognition rate due to the above timing constraint.

Alternatively, our goal is to find an optimal curve of recognition rate with respect to computation time, such that given a required computation time, we can identify the corresponding best recognition rate and set the parameters for the BSRC accordingly.

To make a fair evaluation, we first need to divide the sample data set into a training set and a test set. All the parameter tuning is based on the training set, while the evaluation is based on the test set. Also to make the evaluation less dependent on the way the sample data is divided, the same procedure has to be repeated and the results are averaged to get an unbiased estimate.

To suitably represent the desired BSRC, we use the following parameterised equation:

$$n(x) = c_1 e^{-\lambda_1 x} + c_2 e^{-\lambda_2 x}$$

Ideally, the BSRC should cover the maximal ranking curve to have a 100% recognition rate. At the same time, we should try to minimize the area under BSRC, which is proportional to the overall computation time. To find the BSRC with minimum area, we take the following two-step approach. First, we use the maximal ranking curve as a sample data set to fit the curve of $n(x)$. In particular, we use linear least-square estimate to identify c_1 and c_2 since they are linear parameters of $n(x)$. For λ_1 and λ_2, we employ downhill Simplex method to find these nonlinear parameters. The identified BSRC is shown in the following plot:

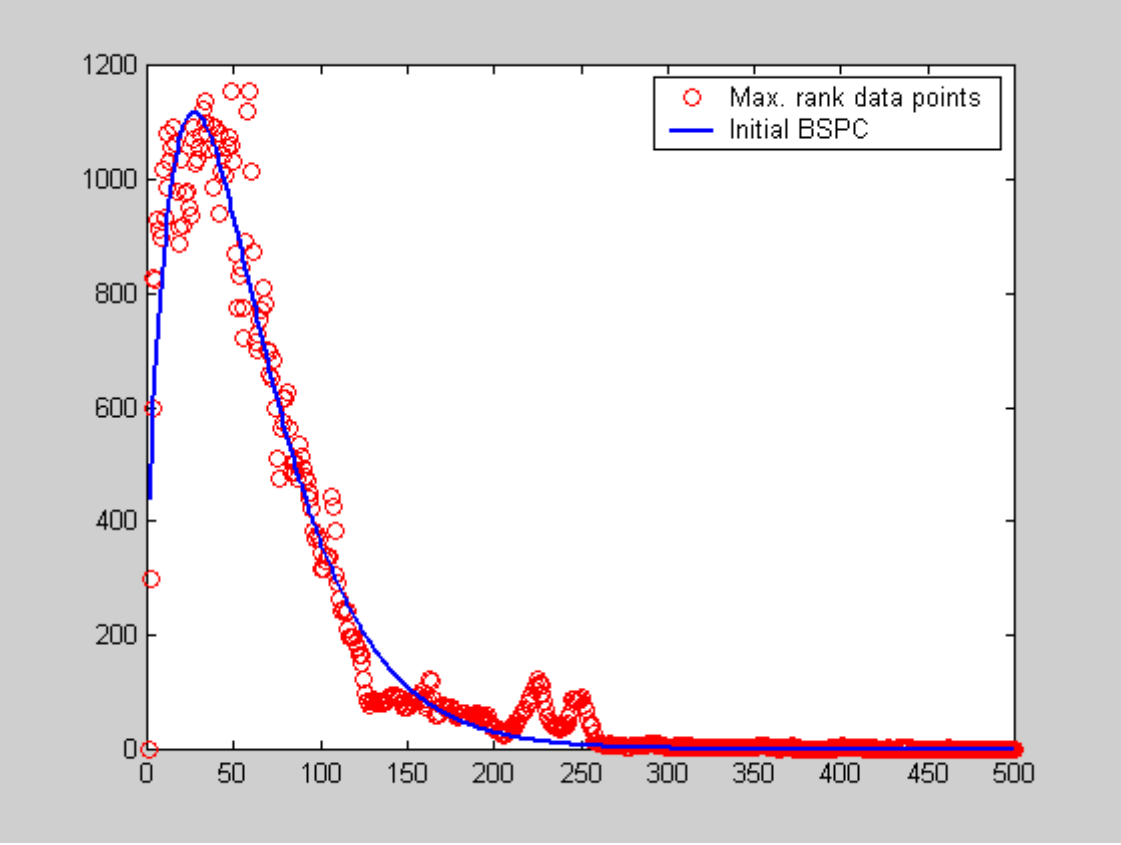

Figure 1-2: number of branching-out candidates versus frame index

The above BSRC is not exactly what we want, since the desired curve should

cover all points of maximal ranking curve to get a 100% recognition rate. The BSRC obtained via data fitting serves as an initial guess for the second-stage optimisation aimed at covering all red dots with minimal area. The objective function can be expressed as follows:

$$J(c_1,c_2,\lambda_1,\lambda_2)=\sum_{i=1}^{n} dist\left(n(x_i,c_1,c_2,\lambda_1,\lambda_2),y_i\right)$$
$$=\sum_{i=1}^{n} dist\left(c_1e^{-\lambda_1 x_i}+c_2e^{-\lambda_2 x_i},y_i\right)$$

where the distance function is asymmetric:

$$dist(n,y)=\begin{cases}k_1(n-y),if\ n\geq y\\ k_2(n-f),if\ n<y\end{cases}$$

Usually k_2 is much larger then k_1, such that the optimisation procedure is likely to lead to a set of $(c_{1,}c_2,\lambda_1,\lambda_2)$ that makes the BSRC cover the red dots. Since the objective function is not a squared sum error, we cannot apply least-squares estimate. Hence we still rely on downhill Simplex method to find the optimal values of $(c_{1,}c_2,\lambda_1,\lambda_2)$. A typical result after the second-stage optimisation, with $k_1=1$ and $k_2=10$, is shown next:

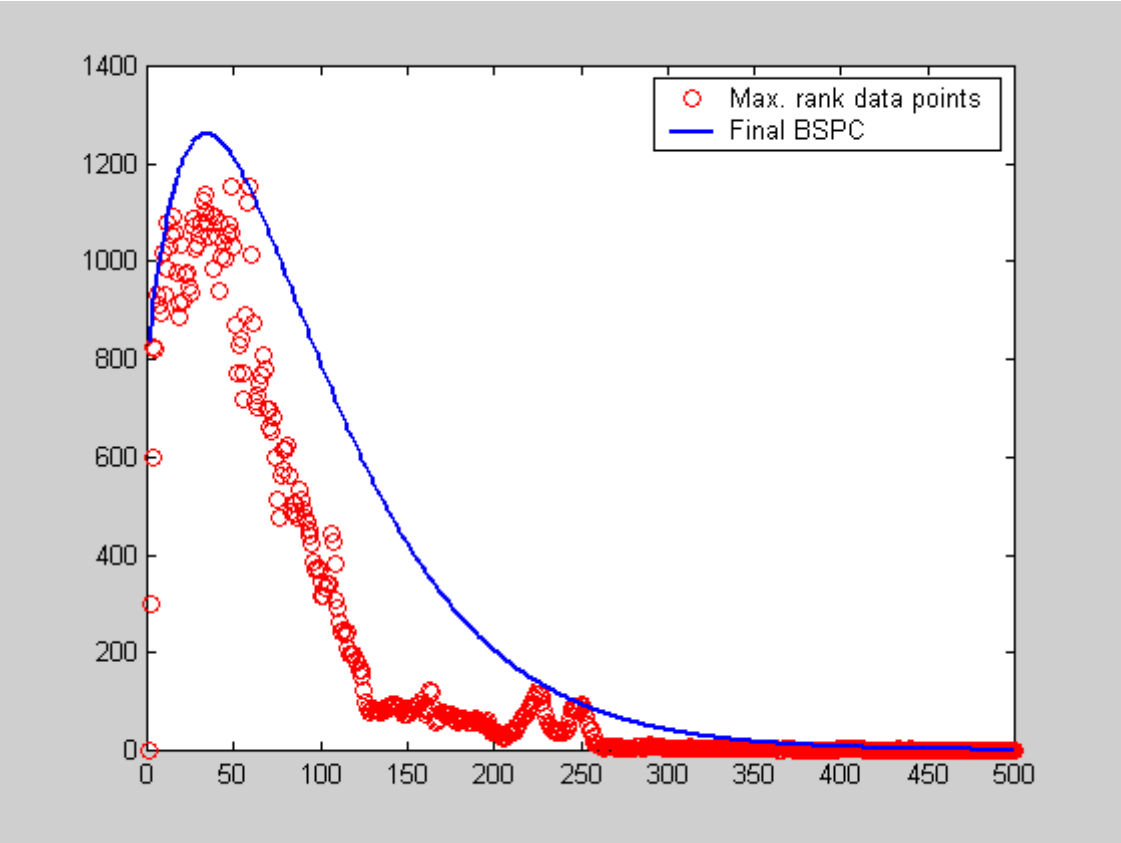

Figure 1-3: BSRC obtained after the second-stage optimisation

Apparently the identified BSRC can cover all the maximal ranking data and achieve a recognition rate of 100%. However, the area under the curve perhaps is still to large and the associated computation time is still too long, considering the computing power of small/mobile device such as PDA or embedded microprocessors. As a result, we need to further tune the parameters to reduce the area. In other words, we need to "sacrifice" some of the touching points and move the BSRC downward. Once a touching point (together with the same-recording points that contribute to the maximal ranking curve) is removed, the recognition rate is lower but the computation time is shorter.

To achieve the best tradeoff between the recognition rate and computing time, we

adopt a exhaustive approach which tries to find a ranking curve of a recording that, when removed, can lead to the largest reduction in the area under the obtained BSRC. This process is repeated until 200 recordings (and their corresponding ranking curves) are removed. Once a recording is removed, we need to recompute the best BSRC and evaluate it using both the training and test sets. The resulting curves of recognition rates (both training and test) with respective to the area under BSRC (which is proportional to the computing time) is shown next.

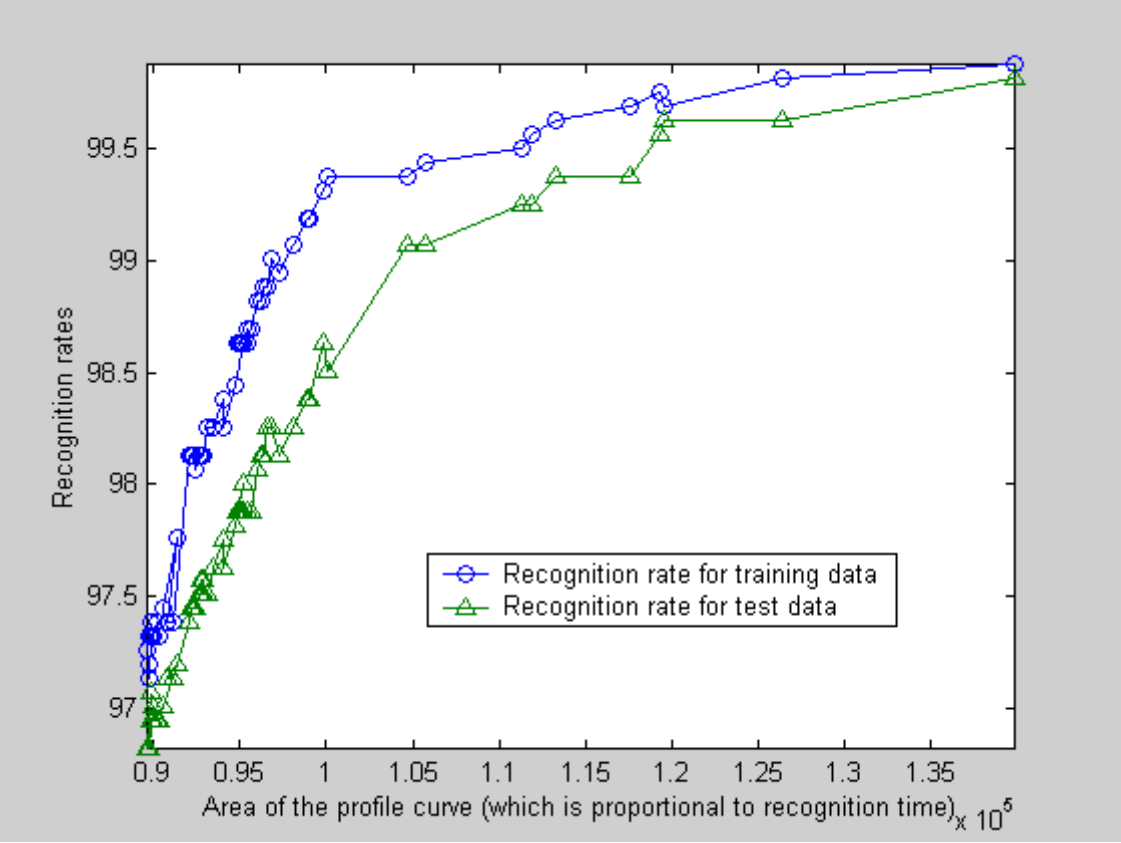

Figure 1-4: resulting curves of recognition rates/time

Conclusion

In this paper, we have proposed a data-driven approach to the optimisation of the ranking curve of beam search in Viterbi decoding. The proposed approach casts the determination of the ranking curve in beam search as a constrained optimisation problem, in which a minimal reduction in recognition rate is achieved with respect to a given allowable computation time. In other words, a optimal tradeoff between speeh and accuracy can be achieved. Though the proposed method is based on a set of sample data, extensive simulation demonstrates its feasibility and reliability, which is suitable for ASR system on limited small/mobile devices such as PDA or microprocessors.

Part2: Multilingual Speech Recognition

Introduction

In the second part of this thesis, we applied the proposed optimisation approach for Viterbi decoding to a multilingual speech recognition system. In the first part of this thesis, the experiment of recognizing 3211 sentences from the most famous 300 poems of Tang Dynasty was based on a bilingual acoustic model of both Mandarin Chinese and Taiwanese. On the other hand, the multilingual speech recognition system to be introduced in this part is constructed from two acoustic models operating in parallel. The first acoustic model is the previous Chinese/Taiwanese one while the other is for English. The test sentences are based on smaller set of vocabularies in which each sentence can contain Chinese, Taiwanese, or English simultaneously. Again, we use a tree lexicon to reduce the computation time.

There are 526 phone models defined in the Chinese/Taiwanese acoustic model, and 2474 in the English acoustic model. Too many phone models lengthen the time for both training and recognition without actually improving the recognition performance. As a result, we need to combine phone models into equivalence classes to reduce the number of phone models in the English acoustic model. The equivalence classes were constructed manually and the number of phone models is successfully cut down to 717.

Experimental results demonstrate that a set of properly combined phone models, a tree lexicon, together with the optimisation in Viterbi decoding, can effectively improve the performance of our multilingual speech recognition system.

Acoustic Model Training

Our multilingual speech recognition system uses two acoustic models: Chinese/Taiwanese and English. Chinese/Taiwanese acoustic model contains 526 RCD (right-context-dependent) phone models, which was basically extracted from a corpus of 70 persons collected by Multimedia Signal Processing Lab, Chang-Gung University. Models representing silence to vocal are composed of 3 states with 3 streams in each state and 10 mixtures in each stream. Other models are made up of 5 states with 3 streams in each state but have 6, 2 and 2 mixtures in respective streams.

The definition of phone models for English follows the dictionary provided by TIMIT. All English words are split to a sequence of phonetic alphabets. Every 2 phonetic alphabets form a pair of RCD phone model. Training configurations of English phone models follow the example of those in Chinese/Taiwanese. Initially we defined 1948 RCD phone models. The corpus texts are from TIMIT and the speech corpus was recorded by 13 persons in our lab. We decided not to use the speech corpus provided by TIMIT since we would like to have a multilingual recognition system that can deal with specific English accents by college students in Taiwan. Some similar vowels and consonants were merged manually to reduce the

number of models such that the trained phone models can be more representative of limited corpora.

The following table illustrates how similar vowels and consonants are merged into equivalence classes. Within the same ID category, symbols with the same equivalence index are considered to be the same class.

ID	Category	Equivalence_Index	Symbol	Example_Word	TRANSCRIPTION
1	Stops	1	b	bee	BCL B iy
2	Stops	2	d	day	DCL D ey
3	Stops	3	g	gay	GCL G ey
4	Stops	4	p	pea	PCL P iy
5	Stops	5	t	tea	TCL T iy
6	Stops	6	k	key	KCL K iy
7	Stops	7	dx	muddy	m ah DX iy
8	Stops	8	q	bat	bcl b ae Q
9	Affricates	1	jh	joke	DCL JH ow kcl k
10	Affricates	2	ch	choke	TCL CH ow kcl k
11	Fricatives	1	s	sea	S iy
12	Fricatives	2	sh	she	SH iy
13	Fricatives	1	z	zone	Z ow n
14	Fricatives	3	zh	azure	ae ZH er
15	Fricatives	4	f	fin	F ih n
16	Fricatives	5	th	thin	TH ih n
17	Fricatives	6	v	van	V ae n
18	Fricatives	7	dh	then	DH e n
19	Nasals	1	m	mom	M aa M
20	Nasals	2	n	noon	N uw N
21	Nasals	2	ng	sing	s ih NG
22	Nasals	3	em	bottom	b aa tcl t EM
23	Nasals	3	en	button	b ah q EN
24	Nasals	4	eng	washington	w aa sh ENG tcl t ax n
25	Nasals	4	nx	winner	w ih NX axr
26	Semivowels/Glides	1	l	lay	L ey

27	Semivowels/Glides	2	r	ray	R ey
28	Semivowels/Glides	3	w	way	W ey
29	Semivowels/Glides	4	y	yacht	Y aa tcl t
30	Semivowels/Glides	5	hh	hay	HH ey
31	Semivowels/Glides	5	hv	ahead	ax HV eh dcl d
32	Semivowels/Glides	6	el	bottle	bcl b aa tcl t EL
33	Vowels	1	iy	beet	bcl b IY tcl t
34	Vowels	1	ih	bit	bcl b IH tcl t
35	Vowels	2	eh	bet	bcl b EH tcl t
36	Vowels	2	ey	bait	bcl b EY tcl t
37	Vowels	2	ae	bat	bcl b AE tcl t
38	Vowels	3	aa	bott	bcl b AA tcl t
39	Vowels	4	aw	bout	bcl b AW tcl t
40	Vowels	5	ay	bite	bcl b AY tcl t
41	Vowels	3	ah	but	bcl b AH tcl t
42	Vowels	6	ao	bought	bcl b AO tcl t
43	Vowels	7	oy	boy	bcl b OY
44	Vowels	6	ow	boat	bcl b OW tcl t
45	Vowels	8	uh	book	bcl b UH kcl k
46	Vowels	8	uw	boot	bcl b UW tcl t
47	Vowels	8	ux	toot	tcl t UX tcl t
48	Vowels	9	er	bird	bcl b ER dcl d
49	Vowels	9	ax	about	AX bcl b aw tcl t
50	Vowels	1	ix	debit	dcl d eh bcl b IX tcl t
51	Vowels	9	axr	butter	bcl b ah dx AXR
52	Vowels	9	ax-h	suspect	s AX-H s pcl p eh kcl k tcl t

Table 2-1: phonemic and phonetic symbols used in the TIMIT lexicon. Equivalence indices are used to defined equivalence class for acoustic model training.

Tree Lexicon

Once the phone models for both Chinese/Taiwanese and English are defined, we need to label each sentence with syllable-like symbols. Moreover, since Chinese texts can be labelled with either Chinese or Taiwanese syllable symbols, a sentence can produce more than one syllable-like sequence, considering possible combinations of

Chinese/Taiwanese interchangeability. For instance, consider the following sentence:

“他穿Tom的布鞋”

The above sentence can be split into four sequences of syllable-like symbols:

0ta-0CuaN-1silt-1taa1-1aa1m-1msil-0dr-0bu-0sie

2i-2ciG-1silt-1taa1-1aa1m-1msil-0dr-0bu-0sie

0ta-0CuaN -1silt-1taa1-1aa1m-1msil-2e-2bo-2e

2i-0CuaN-1silt-1taa1-1aa1m-1msil-0dr-2bo-2e

where the leading digits, 0, 1, 2, represent Mandarin Chinese, English and Taiwanese respectively. The above different sequences of symbols all indicate the same meaning: “He wears Tom’s shoes”, except that the first part of the Chinese texts are labelled with Chinese syllables in sequences 1 and 3, and Taiwanese syllables in sequences 2 and 4, while the second part of the Chinese texts are labelled with Chinese syllables in sequences 1 and 2, and Taiwanese syllables in sequences 3 and 4.

Before these sequences of syllable-like symbols are then converted to models corresponding to a predefined dictionary for a specific language, sequences begins with the same several symbols can be merged to form a tree lexicon. The use of a tree lexicon can greatly reduce the search space and still maintain a high recognition rate. The construction of a tree lexicon can be demonstrated in the following example:

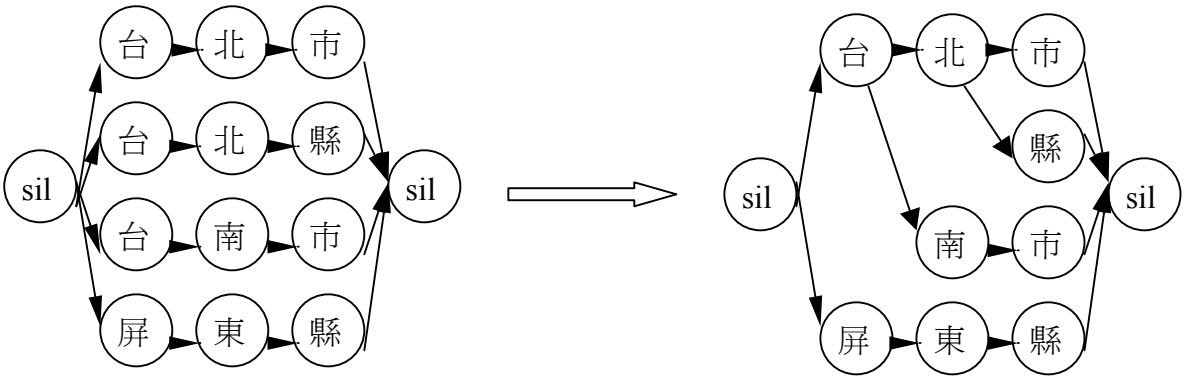

Figure 2-1: County names as a tree lexicon

Tree structure is still beneficial after applying to multilingual speech recognition system. Number of branches of left “sil” are 302 in the recognition of 3211 sentences from the 300 poems of Tang Dynasty while the total number of syllables in mandarin Chinese is 418. This implies that in some certain domain, number of nodes in tree structure wouldn’t grow vastly as a result of the restriction of grammar. Roughly, number of nodes in tree structure is about two third of original linear structure. Furthermore, conceptually, this approach can be applied from the end of sentences to form a double-ended tree with a smaller number of nodes, as show next.

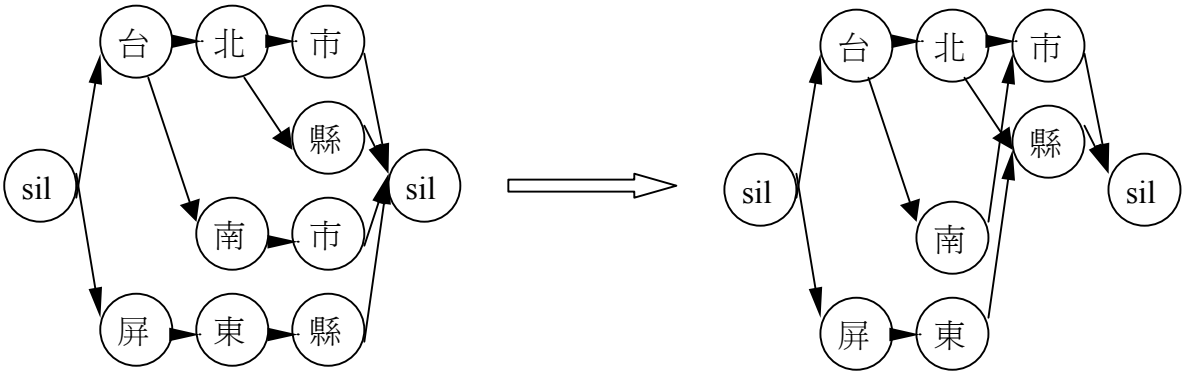

Figure 2-2: County names as a double-ended tree lexicon

However, our multilingual speech recognition system does not use the double-ended tree lexicon since there is no significant improvement of memory saving or computation speedup.

Experimental Results and Discussions

There are about 2000 sentences in the prompt texts, which were then read by 13 subjects to form the English speech corpus. The training was performed via HTK (Hidden Markov Model Toolkit). After analysing experimental results, we found that sentences contain only Mandarin Chinese or Taiwanese have a better performance. This is because these Chinese/Taiwanese phone models have been trained with a larger speech corpus of 70 persons. On the other hand, our English speech corpus only covers 13 subjects with limited recordings, which lead to a less desirable performance. For instance, we can examine the recognition process of a sentence of "They are students". The returned search paths with their log probabilities are listed next:

Search path in the tree lexicon	Log probability
sildh-dheh-ehsil-silaa-aar-rsil-sils-st-tuh-uhd-dem	-5382.076
sildh-dheh-ehsil-silaa-aar-rsil-sils-st-tuh-uhd-dem-emt	-5393.250
sildh-dheh-ehsil-silaa-aar-rsil-sils-st-tuh-uhd-dem-emt-ts	-5421.056

A more detailed model-level listing of the first search path is shown as follows:

Frame	Word	Model	State	Probability
00	W:2(sil)	M:0(sil+sil)	S:0	-76.621887
01	W:4(sildh)	M:0(sil+dh)	S:0	-167.181305
02	W:4(sildh)	M:0(sil+dh)	S:1	-247.925858
03	W:4(sildh)	M:0(sil+dh)	S:2	-335.374329
04	W:5(dheh)	M:0(dh+eh)	S:0	-405.007202
05	W:5(dheh)	M:0(dh+eh)	S:1	-462.736877
06	W:5(dheh)	M:0(dh+eh)	S:2	-525.862122
07	W:6(ehsil)	M:0(eh+sil)	S:0	-587.363586
08	W:6(ehsil)	M:0(eh+sil)	S:1	-646.356689
09	W:6(ehsil)	M:0(eh+sil)	S:1	-707.755432
10	W:6(ehsil)	M:0(eh+sil)	S:1	-777.571045
11	W:6(ehsil)	M:0(eh+sil)	S:2	-856.392700
12	W:7(silaa)	M:0(sil+aa)	S:0	-934.482422
13	W:7(silaa)	M:0(sil+aa)	S:1	-1005.831787
14	W:7(silaa)	M:0(sil+aa)	S:2	-1073.182617
15	W:8(aar)	M:0(aa+r)	S:0	-1133.596313
16	W:8(aar)	M:0(aa+r)	S:1	-1192.788452
17	W:8(aar)	M:0(aa+r)	S:2	-1251.430664
18	W:9(rsil)	M:0(r+sil)	S:0	-1306.490479
19	W:9(rsil)	M:0(r+sil)	S:0	-1362.130615
20	W:9(rsil)	M:0(r+sil)	S:0	-1415.821533
21	W:9(rsil)	M:0(r+sil)	S:0	-1473.932251
22	W:9(rsil)	M:0(r+sil)	S:1	-1538.538208
23	W:9(rsil)	M:0(r+sil)	S:1	-1607.267334
24	W:9(rsil)	M:0(r+sil)	S:1	-1680.518188
25	W:9(rsil)	M:0(r+sil)	S:2	-1756.479492
26	W:9(rsil)	M:0(r+sil)	S:2	-1833.380615
27	W:9(rsil)	M:0(r+sil)	S:2	-1907.511108
28	W:10(sils)	M:0(sil+s)	S:0	-1978.781982
29	W:10(sils)	M:0(sil+s)	S:0	-2044.085693
30	W:10(sils)	M:0(sil+s)	S:0	-2108.867432

```
31      W:10(sils) M:0(sil+s)   S:0     -2175.956543
32      W:10(sils) M:0(sil+s)   S:1     -2243.067627
33      W:10(sils) M:0(sil+s)   S:1     -2302.957764
34      W:10(sils) M:0(sil+s)   S:1     -2359.653320
35      W:10(sils) M:0(sil+s)   S:1     -2419.686768
36      W:10(sils) M:0(sil+s)   S:1     -2477.145508
37      W:10(sils) M:0(sil+s)   S:2     -2543.543457
38      W:11(st)   M:0(s+t)     S:0     -2607.357666
39      W:11(st)   M:0(s+t)     S:0     -2672.505127
40      W:11(st)   M:0(s+t)     S:0     -2733.998291
41      W:11(st)   M:0(s+t)     S:1     -2801.026123
42      W:11(st)   M:0(s+t)     S:1     -2870.102295
43      W:11(st)   M:0(s+t)     S:2     -2950.891357
44      W:11(st)   M:0(s+t)     S:2     -3022.075684
45      W:11(st)   M:0(s+t)     S:2     -3099.131836
46      W:12(tuh)  M:0(t+uh)    S:0     -3180.359131
47      W:12(tuh)  M:0(t+uh)    S:0     -3267.842529
48      W:12(tuh)  M:0(t+uh)    S:1     -3350.899658
49      W:12(tuh)  M:0(t+uh)    S:1     -3440.475830
50      W:12(tuh)  M:0(t+uh)    S:2     -3519.847900
51      W:13(uhd)  M:0(uh+d)    S:0     -3599.172852
52      W:13(uhd)  M:0(uh+d)    S:0     -3667.387939
53      W:13(uhd)  M:0(uh+d)    S:0     -3730.325928
54      W:13(uhd)  M:0(uh+d)    S:0     -3789.993164
55      W:13(uhd)  M:0(uh+d)    S:0     -3847.030273
56      W:13(uhd)  M:0(uh+d)    S:0     -3902.555420
57      W:13(uhd)  M:0(uh+d)    S:0     -3955.351562
58      W:13(uhd)  M:0(uh+d)    S:0     -4006.580811
59      W:13(uhd)  M:0(uh+d)    S:0     -4055.699463
60      W:13(uhd)  M:0(uh+d)    S:0     -4106.821777
61      W:13(uhd)  M:0(uh+d)    S:0     -4160.578613
62      W:13(uhd)  M:0(uh+d)    S:0     -4222.391602
63      W:13(uhd)  M:0(uh+d)    S:1     -4289.562012
64      W:13(uhd)  M:0(uh+d)    S:2     -4365.752441
```

65	W:14(dem)	M:0(d+em)	S:0	-4435.230469
66	W:14(dem)	M:0(d+em)	S:0	-4500.569336
67	W:14(dem)	M:0(d+em)	S:0	-4566.361816
68	W:14(dem)	M:0(d+em)	S:0	-4631.301758
69	W:14(dem)	M:0(d+em)	S:1	-4692.126465
70	W:14(dem)	M:0(d+em)	S:1	-4744.474121
71	W:14(dem)	M:0(d+em)	S:1	-4801.227539
72	W:14(dem)	M:0(d+em)	S:1	-4862.410645
73	W:14(dem)	M:0(d+em)	S:2	-4937.490723
74	W:14(dem)	M:0(d+em)	S:2	-5011.450195
75	W:14(dem)	M:0(d+em)	S:2	-5081.888672
76	W:14(dem)	M:0(d+em)	S:2	-5143.154785
77	W:14(dem)	M:0(d+em)	S:2	-5198.643555
78	W:14(dem)	M:0(d+em)	S:2	-5246.789551
79	W:14(dem)	M:0(d+em)	S:2	-5294.197754
80	W:14(dem)	M:0(d+em)	S:2	-5338.565430
81	W:14(dem)	M:0(d+em)	S:2	-5382.076172

There are 82 frames of speech in total. The path is almost correct, except that it could not reach the end of the sentence. In particular, frame 51 to 64 exhibit incorrect behavior since self-state transitions are repeated again and again instead of next-state transitions.

The difference of the log probabilities of the top-2 sentences is too close even if the second path reaches the first state of the next model. These observations indicate that the sequence of phone models representing a sentence does not match the actual speech closely. Some models like “s+t” or “t+s”, on behalf of transition of two consonants, do not have negative effects if they are omitted in the model sequence during recognition. Combining this sort of model with a vowel as a phone model unit

is likely to improve the recognition system's performance due to reduction in number of models as well as a better representation for interactions in speech signals.

Conclusions

We have successfully constructed a multilingual speech recognition system based on the use of multiple acoustic models operating in parallel. The preliminary experimental results demonstrate its feasibility, though some more error analysis should be performed to enhance its performance. The major advantage of our multilingual speech recognition is the modularity. In other words, we can train several acoustic models for different language independently, and then put them in parallel to form the final multilingual speech recognition system. This can greatly simplify the program design and limit the unnecessary interactions between different acoustic models. The advantage of modularity certainly outweights its disadvantage of corresponding overhead of extra memory requirement and a little bit lengthened computation.

Reference

Lowerre B. (1976) *The HARPY speech recognition system.* PhD thesis, Dept. of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, USA.

Huang X., Acero A., and Hon H.-W. (2001) Chapter 12 of *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development.* Prentice Hall PTR, Upper Saddle River, New Jersey.

Rabiner L. and Juang B.-W. (1993) *Fundamentals of Speech Recognition.* Prentice Hall PTR, Upper Saddle River, New Jersey.

HTK (2002) *Hidden Markov Model Toolkit V3.1.* Speech Vision and Robotics Group of the Cambridge University Engineering Department. (http://htk.eng.cam.ac.uk/)

Jang J.-S., Sun C.-T. and Mizutani E. (1997) *Neural-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence* Prentice Hall PTR, Upper Saddle River, New Jersey.

TIMIT Acoustic-Phonetic Continuous Speech Corpus (http://www.ldc.upenn.edu/Catalog/LDC93S1.html)