Computer Speech and Language (1999) **13**, 125–141
Article No. csla.1998.0118
Available online at http://www.idealibrary.com on IDEAL

Interpolation of n-gram and mutual-information based trigger pair language models for Mandarin speech recognition

Zhou GuoDong and Lua KimTeng

Department of Computer Science, School of Computing, National University of Singapore, Lower Kent Ridge Road, Singapore 119260†

Abstract

While n-gram modeling is simple and dominant in speech recognition, it can only capture the short-distance context dependency within an n-word window where currently the largest practical n for natural language is three. However, many of the context dependencies in natural language occur beyond a three-word window. This paper proposes a new language modeling approach to capture the preferred relationships between words over a short or long distance through the concept of MI-Trigger pairs. Different MI-Trigger-based models are constructed in either a distance-dependent or a distance-independent way within a window from 1 to 10 words. This new MI-Trigger-based modeling is also compared and merged with word bigram modeling. It is found that the MI-Trigger-based modeling has better performance than word bigram modeling. It is also found that n-gram and MI-Trigger models have good complementarity and their proper merging can further increase the recognition rate when tested on Mandarin speech recognition. One advantage of MI-Trigger-based modeling is that the number of parameters needed for MI-Trigger modeling is much less than that of word bigram modeling. Another advantage is that the number of trigger pairs in an MI-Trigger model can be kept to a reasonable size without losing too much of its modeling power.

1. Introduction

In speech recognition, given an acoustic signal A, the goal is to find the linguistic hypothesis L that maximizes $P(L|A)$. Using Bayes Law:

$$\arg\max_L P(L|A) = \arg\max_L P(A|L) \cdot P(L). \tag{1}$$

As shown in Figure 1, $P(A|L)$ is estimated by an acoustic decoder, which compares A to its stored acoustic models of all speech units. Providing an estimate for $P(A|L)$ is the responsibility of the acoustic model while providing an estimate for $P(L)$ is the responsibility of the language model through a language decoder. In our continuous Mandarin speech recognition system, we use semicontinuous hidden Markov models (HMMs) to model acoustic units and adopt Viterbi beam search as the search algorithm.

†E-mail: {zhougd,luakt}@comp.nus.edu.sg

0885–2308/99/020125 + 17 $30.00/0

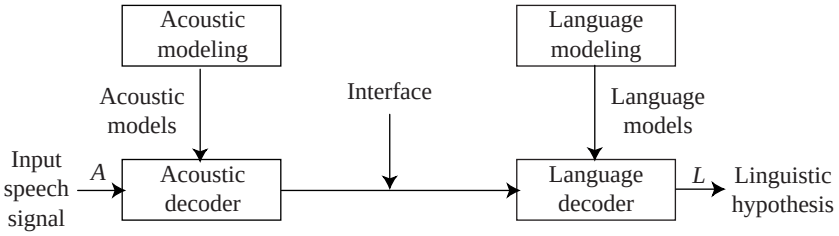

Figure 1. A simple structure of a speech recognition system.

In this paper, we focus on language modeling in Mandarin speech recognition. Language modeling is the attempt to characterize, capture and exploit the regularities and constraints in natural language. Proper language modeling is crucial to a speech recognition system. Among all the language modeling approaches, n-gram models have been the most widely used in speech recognition (Katz, 1987; Gale & Church, 1990; Brown *et al.*, 1992; Lee *et al.*, 1993; Lyu *et al.*, 1995; Yang *et al.*, 1996) and other applications. While n-gram models are simple and have been successfully used in speech recognition and other tasks, they have obvious deficiencies. For instance, n-gram models can only capture the short-distance dependency within an n-word window where currently the largest practical *n* for natural language is three and many kinds of dependencies in natural language occur beyond a three-word window. While we can use conventional n-gram models to capture the short-distance dependency, the long-distance dependency should also be exploited properly.

The purpose of this paper is to propose a new modeling approach to capture the context dependency between words over a short or long distance and use it in the application of Mandarin speech recognition.

The rest of this paper is organized as follows: an n-gram-based baseline continuous Mandarin speech recognition system is first described briefly in Section 2. Section 3 addresses the new MI-Trigger-based language modeling approach while its merging with an n-gram model is described in Section 4. Finally, Section 5 presents our conclusions.

2. Baseline continuous Mandarin speech recognition system

2.1. Acoustic decoder

As Mandarin Chinese is a tonal language, each syllable is composed of a base syllable and a tone. There are 408 different base syllables and five different tones which are high and level (1^{st}), rising (2^{nd}), falling and rising (3^{rd}), falling (4^{th}) and short light (5^{th}). Recognition of Chinese tonal syllable consists of base syllable recognition and tone recognition. For base syllable recognition, 14 cepstral and 14 delta-cepstral coefficients, energy (normalized) and delta-energy are used as feature parameters to form a feature vector with dimension 30, while for tone recognition, the pitch period and the energy together with their first order and second order delta coefficients are used to form a feature vector with dimension 6. All the acoustic units are modeled by semicontinuous HMMs. For base syllable recognition, 138 HMMs are used to model 100 context-dependent INITIALs and 38 context-independent FINALs while five HMMs are used to model the five different tones in Mandarin Chinese. Six sets of 5000 short sentences are used for training and another 600 sentences (6102 Chinese characters) from a Singapore school textbook are used for testing. All the training and testing data are

TABLE I. The top-n recognition rates of BSs (Base syllables)

Number of BS hypotheses	Top 1	Top 5	Top 10	Top 15	Top 20
Recognition rate of BS	88·2%	97·6%	99·2%	99·5%	99·8%

TABLE II. The recognition rates of the tones

	Tone 1	Tone 2	Tone 3	Tone 4	Tone 5
Tone 1	90·4%	0·8%	0·6%	0·8%	7·4%
Tone 2	8·3%	81·1%	5·4%	0·2%	4·9%
Tone 3	5·0%	20·9%	43·0%	29·1%	2·0%
Tone 4	4·3%	0·2%	1·8%	93·5%	0·2%
Tone 5	24·1%	8·6%	0·9%	8·2%	58·2%

TABLE III. The error sources of base syllable recognition

Error type	Type 1	Type 2	Type 3	Type 4
Error rate	0·98%	2·67%	1·52%	6·61%

speaker-dependent and are obtained in an office-like laboratory environment and digitized with a sampling frequency of 16 kHZ.

Recognition rates of base syllable and tone are shown in Table I and Table II, respectively.

In order to realize the error sources of base syllable recognition, all errors are classified into four types: (1) insertion errors; (2) deletion errors; (3) errors indirectly caused by insertion and deletion errors (Insertion and deletion errors may cause the neighboring base syllables to be recognized wrongly); and (4) others, errors except those caused directly and indirectly by insertion and deletion errors, i.e. the correct base syllable is not the best base syllable candidate in the n-best base syllable candidates of the corresponding segment.

Table III shows that about 44% of errors (5·2% of 11·8%) are related to insertion and deletion errors directly or indirectly.

2.2. Interface to language decoder

Given a speech recognizer and a language decoder, the next problem is how to combine the two components together. The classifications suggested in Murveit and Moore (1990), Harper *et al.* (1994), and Ward and Novick (1994), suggest four different approaches for combining a speech recognizer with a language decoder; Top-best Hypothesis, n-best Hypotheses, Speech Lattice and Parallel decoding.

Considering the advantages and disadvantages of these four interface approaches, we choose the third approach: speech lattice. However, to take account of the characteristics of Mandarin speech recognition, a new speech lattice is proposed. In the so-called n-best BST (Base Syllable + Tone) speech lattice, not only are the n-best paths (base syllable sequences with different segmentation) kept, but the n-best base syllable candidates are also reserved for every base syllable segment in the speech lattice. Finally, as stated in Section 2.1, recognition of Chinese tonal syllables consists of base syllable recognition and tone recognition. Therefore, base syllable recognition should be combined properly with the tone recognition.

In this paper, an n-best BS (Base Syllable) speech lattice is first constructed and then synchronized properly with tone recognition. The resultant speech lattice is, thereafter, called the n-best BST (Base Syllable + Tone) speech lattice.

2.2.1. *N-best BS speech lattice*

We compute the raw speech lattice using the Viterbi beam search algorithm. Since the actual raw lattice produced by the speech recognizer is often too large and redundant to be decoded directly, only the n-best paths (n-best base syllable sequences with different segmentation) in the speech lattice are kept while others are pruned. The value of n depends on the speech length. The longer the speech, the larger the value of n. Moreover, in order to increase the recognition rate of base syllables, a base syllable bigram model is used in the Viterbi beam search algorithm.

While keeping the n-best paths in the speech lattice, we find that, for the language decoder to find out the most probable word sequence, multiple base syllable candidates need to be reserved for any base syllable segment in the speech lattice, because the best candidate is not always correct. That is to say, for any base syllable segment in the speech lattice, not only must the most likely base syllable be reserved but the n-best base syllable candidates (i.e. $n = 20$) are also stored. Therefore, as long as the n-best paths are obtained, the n-best base syllable candidates of every base syllable segment in the n-best paths can be obtained simultaneously. This results in an n-best BS speech lattice.

2.2.2. *Synchronization of base syllable and tone recognition*

In this paper, tone recognition is a post process to the base syllable recognition. After the above steps, we obtain an n-best BS speech lattice. Then, the underlying pitch and energy of any base syllable segment are found, processed to form the feature parameters, and scored by the five acoustic models representing the five tones in Mandarin. With this method, base syllable recognition and tone recognition are synchronized properly in the so-called n-best BST speech lattice.

In this way, we have a speech lattice of a size that can be decoded within a reasonable time and space with possibly minimal segmentation error rate, including insertion and deletion errors, and with a high syllable inclusion rate.

As an example, a small portion of a typical n-best BST speech lattice of test sentence "记忆力增强了" ("The faculty of memory has strengthened") is shown in Figure 2. The nodes of the graph represent the start or end points of a lattice syllable. The base syllable candidates (for simplicity, only the best three base syllable candidates are shown in the graph) with their associated scores which are derived from the base syllable recognition, are marked on the top side of every edge in the graph, while the five tone scores which are derived from the tone recognition and normalized by the maximum tone probability, are marked on the bottom side of the graph edge. Here, all the scores are logarithmic probabilities derived from the acoustic decoder.

For any segment (i, j) in the n-best BST speech lattice, the n-best BSSs (Base Syllable Strings) can be retrieved by using a Viterbi algorithm. For each BSS, we look up a Chinese PINYIN-to-Character conversion table. In this way, an n-best BST speech lattice can be converted into an n-best word lattice. Figure 3 shows a small portion of the n-best word lattice related to the n-best BST speech lattice as shown in Figure 2.

2.3. *Language decoder*

In the baseline system, a word bigram model is used in the language decoder. The lexicon includes 28 000 words. The word bigram model is trained on a XinHua news corpus of 29 million words (automatically segmented) and contains 6·1 million bigram items. In order to overcome the problem of sparse data, the bigram model is combined with the unigram model by the backoff method employed by Katz (1987).

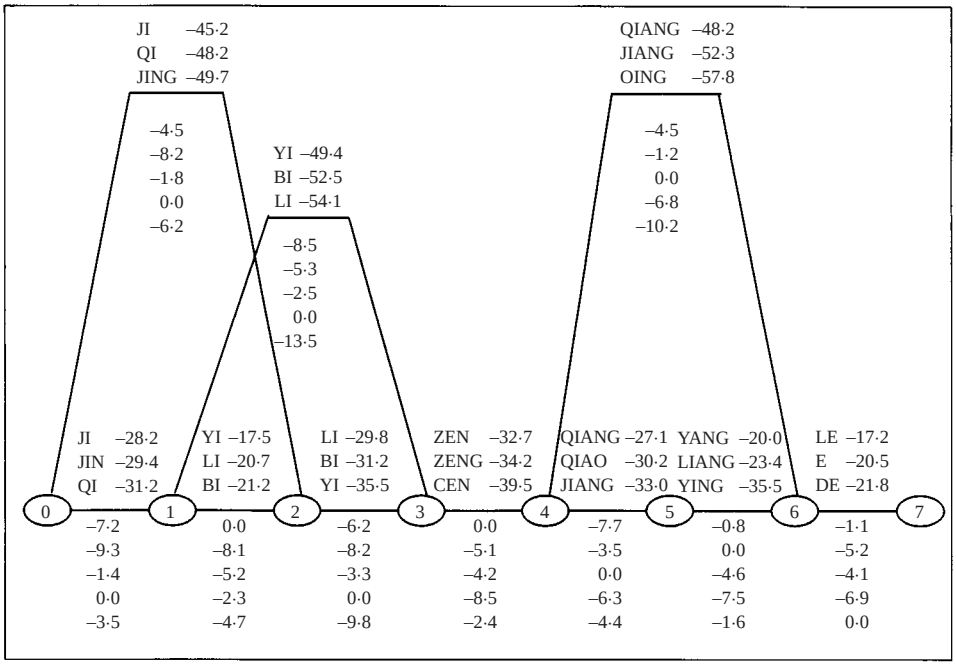

Figure 2. A small portion of an n-best BST speech lattice for the test sentence '记忆力增强了' (JI4 YI4 LI4 ZENG1 QIANG2 LE5).

TABLE IV. The recognition rates (error rates) of Chinese characters using the word bigram model

Model	Top-best BST speech lattice	n-best BST speech lattice
Word bigram	86·2% (13·8%)	89·6% (10·4)

In order to illustrate the advantages of directly decoding the n-best BST speech lattice, the n-best BST speech lattice is compared to the top-best one except that only the top-best path is obtained. The experimental results are shown in Table IV. The figures in the parentheses of Table IV show the overall Chinese character error rates. It shows that decoding the n-best BST speech lattice has much better performance than decoding the top-best one.

In order to identify different sources of the recognition errors, all of the recognition errors are classified into five types: (1) insertion errors; (2) deletion errors; (3) correct BS and correct tone (homophone); (4) correct BS and wrong tone; and (5) wrong BS.

Here, the sources of errors in decoding the top-best and the n-best BST speech lattice both using the same word bigram model are evaluated in Table V. This indicates that the reason why n-best speech lattice is superior is largely due to the fact that most of the insertion and deletion errors can be corrected by using the n-best BST speech lattice.

3. Trigger-based language modeling

In natural language there always exist many preferred relationships between words. Two highly associated word pairs are "not only/but also" and "doctor/nurse". Psychological experiments in Meyer *et al.* (1975) indicated that the human's reaction to a highly associated word pair was stronger and faster than that to a poorly associated word pair. We can obtain useful preference

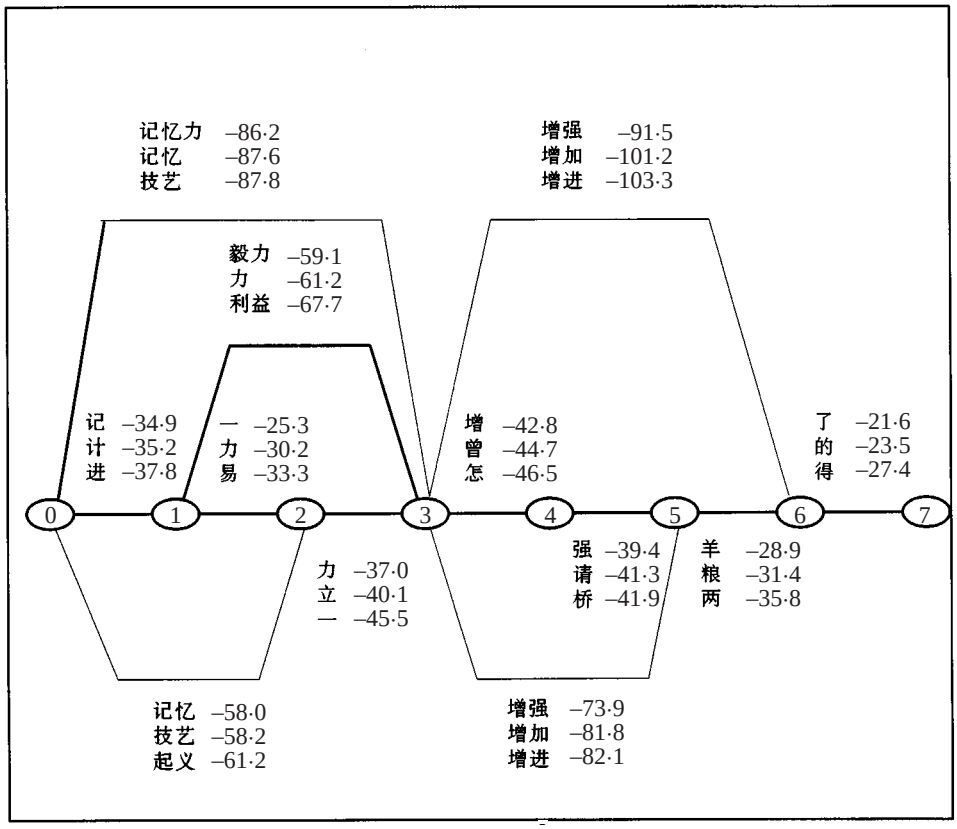

Figure 3. A small portion of an n-best word lattice.

TABLE V. The statistics of the recognition errors using the word bigram model

Error type	Top-best BST speech lattice	n-best BST speech lattice
Insertion error	0·94%	0·24%
Deletion error	2·59%	0·73%
Correct BS and correct tone	2·20%	2·12%
Correct BS and wrong tone	2·92%	2·84%
Wrong BS	4·52%	4·39%

information from the corpus, such as the semantic preference between noun and noun (e.g. "doctor/nurse"), the particular preference between adjective and noun (e.g. "strong/currency"), and solid structure (e.g. "pay/attention") (Calzolori, 1990). These sources of information are useful for automatic sentence disambiguation. Similar research includes Church *et al.* (1991), Magerman *et al.* (1990), Brent (1993), Hindle *et al.* (1993), Kobayashi *et al.* (1994), and Rosenfeld (1994).

The preference relationships between words can expand from a short to long distance. While n-gram models are simple in language modeling, n-gram models can only capture the short-distance dependency within an n-word window where currently the largest practical

n for natural language is three and many kinds of dependencies in natural language occur beyond a three-word window. While we can use conventional n-gram models to capture the short-distance dependency, the long-distance dependency should also be exploited properly.

Based on the above, we decided to use the trigger pair employed in Rosenfeld (1994) as the basic concept for extracting the word association information of an associated word pair over a short or long distance. If a word A is highly associated with another word B, then $(A \rightarrow B)$ is considered a "trigger pair", with A being the trigger and B the triggered word. When A occurs in the document, it triggers B, causing its probability estimate to change. A and B can be also extended to word sequences. For simplicity, here we will concentrate on the trigger relationships between single words although the ideas can be extended to longer word sequences.

In order to build a trigger-based language model, there are two main problems to be solved:

(1) how to select a trigger pair?
(2) how to measure a trigger pair?

3.1. How to select a trigger pair

Even if we can restrict our attention to the trigger pair (A, B) where A and B are both single words, the number of such pairs is too large. Therefore, selecting a reasonable number of the most powerful trigger pairs is important to a trigger-based language model.

3.1.1. Window size

The most obvious way to control the number of the trigger pairs is to restrict the window size, which is the maximum distance between the trigger pair. In order to decide on a reasonable window size, we must know how much the distance between the two words in the trigger pair affects the word probabilities. Here, we construct the long-distance word bigram (WB) models (Huang *et al.*, 1993; Wright *et al.*, 1993) for distance-$d = 1, 2, \ldots, 100$, $P_d(w_i|w_{i-d})$. Adjacent words have distance $d = 1$. These models attempt to capture the dependence of the predicted word on a word which is some distance back directly. For example, a distance-3 bigram $P_3(w_i|w_{i-3})$ predicts w_i based on w_{i-3}. As a special case, distance-1 bigrams are the familiar conventional bigrams. The distance-100 is used as a control, since we expect no significant information after that distance. We compute the conditional perplexity (Shannon, 1951) for each long-distance WB model. Conditional perplexity is a measure of the average number of possible choices there are for a conditional distribution. The conditional perplexity of a conditional distribution with conditional entropy $H(Y|X)$ is defined to be $2^{H(Y|X)}$. Conditional entropy is the entropy of a conditional distribution. Given two random variables X and Y, a conditional probability mass function $P_{Y|X}(y|x)$, and a marginal probability mass function $P_Y(y)$, the conditional entropy of Y given X, $H(Y|X)$ is defined as:

$$H(Y|X) = -\sum_{x \in X} \sum_{y \in Y} P_{X,Y}(x, y) \log_2 P_{Y|X}(y|x). \tag{2}$$

For a large enough corpus, the conditional perplexity is usually an indication of the amount of information conveyed by the model: the lower the conditional perplexity, the more information it conveys and the better is the model. This is because the model captures as much as it can of that information, and whatever uncertainity remains shows up in the conditional perplexity. Here, the training corpus is the XinHua corpus, which has about 57 million characters or 29 million words.

TABLE VI. Perplexities of the long-distance WB models for different distances

Distance	Perplexity	Distance	Perplexity
1	230	7	1479
2	575	8	1531
3	966	9	1580
4	1157	10	1599
5	1307	11	1611
6	1410	100	1674

From Table VI we find that the conditional perplexity is lowest for $d = 1$, and it increases significantly as we move through $d = 2, 3, 4, 5$ and 6. For $d = 7, 8, 9, 10$ and 11, the conditional perplexity increases slightly. We conclude that significant information exists only in the last six words of the history. However, in this paper we nevertheless restrict the maximum window size to 10.

3.1.2. How to select a trigger pair

A simple way to assess the significance of the correlation between the two words A_o and B in the trigger $(A_o \rightarrow B)$ is to measure their cross product ratio (CPR). Here, B is the next word and A_o occurs somewhere in the document's history. One often used measure is the logarithmic measure of that quality, which has units of bits and is defined as:

$$\log CPR(A_o, B) = \log \frac{P(A_o, B)P(\overline{A_o}, \overline{B})}{P(A_o, \overline{B})P(\overline{A_o}, B)}, \tag{3}$$

where $P(X_0, Y)$ is the probability of a word pair (X_0, Y) occurring in the window.

Although the cross product ratio measure is simple, it is not sufficient to determine the utility of a proposed trigger pair. Consider a highly correlated pair consisting of two rare words "树梢→白皑皑" ("tail of tree/pure white"), and compare it to a less well correlated, but more common pair "医生→护士" ("doctor/nurse"). An occurrence of the word "树梢" ("tail of tree") provides more information about the word "白皑皑" ("pure white") than an occurrence of the word "医生" ("doctor") about the word "护士" ("nurse"). Nevertheless, since the word "医生" ("doctor") is likely to be much more common in the test data, its average utility may be much higher. If we can afford to incorporate only one of the two pairs into our trigger-based model, the trigger pair "医生→护士" ("doctor/nurse") may be preferable.

Therefore, an alternative measure of the expected benefit provided by A_o in predicting B is the average mutual information (AMI) between the two:

$$\begin{aligned}\mathrm{AMI}(A_o; B) = {} & P(A_o, B) \log \frac{P(A_o B)}{P(A_o)P(B)} + P(A_o, \overline{B}) \log \frac{P(A_o \overline{B})}{P(A_o)P(\overline{B})} \\ & + P(\overline{A_o}, B) \log \frac{P(\overline{A_o} B)}{P(\overline{A_o})P(B)} + P(\overline{A_o}, \overline{B}) \log \frac{P(\overline{A_o B})}{P(\overline{A_o})P(\overline{B})}.\end{aligned} \tag{4}$$

Obviously, Equation (4) takes the joint probability into consideration. We use this equation to select the trigger pairs. In related works, Rosenfeld (1994) used this equation and Church *et al.* (1991) used a variant of the first term to automatically identify the associated word pairs.

3.2. How to measure a trigger pair

For a trigger pair $(A_o \rightarrow B)$, mutual information $\mathrm{MI}(A_o; B)$ reflects the degree of the preference relationship between the two words in the trigger pair, which can be computed as

follows:

$$\mathrm{MI}(A_o;\, B) = \log \frac{P(A_o,\, B)}{P(A_o) \cdot P(B)}, \tag{5}$$

where $P(X)$ is the probability of the word X occurring in the corpus and $P(A, B)$ is the probability of the word pair (A, B) occurring in the window.

Several properties of mutual information are apparent:

- $\mathrm{MI}(A_o;\, B)$ is different from $\mathrm{MI}(B_0;\, A)$, i.e. it is order dependent.
- If A_o and B are independent, then $\mathrm{MI}(A;\, B) = 0$.

$\mathrm{MI}(A_o;\, B)$ reflects the change of the information content when the two words A_o and B are correlated. That is to say, the higher the value of $\mathrm{MI}(A_o;\, B)$, the stronger affinity the words A_o and B have. Therefore, we use mutual information to measure the preference relationship degree of a trigger pair.

3.3. Trigger-based language modeling

As discussed above, we can restrict the number of the trigger pairs using a reasonable window size, select the trigger pairs using average mutual information and then measure the trigger pairs using mutual information. In this section, we will describe in greater detail about how to build a trigger-based model. As the triggers are mainly determined by mutual information, we call them MI-Triggers. To build a concrete MI-Trigger model, two factors have to be considered.

One factor is the window size. As we have restricted the maximum window size to 10, we will experiment on 10 different window sizes ($ws = 1, 2, \ldots, 10$).

Another one is whether to measure an MI-Trigger in a distance-independent (DI) or distance-dependent (DD) way. While a DI MI-Trigger model is simple, a DD MI-Trigger model has the potential of modeling the word association better and is expected to have better performance because many of the trigger pairs are distance-dependent. We have studied this issue using the XinHua corpus of 29 million words by creating an index file that contains, for every word, a record of all of its occurrences with DD co-occurrence statistics. Some examples are shown in Table VII, which shows that "越 / 越" ("the more/the more") has the highest correlation when the distance is 2, that "不但→而且" ("not only/but also") has the highest correlation when the distances are 3, 4 and 5, and that "医生→护士" ("doctor/nurse") has the highest correlation when the distances are 1 and 2. After manually browsing hundreds of the trigger pairs, we draw following conclusions:

- Different trigger pairs display different behaviors.
- Behaviors of trigger pairs are distance-dependent and should be measured in a DD way.
- Most of the potential of triggers are concentrated on high-frequency words. For example, "医生→护士" ("doctor/nurse") is indeed more useful than "树梢→白皑皑" ("tail of tree/pure white").

To compare the effects of the above two factors, 20 MI-Trigger models (in which DI and DD MI-Trigger models with a window size of 1 are the same) are built. Each model differs in different window sizes, and whether the evaluation is done in the DI or DD way. Moreover, for ease of comparison, each MI-Trigger model includes the same number of the best trigger pairs. In our experiments, only the best 1 million trigger pairs are included. Experiments to determine the effects of different numbers of the trigger pairs in a trigger-based model will be conducted later.

TABLE VII. The occurrence frequency of word pairs as a function of distance

Distance	越/ 越 (the more/the more)	不但/ 而且 (not only/but also)	医生/ 护士 (doctor/nurse)
1	0	0	24
2	3848	5	15
3	72	24	1
4	65	18	1
5	45	14	0
6	45	4	0
7	40	2	0
8	23	3	0
9	9	2	1
10	8	4	0

TABLE VIII. The numbers of the trigger pairs for different distances in the DD-6-MI-Trigger model

Distance	1	2	3	4	5	6
Number of MI-Trigger pairs ($\times$1000)	246	219	180	146	117	92

For simplicity, we represent a trigger pair as XX-ws-MI-Trigger, and call a trigger-based model as the XX-ws-MI-Trigger model, where XX represents DI or DD and ws represents the window size. For example, the DD-6-MI-Trigger model represents a DD MI-Trigger-based model with a window size of six.

All the models are built on the XinHua corpus of 29 million words. To take the DD-6-MI-Trigger model as a example, we filter about $28 \times 28 \times 6 \times 10^6$ (with six different distances and with about 28 000 Chinese words in the lexicon) possible DD word pairs. As a first step, only word pairs that co-occur at least three times are kept. This results in 5.7 million word pairs. Then the best 1 million word pairs are kept as trigger pairs by selection using average mutual information. Finally, the best 1 million MI-Trigger pairs are measured by mutual information. In this way, we build a DD-6-MI-Trigger model which includes the best 1 million trigger pairs. Some statistics about the DD-6-MI-Trigger model are shown in Table VIII, Table IX and Table X. These tables show the number of trigger pairs for different distances, average mutual information and mutual information, respectively.

As the MI-Trigger-based models measure the trigger pairs using mutual information which only reflects the change of information content when the two words in the trigger pair are correlated, a word unigram model is combined with them. Given $S = w_1 w_2 \dots w_n$, we can estimate the logarithmic probability $\log P(S)$.

For a DI- ws-MI-Trigger-based model,

$$\log P(S) = \sum_{i=1}^{n} \log P(w_i) + \sum_{i=n}^{2} \sum_{j=i-1}^{\max(1,\, i-ws)} \text{DI-}ws\text{-MI-Trigger}\ (w_j \rightarrow w_i) \cdot \qquad (6)$$

For a DD-ws-MI-Trigger-based model,

$$\log P(S) = \sum_{i=1}^{n} \log P(w_i) + \sum_{i=n}^{2} \sum_{j=i-1}^{\max(1,\, i-ws)} \text{DD-}ws\text{-MI-Trigger}(w_j \rightarrow w_i, i-j+1), \qquad (7)$$

TABLE IX. The numbers of the trigger pairs for different average mutual information in the DD-6-MI-Trigger model

Average MI ($\times 10^{-7}$)	Number of trigger pairs	Average MI ($\times 10^{-7}$)	Number of trigger pairs	Average MI ($\times 10^{-7}$)	Number of trigger pairs
[1,2)	183 677	[21,22)	3463	[41,42)	873
[2,3)	270 035	[22,23)	3161	[42,43)	827
[3,4)	137 520	[23,24)	2908	[43,44)	772
[4,5)	84 219	[24,25)	2665	[44,45)	763
[5,6)	56 935	[25,26)	2441	[45,46)	676
[6,7)	41 025	[26,27)	2234	[46,47)	651
[7,8)	30 711	[27,28)	2095	[47,48)	648
[8,9)	23 517	[28,29)	1835	[48,49)	629
[9,10)	18 590	[29,30)	1788	[49,50)	644
[10,11)	15 240	[30,31)	1675	[50,51)	613
[11,12)	12 424	[31,32)	1599	[51,52)	554
[12,13)	10 482	[32,33)	1500	[52,53)	559
[13,14)	8727	[33,34)	1419	[53,54)	538
[14,15)	7710	[34,35)	1307	[54,55)	485
[15,16)	6845	[35,36)	1197	[55,56)	503
[16,17)	5946	[36,37)	1119	[56,57)	451
[17,18)	5368	[37,38)	1126	[57,58)	462
[18,19)	4644	[38,39)	972	[58,59)	432
[19,20)	4289	[39,40)	1001	[59,60)	412
[20,21)	3819	[40,41)	922	>= 60	20 358

TABLE X. The numbers of the trigger pairs for different mutual information in the DD-6-MI-Trigger model

MI	Number of trigger pairs	MI	Number of trigger pairs
[0,1)	9222	[10,11)	11 231
[1,2)	45 368	[11,12)	6814
[2,3)	300 324	[12,13)	3950
[3,4)	201 059	[13,14)	2116
[4,5)	154 631	[14,15)	1031
[5,6)	102 850	[15,16)	481
[6,7)	68 055	[16,17)	190
[7,8)	45 186	[17,18)	56
[8,9)	29 124	[18,19)	6
[9,10)	18 306	>= 19	0

where ws is the windows size and $i - j + 1$ is the distance between the words w_i and w_j. The first item in each of Equations (6) and (7) is the logarithmic probability of S using a word unigram model, and the second one is the value contributed to mutual information of the MI-Trigger pairs in the MI-Trigger model. DI-ws-MI-Trigger() and DD-ws-MI-Trigger() of a trigger pair are calculated by mutual information which are given by Equation (5). The reason for adding the unigram logarithmic probability is that the mutual information only reflects the change of the information content.

Compared with the long-distance bigram models employed in Huang *et al.* (1993) and Wright *et al.* (1993), the similarity between the long-distance bigram models and MI-Trigger-based models is that they are all used to capture the long-distance dependency of word pairs. Moreover, as stated in Section 3.1.1, the perplexities of long-distance bigram models are computed to estimate the window size for the trigger pair. One difference between the long-distance bigram models and MI-Trigger-based models lies in the fact that only the word pairs whose mutual information are positive are included in MI-Trigger-based models as shown in

TABLE XI. The perplexities and recognition rates of the 20 MI-Trigger models

Window Size	Distance-independent		Distance-dependent	
	Perplexity	Recog. rate	Perplexity	Recog. rate
1	301	87·2%	301	87·2%
2	288	88·0%	259	88·9%
3	280	88·3%	238	89·5%
4	272	88·6%	221	89·9%
5	267	88·8%	210	90·1%
6	262	88·9%	201	90·2%
7	270	88·5%	216	90·0%
8	275	88·2%	227	89·5%
9	282	88·1%	241	89·5%
10	287	87·9%	252	89·2%

Table X. Another difference lies in the fact that only the word pairs whose average mutual information are biggest are included in the MI-Trigger-based models shown in Table IX and Section 3.1.2. Finally, different MI-Triggers with different distance can be easily merged as shown in Equations (6) and (7).

In order to measure the efficiency of the MI-Trigger-based models, the conditional perplexities of the 20 different models (each has 1 million trigger pairs) are computed from the XinHua corpus of 29 million words and are shown in Table XI while the character recognition rates of decoding the n-best BST speech lattice using different MI-Trigger models are also shown in Table XI.

Table XI shows that

- The DD MI-Trigger models have better performances than the DI MI-Trigger models for the same window size.
- In both the DI or DD methods, the MI-Trigger model of distance 6 has the best performance. This means that most of the context dependency information exists within a window of 6. The DD-6-MI-Trigger model has the best performance.

As stated above, the MI-Trigger models only include the best 1 million trigger pairs. One may ask: what is a reasonable number of the trigger pairs that an MI-Trigger model should include? Here, we will examine the effect of different numbers of trigger pairs in an MI-Trigger model. We also use the DD-6-MI-Trigger model. The character recognition rates in decoding the n-best BST speech lattice are shown in Table XII.

We can see from Table XII that the recognition rate rises quickly as the number of MI-Trigger pairs increase from 100 000 to 800 000 and then it rises slowly. Therefore, the best 800 000 trigger pairs should at least be included. In our models we adopt the best 1 million MI-trigger pairs.

It is clear that MI-Trigger-based language modeling has much better performance than the word bigram modeling. One advantage of the MI-Trigger-based language model is that the number of parameters is much lower than that of the word bigram model. Another advantage of MI-Trigger modeling is that the number of trigger pairs can be reasonable in size without losing too much of its modeling power. Although MI-Trigger modeling has the disadvantage of capturing only highly correlated word pairs, it does not have the problem of zero frequency that occurs in n-gram modeling.

TABLE XII. The effect of different numbers of the trigger pairs in the DD-6-MI-Trigger model on the recognition rates

Number of the MI-Trigger pairs DD-6-MI-Trigger model	Perplexity	Recognition rate
0	1967	80·4%
100 000	672	84·3%
200 000	358	86·5%
400 000	293	88·1%
600 000	260	89·3%
800 000	224	89·9%
1 000 000	201	90·2%
1 200 000	197	90·4%
1 400 000	194	90·5%
1 600 000	191	90·6%
1 800 000	188	90·7%
2 000 000	186	90·8%
2 500 000	184	90·8%
3 000 000	183	90·9%
3 500 000	182	90·9%
4 000 000	181	91·0%
4 500 000	179	91·1%
5 000 000	178	91·1%
6 000 000	175	91·2%

4. n-gram and trigger model merging

One of the most powerful abilities of a person is to use language models implicitly and subconsciously by properly ranking different language constraints from different knowledge sources (though only partially) when processing natural language. Similarly, properly combining different language constraints from different knowledge sources is also beneficial to automatic speech recognition.

In this section we will check the effects of combining different models together. This can be done by using linear interpolation.

Given $S = w_1^n = w_1 w_2 \dots w_n$ and k language models $\{P_i(S), i = 1, 2, \dots, k, \}$, we can combine them linearly with:

$$\log P_{\text{MERGED}}(S) = \sum_{i=1}^{k} a_i \cdot \log P_i(S), \text{ where } 0 \le a_i \le 1 \text{ and } \sum_{i=1}^{k} a_i = 1. \tag{8}$$

Here, two language models are used:

- $P_1(S) = P_{\text{N-GRAM}}(S)$ is the probability scored by the same word bigram model as described in Section 2, and a_1 is its model weight.
- $P_2(S) = P_{\text{MI}}(S)$ is the probability scored by the MI-Trigger model and a_2 is its model weight. Here the DD-6-MI-Trigger model, which includes 1 million trigger pairs, is applied.

Here, we use the concept of complementarity to evaluate the potential for combining two language models M_1 and M_2, which is defined as:

$$CP_{M_1+M_2} = \frac{C_{M_1} \cup C_{M_2}}{S} - \frac{C_{M_1} \cap C_{M_2}}{S} - \left| \frac{C_{M_1}}{S} - \frac{C_{M_2}}{S} \right|, \tag{9}$$

where

$$\begin{aligned} S &= \{\text{character set of test sentences}\},\\ C_{M_1} &= \{\text{correctly recognized characters by } M_1\},\\ C_{M_2} &= \{\text{correctly recognized characters by } M_2\},\\ C_{M_1} \cup C_{M_2} &= \{\text{correctly recognized characters by either } M_1 \text{ or } M_2\},\\ C_{M_1} \cap C_{M_2} &= \{\text{correctly recognized characters by both } M_1 \text{ and } M_2\}. \end{aligned}$$

Obviously, $\frac{C_{M_1} \cup C_{M_2}}{S}$ is the recognition accuracy rate by either M_1 or M_2 and $\frac{C_{M_1} \cap C_{M_2}}{S}$ is the recognition accuracy rate by both M_1 and M_2. Clearly, the value of the complementarity gives us an idea of the potential performance improvement we can expect from the merged model. The larger the complementarity, the more improved performance the merged model is expected to have.

By examining the results presented in Sections 2 and 3, we have:

$$\frac{C_{\text{N-GRAM}}}{S} = 89{\cdot}6\%, \quad \frac{C_{\text{MI}}}{S} = 90{\cdot}2\%,$$
$$\frac{C_{\text{N-GRAM}} \cup C_{\text{MI}}}{S} = 94{\cdot}0\%, \quad \frac{C_{\text{N-GRAM}} \cap C_{\text{MI}}}{S} = 85{\cdot}8\%.$$

Using Equation (8), we can compute the complementarity of the WB and DD-6-MI-Trigger models:

$$CP_{\text{N-GRAM+MI}} = \frac{C_{\text{N-GRAM}} \cup C_{\text{MI}}}{S} - \frac{C_{\text{N-GRAMt}} \cap C_{\text{MI}}}{S} - \left| \frac{C_{\text{N-GRAM}}}{S} - \frac{C_{\text{MI}}}{S} \right| = 7{\cdot}6\%.$$

From above, we can assume that n-gram and MI-Trigger models have good complementarities with each other. Therefore, better performance can be expected by merging them together.

Experiments are also done on the same 600 Chinese sentences. The recognition rates for various values of bigram weight are shown in Figure 4.

It is shown in Figure 4 that

- The recognition rate reaches up to 91·8% when the bigram weight is 0·4 and the MI-Trigger weight is 0·6.
- The recognition rate of the best merged model is 1·6% (absolutely) higher than the MI-Trigger model and 2·2% (absolutely) higher than the bigram model.

Since the word trigram model is frequently used in current research, experiments have also been done on word trigram model and MI-Trigger merging. Here, the same DD-6-MI-Trigger with 1 million trigger pairs are used. The word trigram model is also trained on the XinHua news corpus of 29 million words (automatically segmented) and contains 9·7 million trigram items. In order to overcome the problem of sparse data, the trigram model is combined with the bigram and unigram models by the backoff method employed in Katz (1987). The recognition rates for various values of trigram weight are shown in Figure 5.

It is shown in Figure 5 that

- The recognition rate reaches up to 92·0% when the trigram weight is 0·5 and the MI-Trigger weight is 0·5.
- The recognition rate of the best merged model is 1·8% (absolutely) higher than the MI-Trigger model and 1·5% (absolutely) higher than the trigram model.

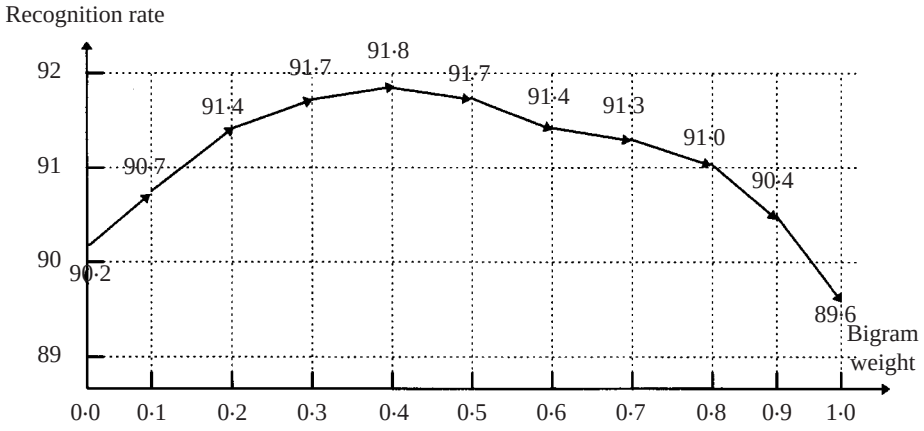

Figure 4. The recognition rates of word bigram and MI-Trigger merging.

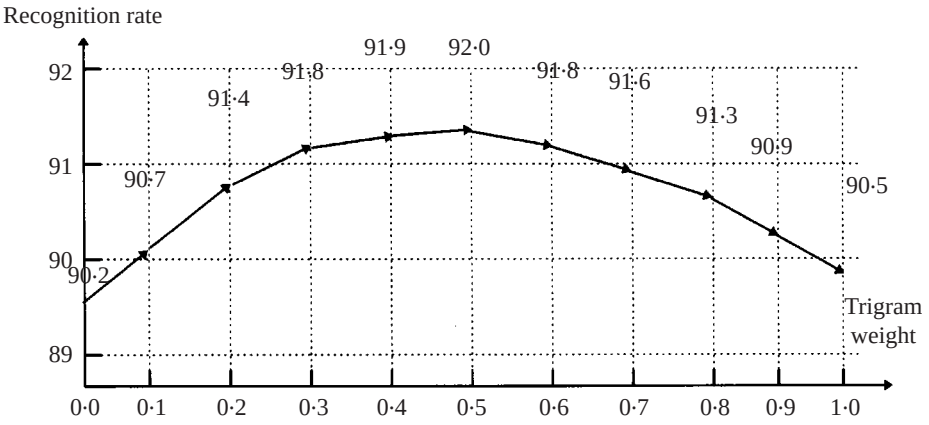

Figure 5. The recognition rates of word trigram and MI-Trigger merging.

Through the experiments, it has been proven that the merged model has better results over both n-gram and MI-Trigger models. Compared to the n-gram model, the merged model also captures the long-distance dependency of word pairs using the concept of mutual information. Compared to the MI-trigger model which only captures highly correlated word pairs, the merged model also captures poorly correlated word pairs within a short distance by using the n-gram model.

5. Conclusion

This paper proposes a new MI-Trigger-based modeling approach to capture the preferred relationships between words over a short or long distance by using the concept of trigger pair. In order to restrict the numbers of trigger pairs, not only is average mutual information of a trigger pair used, but the long-distance word bigram models are also proposed to estimate the window size for the trigger pair. Then the utility of a trigger pair is measured by its mutual information.

Both DI and DD MI-Trigger-based models are constructed within a window of a size from 1 to 10. These are also compared and merged with word bigram models. It is found that

- The DD MI-Trigger models have better performance than the DI MI-Trigger models for the same window size. Therefore, it is better to model the preferred relationships between words in a DD way.
- The number of trigger pairs in an MI-Trigger model can be kept to a reasonable size without losing too much modeling power.
- The MI-Trigger-based language modeling has better performance than the word bigram model while the parameter number of the MI-Trigger model is much less than that of the word bigram model.
- n-gram and MI-Trigger model merging further increases the recognition rate of our Mandarin speech recognition system. This shows that they have very good complementarity.
- The recognition rate of word bigram modeling is 89·6% while proper MI-Trigger modeling can reach up to 91·2% (with 6 million trigger pairs). The recognition rate further reaches 91·8% by proper merging of a word bigram model and a MI-Trigger mode (including only 1 million trigger pairs).

References

Brent, M. (1993). From grammer to lexicon: unsupervised learning of lexical syntax. *Computational Linguistics* **19**, 263–311.

Brown, P. F., Della Pietra, V. J., de Souza, P. V., Lai, J. C. & Mercer, R. L. (1992). Class-based n-gram models of natural language. *Computational Linguistics* **18**, 467–479.

Calzolori, N. (1990). Acqusition of lexical information from a large textual Italian corpus. *Proceedings of COLING*, Helsinki, Finland, August 1990, vol. 2, pp. 54–59.

Church, K. W. & Gale, W. A. (1991). Enhanced good turing and cat-cal: two new methods for estimating probabilities of English bigrams. *Computer Speech and Language* **5**, 19–54.

Gale, W. A. & Church K. W. (1990). Poor estimates of context are worse than none. *Proceedings of DARPA Speech and Natural Language Workshop*, Hidden Valley, PA, June 1990, pp. 293–295.

Harper, M. P. *et al.* (1994). Integrating language models with speech recognition. *Proceedings of AAAI Workshop on Integration of Natural Language and Speech Processing*, Seattle, WA, 31 July–4 August 1994, pp. 139–145.

Hindle, D. & Rooth, M. (1993). Structural ambiguity and lexical relations. *Computational Linguistics* **19**, 103–120.

Huang, X. D., Alleva, F., Hon, H. W., Hwang, M. Y., Lee, K. F. & Rosenfeld, R. (1993). The SPHINX-II speech recognition system: an overview. *Computer Speech and Language* **7**, 137–148.

Katz, S. M. (1987). Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing* **ASSP-35**, 400–401.

Kobayashi, T., Tokunaga, T. & Tanaka, H. (1994). Analysis of Japanese compound nouns using collocational information. *Proceedings of COLING*, Kyoto, Japan, 5–9 August 1994, vol. 6, pp. 865–874.

Lee, L. S., Chen, K. J., Lyu, R. Y., Shen, J. L., Wang, H. M. & Chien, L. F. (1993). Golden Mandarin(II) — an improved single chip real-time Mandarin dictation machine for Chinese language with very large vocabulary. *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, Minneapolis, MN, 27–30 April 1993, vol. II, pp. 503–506.

Lyu, R. Y. *et al.* (1995). Golden Mandarin(III) — a user-adaptive prosodic-segment-based Mandarin dictation machine for Chinese Language with very large vocabulary. *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, 8–12 May 1995, vol. 1, Detroit, MI, USA, pp. 57–60.

Magerman, D. & Marcus, M. (1990). Parsing a natural language using mutual information statistics. *Proceedings of AAAI*, vol. 1, Boston, MA, 29 July–3 August 1990, pp. 984–989.

Meyer, D., Schvaneveldt, R. & Ruddy, M. (1975). Loci of contextual effects on visual word recognition. In *Attention and Performance V*. (P. Rabbitt and S. Dornie, eds) pp. 98–116. Academic Press, New York.

Murveit, H. & Moore (1990). Integrating natural language constraints into HMM-based speech recognition. *Proceedings of International Conference on Acoustic, Speech and Signal Processing*, vol. 1, Albuquerque, NM, April 1990, pp. 573–576.

Rosenfeld, R. (1994). Adaptive statistical language modeling: a maximum entropy approach, PhD Thesis, Carneige Mellon University, Boston, MA.

Shannon, C. E. (1951). Prediction and entropy of printed English. *Bell Systems Technical Journal* **30**, 50–64.

Ward, K. & Novick, D. G. (1994). On the need for a theory of integration of knowledge sources for spoken language understanding. *Proceedings of AAAI Workshop on Integration of Natural Language and Speech Processing*, Seattle, WA, 31 July–4 August 1994, pp. 23–30.

Wright, J. H., Jones, G. J. F. & Lloyd-Thomas, H. (1993). A consolidated language model for speech recognition. *Proceedings of EuroSpeech*, vol. 2, Berlin, Germany, September 1993, pp. 977–980.

Yang, Y. J., Chen, L. F. & Lee, L. S. (1996). Adaptive linguistic decoding system for Mandarin speech recognition applications. *Computer Processing of Chinese & Oriental Languages* **10**, 211–224.

(Received 31 March 1998 and accepted for publication 27 July 1998)