A Corpus-based Chinese Syllable-to-Character System

CHIEN-PANG WANG AND TYNE LIANG*

Department of Computer and Information Science,

National Chiao Tung University,

1001 Ta Hsueh Rd., Hsinchu Taiwan 30050 R.O.C.

**:responsible for all correspondences.*

Keywords: *Mandarin, Phoneme, Homonym, Tolerance.*

Abstract

One of the popular input systems is based on Chinese phonetic symbols. Designing such kind of a syllable-to-character (STC) input system involves two major issues, namely, fault tolerance handling and homonym resolution. In this paper, the fault tolerance mechanism is constructed on the basis of a user-defined confusing set and a modified bucket indexing scheme is incorporated so as to satisfy real-time requirement. Meanwhile the homonym resolution is handled by binding force and heuristic selection rules. Both the system performance and tolerance ability are justified with real corpus in terms of searching speed and character conversion accuracy rate. Experimental results show that the proposed scheme can achieve 93.54% accuracy for zero-error syllable inputs and 80.13% for zero-tone syllable inputs. Furthermore both robustness and tolerance of the proposed system are proved for high input error rates.

1 Introduction

Among various kinds of Chinese input methods, the most popular one is based on phonetic symbols. This is because most of Chinese-speaking users are taught to use phonetic symbols in their elementary schools when they learn Chinese. However a syllable-to-character (STC) system is inherently associated with the serious homonym and similarly-pronounced phoneme problems. This is because a single syllable may correspond to several Chinese characters and there are indeed several Mandarin syllables which are sounded similarly. So it is not easy for users or acoustic recognizer to distinguish them when they are used. We call these syllables as confusing syllables. For example, syllable ㄕˋ (shih4) and ㄙˋ (szu4) are sounded similarly in speaking and listening, and a user might treat ㄕˋ (shih4) as ㄙˋ (szu4) at typing or pronouncing. Thus robust fault tolerance ability of a STC system has to be concerned so as to improve the phoneme-to-character conversion accuracy.

In recent years, various approaches have been proposed to construct a Chinese STC system either for speech input or keyboard input. For speech input, Chang [1994] used vector quantization to cluster words into classes when training Hidden Markov model so that words in the same class share the model's parameters. Contrast to the character N-gram based Markov model, a word N-gram based Markov model was proposed by Yang [1998]. Though Markov-based models are easy for implementation, they require large training corpus and large storage for large numbers of parameters. Furthermore, the parameters of Markov model are needed to be fixed, so they reflect the characters of training corpus only. Rather than using Markov model, Lin [1995] used mutual information to find the relation between base syllables and applied Heuristic Divide-and-Conquer Maximum Match (H-DCMM) Algorithm to detect prosodic-segment in a sentence. To train the robustness of prosodic-segment detection, a segmental K-means algorithm is also used.

As for syllable-based keyboard input, Gie [1990] used a hand-crafted dictionary for matching syllables of phrases and a set of impression rules for homonym selection. In Gie [1991], homonyms for new phrases are furtherly dealt by using a dictionary and occurrence frequencies. On the other hand, Lai [2000] used maximum likelihood ratio and good-tuning estimation to handle characters with multiple syllables. Lin [2002] combined N-gram model and selection rules for dealing with multiple PingIn codes. Unlike statistical approaches, context sensitive method proposed by Hsu [1995] was applied in a Chinese STC system called "Going." The system relies heavily on semantic pattern matching which can reduce the huge amount of data processing required for homophonic character selection. The conversion accuracy rate is close to 96%. In [Tsai and Hsu 2002], a semantically-oriented approach was also presented by using both noun-verb event-frame word-pairs and statistical calculation. The experimental results show that their overall syllable-to-word accuracy can be 96.5%.

In this paper a corpus-based STC system to support high tolerance is presented and it can be used as a keyboard input method as well as a post-processor incorporated with an acoustic system. To support high tolerance ability, we used a bucket-based searching mechanism so that the searching time of confusing syllable is reduced. The presented homonym resolution is based on binding force information and selection rules. Various tests are implemented to justify the system performance. In zero-tolerance test, our character conversion accuracy is 93.54% out of 1052 characters. For zero-tone testing, the character conversion accuracy is 80.13%. In input syllables with 20% and 40% confusing set member replacement, the character conversion accuracy is 83.08% and 78.23% respectively. The feasibility and robustness of fault tolerance handling to a STC system are also proved by the experiments.

The outline of the paper is as follows. Section 2 introduces the preliminary

background of Chinese syllable structure. Section 3 describes the system architecture and section 4 presents the proposed searching mechanism. Section 5 explains our selection module and section 6 reports various experimental tests. Finally Section 7 gives the conclusion.

2 Mandarin syllable and Confusing set

2.1 Sets of syllables in Mandarin

According to [Wu 1998], a general Mandarin syllable structure contains four parts consonant, head of diphthong, vowel and tone. There are twenty-one consonants, sixteen vowels, and five tones. Since users usually pronounce head of diphthong and vowel simultaneously, so the syllable structure can be simplified to combine head of diphthong and vowel such as ㄧ and ㄥ [Chen 1998].

Table 1 is the list of consonants, vowels, tones and the code number in our system. In this paper, we treat the syllable with tone=0 as tone=1. Because the amount of the syllables with tone=0 is quite few (19 out of 1302 Mandarin syllables), and their corresponding characters are few too (29 out of 14105 unique Mandarin characters).

Table 1: Consonants and vowels.

(a) Consonants

01	Nil	02	ㄅ	03	ㄆ	04	ㄇ	05	ㄈ
06	ㄉ	07	ㄊ	08	ㄋ	09	ㄌ	10	ㄍ
11	ㄎ	12	ㄏ	13	ㄐ	14	ㄑ	15	ㄒ
16	ㄓ	17	ㄔ	18	ㄕ	19	ㄖ	20	ㄗ
21	ㄘ	22	ㄙ						

(b) Vowels

01	Nil	02	ㄚ	03	ㄛ	04	ㄜ	05	ㄝ
06	ㄞ	07	ㄟ	08	ㄠ	09	ㄡ	10	ㄢ
11	ㄣ	12	ㄤ	13	ㄥ	14	ㄧ	15	ㄨ
16	ㄩ	17	ㄧㄚ	18	ㄧㄝ	19	ㄧㄞ	20	ㄧㄠ
21	ㄧㄡ	22	ㄧㄢ	23	ㄧㄣ	24	ㄧㄤ	25	ㄧㄥ
26	ㄨㄚ	27	ㄨㄛ	28	ㄨㄞ	29	ㄨㄟ	30	ㄨㄢ
31	ㄨㄣ	32	ㄨㄤ	33	ㄨㄥ	34	ㄩㄝ	35	ㄩㄢ
36	ㄩㄣ	37	ㄩㄥ	38	ㄦ	39	ㄧㄛ		

(c) Tones

˙	Nil	ˊ	ˇ	ˋ
1	1	2	3	4

2.2 Confusing set

The confusing sets are the groups of syllables, which are recognized to be the same by the human or the acoustic recognizer. For example, ㄈㄟ(fei1) and ㄏㄨㄟ (hui1) are confusing syllables for many Chinese-speaking people in Taiwan.

Suppose Table 2 is the statistical results from an acoustic recognizer. Then the confusing sets of phonemes can be found by using the find-connected-components algorithm [Thomas 1998] in which phonemes are vertices of a graph and the confusing sets are those edges whose recognition probabilities are greater than a threshold. For example, two confusing sets of phonemes, {ㄡ} and {ㄥ, ㄣ} are generated from Table 2 when their probabilities are greater than a given threshold at 25%.

Table 2: An example of acoustic data.

Phoneme	Result	Prob.	Result	Prob.	Result	Prob.
ㄡ	ㄡ	0.75	ㄥ	0.2	ㄣ	0.05
ㄥ	ㄥ	0.6	ㄣ	0.4		
ㄣ	ㄣ	0.7	ㄥ	0.3		

2.3 Bucket of confusing set

The confusing sets of syllable are obtained by using Cartesian product on two confusing sets of consonants and vowels, (an example shown in Table 3). Then a bucket $B(\alpha\ \beta)$ will contain the grams from $C(\alpha)$ of consonant confusing set and $V(\beta)$ of vowel confusing set. Fig. 1 is an example of bucket of bigram syllable confusing set, and its corresponding bucket is *B(08140607).*

Table 3: An Example of confusing sets.

(a) Confusing sets of consonant

C(01)	Nil, ㄇ, ㄈ, ㄏ	C(04)	ㄋ, ㄌ, ㄖ	C(07)	ㄓ, ㄗ
C(02)	ㄅ, ㄆ	C(05)	ㄍ, ㄎ	C(08)	ㄔ, ㄘ
C(03)	ㄉ, ㄊ	C(06)	ㄐ, ㄑ, ㄒ	C(09)	ㄕ, ㄙ

(b) Confusing sets of vowel

V(01)	Nil	V(06)	ㄚ、ㄢ、ㄤ	V(11)	ㄣ、ㄥ、ㄨㄣ
V(02)	ㄜ, ㄦ	V(07)	ㄧㄚ、ㄧㄤ	V(12)	ㄧㄛ、ㄧㄡ、ㄩㄥ
V(03)	ㄞ、ㄧㄞ	V(08)	ㄨㄞ、ㄨㄢ	V(13)	ㄧ、ㄧㄣ、ㄧㄥ、ㄩㄣ
V(04)	ㄠ、ㄧㄠ	V(09)	ㄨㄚ、ㄨㄤ	V(14)	ㄛ、ㄡ、ㄨㄛ、ㄨㄥ
V(05)	ㄨ、ㄩ	V(10)	ㄨㄟ、ㄩㄝ	V(15)	ㄝ、ㄟ、ㄧㄝ、ㄧㄢ、ㄩㄢ

C(08)	V(14)	×	C(06)	V(07)	=	
ㄔ, ㄘ	ㄛ、ㄡ、ㄨㄛ、ㄨㄥ		ㄐ, ㄑ, ㄒ	ㄧㄚ、ㄧㄤ		抽獎, 衝向, 抽象, …

Fig. 1: a bucket example of *B(08140607)*.

3 System Architecture

Fig. 2 shows the system flowchart containing foreground process and background process. In the background process, we used news documents collected from the Chinatimes website (http://news.chinatimes.com/) in March 2001 and segmented these documents into grams. Next, the Mandarin syllables for each gram were generated by our syllable generation method. We also encoded the grams by its confusing set which is obtained from acoustic statistic data. Then grams with confusing set information are stored into gram database.

The foreground process consists of fault tolerance matching module and selection module. Fault tolerance matching module encodes the phonetic symbol sequence and searches the corresponding grams that have minimum error distance with phonetic symbol sequence in the corpus database. Then corresponding unigrams, bigrams, and trigrams will be searched and passed into selection module. Selection module is constructed on the basis of selection rules to decide the output gram. The binding force information is calculation is done with CKIP word database contains 78,410 Mandarin words and their corresponding syllables. Finally, the output gram

will replace the characters of character sequence from the tail.

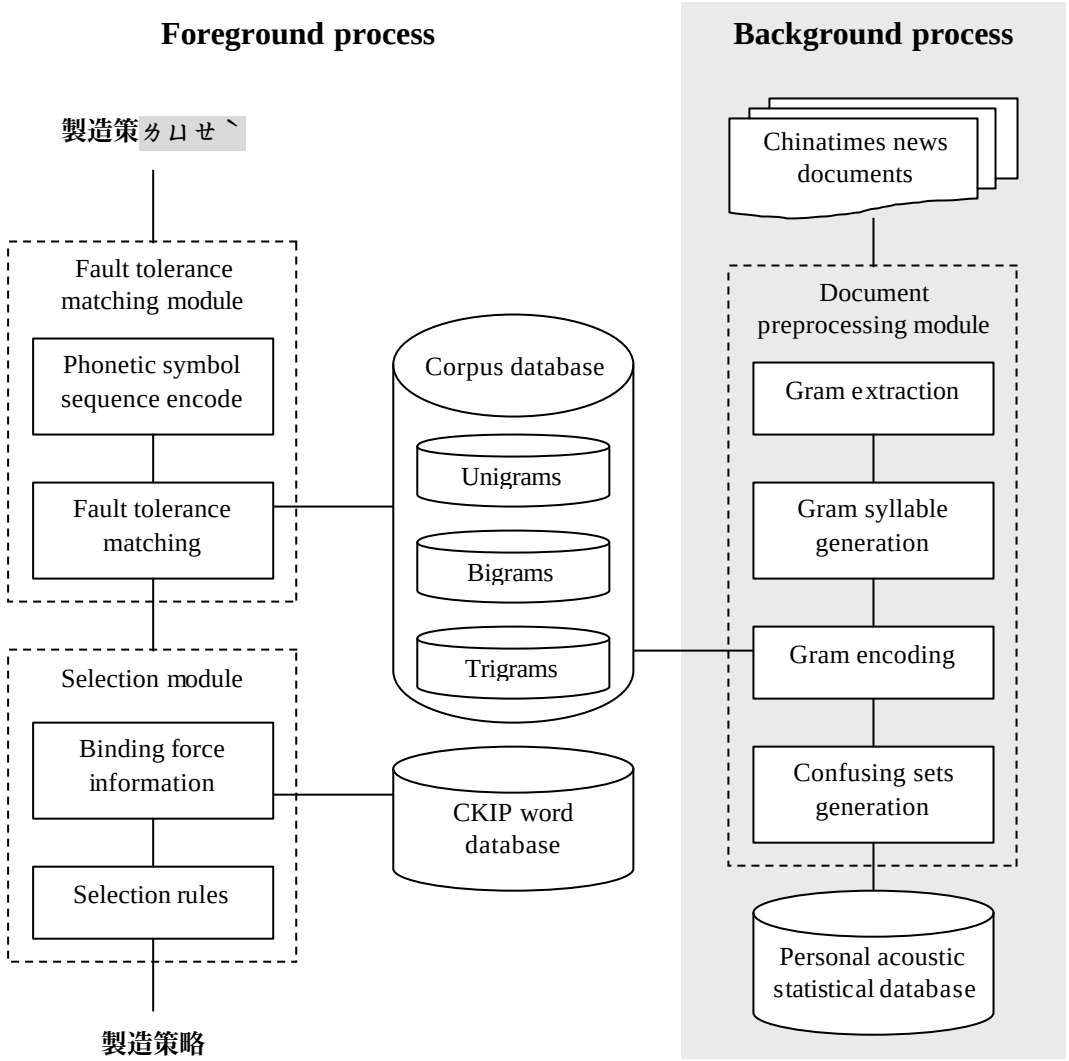

Fig. 2: The system architecture.

4 Fault Tolerance Matching Module

4.1 Base syllable distance

Let $N_{C,}$ N_V denote a confusing set number for consonant/vowel confusing set respectively. We define base syllable to be a syllable without tone. Then a bucket $B(N_C N_V)$ will contain those grams having the syllable confusing set $N_C N_V$.

A base syllable distance is the number of different consonant or vowel confusing set pairs between two base syllables. Suppose a base syllable sequence

SylSeq1=$c_1v_1c_2v_2c_3v_3$ which belongs to $B(N_{C1}N_{V1}N_{C2}N_{V2}N_{C3}N_{V3})$, and another syllable sequence *SylSeq2*=$c_1'v_1'c_2'v_2'c_3'v_3'$ which belongs to $B(N_{C1}'N_{V1}'N_{C2}'N_{V2}'N_{C3}'N_{V3}')$. *SylSeq2* has two base syllable distance from *SylSeq1* if there exists any two mismatch pairs of consonant or vowel confusing sets, like $N_{C1} \neq N_{C1}'$ and $N_{V2} \neq N_{V2}'$. Similarly, there will be K-distance if there are *K* mismatch pairs between *SylSeq1* and *SylSeq2*.

4.2 Bucket index

To find the grams with minimum base syllable distance from a given gram, we start to find the bucket first which the grams belong to. Our searching is done with the string matching algorithm proposed by Du and Chang [1994]. We start from the buckets with zero syllable distance. If there is no such gram in these buckets, we increase base syllable distance by 1. The maximum distance is defined to be 2 in this paper. We use index structure to memorize these buckets for every base syllable distance.

Let $[\gamma, \delta]_{(\phi,\omega)}$ denote a extension bucket index. Symbol γ and δ are the buckets whose errors are at any position except γ and δ; symbol ϕ is the base syllable distance and $\phi \in \{1, 2\}$; symbol ω represents bigram (ω=2) or trigram bucket index (ω= 3). For example, extension bucket index $[1,2]_{(1,3)}$ is a trigram index with one base syllable distance, and contains the buckets whose errors are at any position except the first and second ones. Therefore, $[1,2]_{(1,3)}$ contains the following buckets: $B(O_1O_2XO_4O_5O_6)$, $B(O_1O_2O_3XO_5O_6)$, $B(O_1O_2O_3O_4XO_6)$, *and* $B(O_1O_2O_3O_4O_5X)$ (we use X to indicate error occurrence and O correct one for notation simplification); similarly, extension bucket index $[5,6]_{(1,3)}$ contains buckets: $B(XO_2O_3O_4O_5O_6)$, $B(O_1X\ O_3O_4O_5O_6)$, $B(O_1O_2XO_4O_5O_6)$, *and* $B(O_1O_2O_3XO_5O_6)$.

In fact extension bucket index $[1,2]_{(1,3)}$ and $[5,6]_{(1,3)}$ together will include all buckets with one base syllable distance. The combination of extension bucket index

set which contains all the buckets is called a covering extension bucket index. Similarly, extension bucket index $[1,2]_{(1,3)}$ and $[5,6]_{(1,3)}$ are the members of trigram covering extension bucket index with one base syllable distance. Thus, there exists more than one solution in finding covering extension bucket index. In fact, finding the covering extension bucket index is a NP-complete problem [Garey and Johnson 1979]. Since the length of syllable sequence is short and the number of errors is small, it is easy to find the covering extension bucket index. Thus, searching buckets can be done in real time.

5 Selection Module

The designed selection module is based on sliding window whose size is set to be five in the proposed system. Let $C(S_{i-4})$, $C(S_{i-3})$, $C(S_{i-2})$, and $C(S_{i-1})$ be the characters in front of $C(S_i)$ at inputting syllable S_i. Then the ranking scheme shown as equation (1) is used to rank monograms $C(S_i)$, bigrams $C(S_{i-1})C(S_i)$ and trigrams $C(S_{i-2})C(S_{i-1})C(S_i)$ which exist in the gram database and each type of the grams with the top values will be treated as our candidate outputs and will be placed at corresponding positions.

$$Rank(g) = \begin{cases} P(g) \text{ if } g \text{ is monogram or trigram} \\ P(g) \times BF(g) \text{ if } g \text{ is bigram} \end{cases} \quad (1)$$

In (Eq. 1) $p(g)$ is the occurrence probability of g in the training corpus and the $BF(g)$ is the binding force for two characters C_i, C_{i+1} composing bigram g [Sproat 1990] and it is calculated as following equation:

$$BF(C_iC_{i+1}) = \log_2 \frac{P(C_iC_{i+1})}{P(C_i)P(C_{i+1})} \quad (2)$$

Then selection rules applied to select the candidate grams are as follows:

1. For a trigram candidate $C(S_{i-2})C(S_{i-1})C(S_i)$

   1.1. If either $C(S_{i-4})C(S_{i-3})C(S_{i-2})$ or $C(S_{i-3})C(S_{i-2})C(S_{i-1})$ exists in gram database, then if it has overlapping $C(S_{i-1})$ *or* $C(S_{i-2})C(S_{i-1})$ with $C(S_{i-2})C(S_{i-1})C(S_i)$, then output $C(S_{i-2})C(S_{i-1})C(S_i)$, otherwise abort $C(S_{i-2})C(S_{i-1})C(S_i)$

   1.2. If neither $C(S_{i-4})C(S_{i-3})C(S_{i-2})$ nor $C(S_{i-3})C(S_{i-2})C(S_{i-1})$ is in trigram database, then

      1.2.1 if both $BF(C(S_{i-3})C(S_{i-2}))$ and $BF(C(S_{i-2})C(S_{i-1}))$ is less than a threshold, then output $C(S_{i-2})C(S_{i-1})C(S_i)$

      1.2.2 if either $BF(C(S_{i-3})C(S_{i-2}))$ or $BF(C(S_{i-2})C(S_{i-1}))$ is greater than a threshold, and there exists overlapping $C(S_{i-1})$ *or* $C(S_{i-2})C(S_{i-1})$ with $C(S_{i-2})C(S_{i-1})C(S_i)$, then output $C(S_{i-2})C(S_{i-1})C(S_i)$.

      1.2.3 if either $BF(C(S_{i-3})C(S_{i-2}))$ or $BF(C(S_{i-2})C(S_{i-1}))$ is greater than a threshold but without any overlapping $C(S_{i-1})$ *or* $C(S_{i-2})C(S_{i-1})$ with $C(S_{i-2})C(S_{i-1})C(S_i)$, then abort $C(S_{i-2})C(S_{i-1})C(S_i)$.

2. If there is no $C(S_{i-2})C(S_{i-1})C(S_i)$ in database or $C(S_{i-2})C(S_{i-1})C(S_i)$ is aborted, then

   2.1. if $C(S_{i-3})C(S_{i-2})C(S_{i-1})$ exists in database, then check：

   if $C(S_{i-3})C(S_{i-2})C(S_{i-1})$ has overlapping $C(S_{i-1})$ with candidate $C(S_{i-1})C(S_i)$, then output $C(S_{i-1})C(S_i)$, otherwise output the candidate $C(S_i)$

   2.2. if $C(S_{i-3})C(S_{i-2})C(S_{i-1})$ is not in database but $C(S_{i-2})C(S_{i-1})$ is, then check:

   if $C(S_{i-2})C(S_{i-1})$ has overlapping $C(S_{i-1})$ with the candidate $C(S_{i-1})C(S_i)$

   then output $C(S_{i-1})C(S_i)$,

   else if $BF(C(S_{i-2})C(S_{i-1}))<$threshold then output candidate $C(S_{i-1})C(S_i)$;

   else if threshold$<BF(C(S_{i-2})C(S_{i-1}))<BF(C(S_{i-1})C(S_i))$, then output candidate $C(S_{i-1})C(S_i)$;

   else if $BF(C(S_{i-1})C(S_i))<BF(C(S_{i-2})C(S_{i-1}))$, then output candidate $C(S_i)$.

6 Experimental results

The experiments were implemented to justify the system feasibility and tolerance ability. Our training data includes CKIP word database which contains 78,410 words from length 1 to length 9 and Chinatimes News on the website (http://news.chinatimes.com/) containing 6,582 articles in March 2001. The testing data are collected from Chinetimes News on the website containing 7,828 articles in April 2001. The system development and testing environment is Windows 98 on PII 450mHz PC with 128MG Ram.

One experiment is to measure the response time of searching a word in a database. A database without bucket indexing 'no-bucket' is compared with 'bucket$_{9x15}$' which consists of nigh consonant and fifteen vowel confusing sets as listed in Table 3 of Section 2. The searching time of the databases with bucket indexing mechanism is less than one second. Table 4 shows the best case of searching time and there B(50K) means 50K bigrams, T(11K) means 11K trigram and so on.

Table 4: Best case of searching time (seconds)

	B(50K)+T(11K)	B(100K)+T(210K)	B(200K)+T(410K)	B(400K)+T(1350K)
No bucket	0.2	0.77	1.69	15.2
Bucket$_{9x15}$	0.02	0.03	0.03	0.04

Experiments are also implemented for various tolerance tests. There are 100 sentences randomly selected from the testing data and each sentence has 10.5 characters in average. We use two commercial STC systems for comparison, namely Microsoft IME 2002a (微軟新注音輸入法 XP), and Going 6.5 (自然注音輸入法). We compare the accuracy in various tolerance rates which is defined as Eq. 3. In this experiment, we disabled the system-defined confusing phonemes of MS 2002a, because its confusing mechanism and sets are quite different from ours. Table 5 shows the testing results with respect to different the accuracy among four systems.

$$Tolerance\ Rate = \frac{\sum \frac{Number\ of\ character's\ syllable\ replaced\ by\ confusing\ sets\ in\ a\ Sentence}{Number\ of\ characters\ in\ a\ Sentence}}{Total\ Numner\ of\ Sentence} \quad (3)$$

Table 5: the character accuracy of 100 testing sentences.

Tolerance rate	0%	20%	30%	40%
9x15	83.94%	83.08%	81.46%	78.23%
Going6.5	94.30%	67.97%	57.80%	45.34%
MS 2002a	94.87%	69.30%	56.18%	43.44%

On the other hand experiments to investigate the correlation between tolerance rate and positions were also implemented. The tolerance position is selected by testing users randomly. Both Table 6 and Table 7 show that the proposed STC system indeed supports robust fault tolerance ability.

Table 6: Character accuracy rate of $bucket_{9x15}$ using 30 training sentences.

	Tolerance at Consonant	Tolerance at Vowel	Tolerance at Any Position
Tolerance rate = 20%	91.77	92.89	94.33
Tolerance rate = 35%	89.3	85.76	86.6
Tolerance rate = 45%	86.42	86.27	86.22

Table 7: Character accuracy rate of $bucket_{9x15}$ using 30 testing sentences.

	Tolerance at Consonant	Tolerance at Vowel	Tolerance at Any Position
Tolerance rate = 20%	87.34	89.73	85.93
Tolerance rate = 30%	86.4	85.78	85.35
Tolerance rate = 40%	85.57	85.99	83.38

7 Conclusions

In this paper a high tolerant STC system useful for traditional Chinese input was presented. The proposed fault tolerance mechanism is constructed on the basis of a user-defined confusing set and a modified bucket indexing scheme is incorporated so as to satisfy real-time requirement. Meanwhile the homonym resolution is handled by

binding force and heuristic selection rules. The performance of the presented system is also justified and compared with various tests. However the drawbacks with the proposed system are its lack of semantic and syntactic checking at output selection. Hence errors like "**珊瑚下單(蛋)**", "**工作室(是)一種享受**" will occur. So how to strengthen the selection module with more linguistic reasoning will be our next step to design an intelligent STC system.

REFERENCE

Chang T. Z. 1994. A Word-Class-Based Chinese Language Model and its Adaptation for Mandrain Speech Recognition, *Master Thesis, National Taiwan University.*

Chen J. T. 1998. Neural Network-based Continuous Mandarin Speech Recognition System, *Master Thesis, National Chiao Tung University.*

Chinese Knowledge Information Processing Group (CKIP) Corpus 3.0. http://godel.iis.sinica.edu.tw/CKIP/

Du M. W. and Chang S. C. 1994. An Approach to Designing Very Fast Approximate String Matching Algorithms, *IEEE Transactions on Knowledge and Data Engineering*, 6, 4, 620-633.

Garey M. R. and Johnson D. S. 1979. Computers and Intractability. A Guide to the Theory of NP-Completeness, *Freeman, San Francisco.*

Gie C. X. 1990. A Phonetic Chinese Input System Based on Impression Principle, *Master Thesis, National Taiwan University.*

Gie T. H. 1991. A Phonetic Input System for Chinese Characters Using A Word Dictionary and Statistics, *Master Thesis, National Taiwan University.*

Hsu W. L. 1995. Chinese Parsing in a Phoneme-to-Character Conversion System based on Semantic Pattern Matching, *International Journal on Computer Processing of Chinese and Oriental Languages*, 40, 227-236.

Lai S. C. 2000. The Preliminary Study of phonetic symbol-to-Chinese character Conversion, *Master Thesis, National Tsing Hua University.*

Lee L. S., Tseng C.Y., Gu H. Y., Liu F. H., Chang C. H., Lin Y. H., Lee Y., Tu S. L., Hsieh S. H. and Chen C. H. 1993. Golden Mandarin (I) - A Real Time Mandarin Dictation Machine for Chinese Language with Vary Large Vocabulary, *IEEE Transactions on Speech and Audio Proceeding*, 1, 2.

Lin S. W. 1995. Prosodic-Segment Based Chinese Language Processing for Continous Mandarin Speech Recognition with very large Vocabulary, *Master Thesis, National Taiwan University.*

Lin J. X. 2002. A Mandarin Input System Compatible With Multiple Pinyin Methods, *Master Thesis, National Chung Hsing University.*

Tsai J. L. and Hsu W. L. 2002. Applying an NVEF Word-Pair Identifier to the Chinese Syllable-To-Word Conversion Problem, *The 19th International Conference on Computational Linguistics*.

Sproat R. and Shih C. 1990. A Statistic Method for Finding Word Boundaries in Chinese Text, *Computer Process of Chinese and Oriental Languages*, 4, 336-349.

Thomas H. C., Charles E. L. and Ronald L. R. 1998. Introduction To Algorithms, *McGraw-Hill Book Company, New York St. Louis San Francisco Montreal Toronto*, 440-443.

X. X. Wu. 1998. A Bucket Indexing Scheme for Error Tolerant Chinese Phrase Matching. *Master Thesis, National Chiao-Tung University.*

Yang K. C. 1998. Further Studies for Practical Chinese Language Modeling, *Master Thesis, National Taiwan University.*