Error Recovery in Natural Language Parsing with a Level-Synchronous Approach

Yi-Chung Lin* and Keh-Yih Su†

* Advanced Technology Center, Computer & Communication Research Lab.

Industrial Technology Research Institute, Chutung, Taiwan 310, R.O.C.

lyc@atc.ccl.itri.org.tw

† Behavior Design Corporation, 2F, No. 5, Industrial East Road IV

Science-Based Industrial Park, Hsinchu, Taiwan 300, R.O.C.

kysu@bdc.com.tw

Abstract

A level-synchronous probabilistic scoring function, which takes advantage of the wide-scope contextual information in the same phrase-level, is proposed in this paper to detect and recover the errors in ill-formed inputs. Traditionally, partial parsing is used in dealing with ill-formed inputs without fixing the errors. However, such an approach only provides coarse and limited information, which prevents the sentence from being processed further (such as translation). To fix the errors in ill-formed inputs, a recovery mechanism using the probabilistic scoring function is proposed in this paper. Experimental results show that 35% of the ill-formed inputs can be recovered to well-formed parses. The recall of constituent brackets is also significantly improved from 68.49% to 76.60%, while the precision of brackets is slightly improved from 79.49% to 80.69%.

1. Introduction

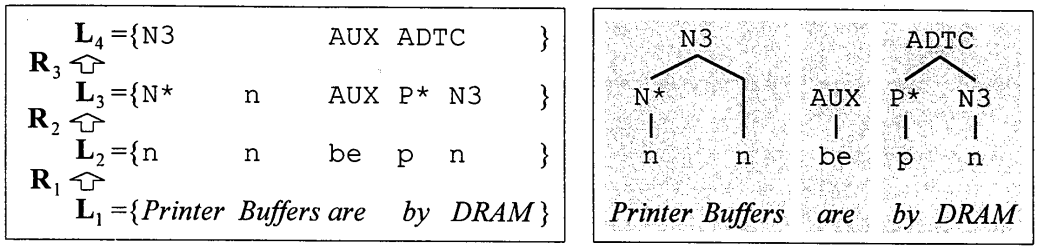
In real world applications, ill-formed inputs are inevitable in a natural language processing system. It is infeasible to limit users to speak or write with only a limited vocabulary or a predefined grammar. In many systems, ill-formed inputs are handled by a partial parsing mechanism. In these systems, an ill-formed input is first partitioned (or parsed) into recognizable pieces of phrases (i.e., partial parses). Then, by consulting the forest (Tomita, 1987) of the partial parses, the system takes some application-specific actions to deal with the ill-formed input. However, many errors in the inputs are not isolated from their neighbors; they may disguise their neighbors not to be recognized by the system. As a result, only coarse

and limited information is provided by partially parsing an ill-formed input if its errors are not fixed.

To recover the errors, some systems had tried to fix the errors during or after parsing. In 1981, Kwasny and Sondheimer (1981) proposed to parse ill-formed inputs with an Augmented Transition Network (ATN) parser. In their approach, the types of errors are carefully identified and the corresponding transition arcs, called relaxed arcs, are manually created in the network to recover those errors. These relaxed arcs are blocked in normal cases. Once all the grammatical paths fail, these relaxed arcs are attempted. Weischedel and Sondheimer (1983) also used a similar approach. They used meta-rules to associate certain ill-formed inputs with particular well-formed structures by modifying the violated grammar rules. In 1989, Mellish proposed to find the full parse by running a modified top-down parser over the partial parses generated by a bottom-up chart parser. The modified top-down parser attempts to find a full parse tree by considering one word error. On the other hand, in 1990, Abney (1990; 1991) proposed to parse natural language by segmenting the parts-of-speech into chunks and then assembling the chunks into a complete parse tree. In his work, the chunks were repaired and assembled by predefined heuristic rules. Recently, Lee et al. (1995) generalized the least-error recognition algorithm (Lyon, 1974) to find the full parses of minimum error with a small grammar of only 192 grammar rules. Since exhaustively finding the full parses with minimum error is very time-consuming, they used heuristic rules and heuristic scores to cut down the search space.

All the above mentioned approaches involve some ad hoc heuristic rules to fix the errors or restrict the search space. Those heuristic rules are usually system-specific and hard to be reused by other systems. Besides, although those approaches may work well in small tasks on specific domains, they lack extensibility and are hard to be scaled-up. Therefore, a generalized approach, independent of any particular system and domain, is highly demanded.

In this paper, we propose an error recovery mechanism using a generalized probabilistic scoring function to identify and recover the errors. Since the errors could occur at any places in the inputs, exhaustively searching all possibilities is infeasible. Thus, a two-stage strategy is proposed to limit the search space. In the first stage, the most possible forest of partial parses is tried to fit into the S-productions, whose left-hand side symbols are the "S" symbol (i.e., the start symbol). If the forest cannot be well fitted by applying one or two modification actions, the part-of-speech errors are considered in the second stage. Experimental results



(a) (b)

Figure 1. (a) Parsing by building phrase-levels. (b) The corresponding partial parses.

show that 35% of the ill-formed inputs are recovered to their correct well-formed full parses. The recall of constituent brackets is increased from 68.49% to 76.60%, while the precision of brackets is slightly improved from 79.49% to 80.69%.

2. Robust Parsing by Building Phrase-levels

In natural language processing, contextual information is helpful for resolving ambiguity problems. One obvious example is the trigram part-of-speech tagger (Church, 1989). In order to incorporate wide-scope contextual information into the parsing process, we (Lin, 1995; Lin and Su, 1997) proposed to parse a sentence in a level-synchronous manner, in which parsing a sentence is treated as building a set of *phrase-levels*.

Figure 1 is an example of the building process. Initially, the first (i.e., the lowest) phrase-level, $\mathbf{L}_1$, consists of the input words. Then, according to the dictionary, $\mathbf{L}_1$ is built-up to the second phrase-level, $\mathbf{L}_2$, which consists of parts-of-speech. After then, phrase-levels are built-up according to the grammar rules. This process repeats until no more rules can be applied. If the input is well-formed, we can get the "S" symbol in the topmost phrase-level; otherwise, the topmost phrase-level will contain the roots of the partial parses derived from the input. As a result, a parse tree, either a full parse or a forest of partial parses, can be represented as $\mathbf{T} = \{\mathbf{L}_N, \mathbf{R}_{N-1}, \mathbf{L}_{N-1}, \cdots, \mathbf{R}_1, \mathbf{L}_1\}$, where N is the number of phrase-levels, $\mathbf{L}_i$ indicates the i-th phrase-level, and $\mathbf{R}_i$ denotes the set of actions (i.e., reduce actions) used to build $\mathbf{L}_{i+1}$ from $\mathbf{L}_i$.

We also proposed the following scoring function to evaluate the likelihood of a parse tree:

$$S_{\mathbf{T}} = \prod_{A_j \in \mathbf{L}_N} P(A_j \mid A_{j-2}, A_{j-1}) \times \prod_{i=1}^{N-1} \prod_{\rho_j = \langle r; t \rangle \in \mathbf{R}_i} P(r \mid A_{t-1}^{t+1}), \tag{1}$$

where A_j is the j-th symbol in $\mathbf{L}_N$, r is the production rule applied by the j-th action ρ_j in $\mathbf{R}_i$, A_{l-1}^{l+1} is the short-hand for "A_{l-1}, A_l, A_{l+1}" in $\mathbf{L}_{i+1}$. Intuitively, the first term $\prod_{A_j \in \mathbf{L}_N} P(A_j \mid A_{j-2}, A_{j-1})$ accounts for the prior probability of the topmost phrase-level; the second term $\prod_{i=1}^{N-1} \prod_{\rho_j = <r;l> \in \mathbf{R}_i} P(r \mid A_{l-1}^{l+1})$ ascribes to the likelihood of applying the grammar rules.

The proposed level-synchronous approach is tested on 200 ill-formed English sentences collected from an English-to-Chinese machine translation system. Compared with the approach of using the stochastic context-free grammar and the heuristics of preferring the longest phrase (Mellish, 1989; Seneff, 1992), the level-synchronous approach improves the precision and recall of brackets from 69.37% to 79.49% and from 78.73% to 81.39%, respectively (Lin, 1995; Lin and Su, 1997).

3. Error Recovery with Modification Actions

The above level-synchronous approach does not fix any errors. However, many errors of ill-formed inputs are not isolated from their neighbors. They may affect their neighbors and prevent them from being recognized by the parser. In order to reduce the effects of errors, a recovery mechanism should be incorporated. For this purpose, the building process is further generalized to fix the errors of ill-formed inputs as follows.

To fix the errors in a phrase-level, the modification actions of insertion, deletion and substitution are incorporated into the parsing process. These actions are realized by three modification productions: $X \rightarrow \varepsilon$ (inserting a symbol X), $\varepsilon \rightarrow X$ (deleting a symbol X) and $Y \rightarrow X$ (replacing the symbol X with the symbol Y). A modification action consists of a rule argument and two position arguments, like $\tilde{\rho} = \langle \tilde{r}; u, v \rangle$, where $\tilde{r}$ stands for the applied modification production rule, u and v indicate that this modification action is applied between the u-th and the v-th symbols in the modified phrase-level.

For example, the input "*Printer Buffers are by DRAM*" missed a verb after the auxiliary "*are*". To correct such an error, a verb should be inserted so that $\mathbf{L}_2$={n n be p n} could be modified to $\tilde{\mathbf{L}}_2$={n n be **v** p n}. In this case, the corresponding modification action set $\tilde{\mathbf{R}}_2$ will consist of one modification action $\tilde{\rho} = \langle \tilde{r} = \mathrm{v} \rightarrow \varepsilon; u = 3, v = 5 \rangle$. In that

modification action, the rule argument $\tilde{r} = \mathrm{v} \rightarrow \varepsilon$ indicates to insert a verb; the position arguments u=3 and v=5 indicate the inserted verb is placed between the third and the fifth symbols of the modified phrase-level $\tilde{\mathbf{L}}_2$.

Incorporated with modification actions, the parsing process can be considered as applying modification actions and normal actions in turn. Therefore, a modified parse tree $\tilde{\mathbf{T}}$ of N phrase-levels can be represented as $\tilde{\mathbf{T}} = \{\mathbf{L}_N, \mathbf{R}_{N-1}, \tilde{\mathbf{L}}_{N-1}, \tilde{\mathbf{R}}_{N-1}, \mathbf{L}_{N-1}, \cdots, \mathbf{L}_2, \mathbf{R}_1, \tilde{\mathbf{L}}_1, \tilde{\mathbf{R}}_1, \mathbf{L}_1\}$, where $\mathbf{L}_i$ is the i-th phrase-level; $\tilde{\mathbf{R}}_i$ is the set of modification actions to modify $\mathbf{L}_i$; $\tilde{\mathbf{L}}_i$ is the result of applying $\tilde{\mathbf{R}}_i$ to modify $\mathbf{L}_i$; $\mathbf{R}_i$ is the set of normal actions applied to build $\mathbf{L}_{i+1}$ from $\tilde{\mathbf{L}}_i$. After some derivations like those deriving Equation (1) (Lin and Su, 1997), the score of a modified parse tree is defined as

$$S_{\tilde{\mathbf{T}}} = \prod_{A_j \in \mathbf{L}_N} P(A_j \mid A_{j-2}, A_{j-1}) \times \prod_{i=1}^{N-1} \prod_{\rho_j = <r;l> \in \mathbf{R}_i} P(r \mid A_{l-1}^{l+1}) \times \prod_{\tilde{\mathbf{R}}_i = \phi} P(\tilde{\mathbf{R}}_i) \times \prod_{\tilde{\mathbf{R}}_i \neq \phi} \prod_{\tilde{\rho}_j = <\tilde{r};u,v> \in \tilde{\mathbf{R}}_i} P(\tilde{r} \mid \tilde{A}_u^v), \tag{2}$$

where the notation $\tilde{\mathbf{R}}_i = \phi$ indicates that $\tilde{\mathbf{R}}_i$ is an empty set (i.e., no modifications are applied to modify $\mathbf{L}_i$); $\tilde{\mathbf{R}}_i \neq \phi$ denotes that it is not an empty set, $\tilde{A}_u^v$ is the short-hand for symbols from the u-th to the v-th in $\tilde{\mathbf{L}}_i$. The first two product terms, which are related to the normal productions, are the same as those in Equation (1). The third product term $\prod_{\tilde{\mathbf{R}}_i = \phi} P(\tilde{\mathbf{R}}_i)$ is related to the phrase-levels which need not be modified. The last product term $\prod_{\tilde{\mathbf{R}}_i \neq \phi} \prod_{\tilde{\rho}_j = <\tilde{r};u,v> \in \tilde{\mathbf{R}}_i} P(\tilde{r} \mid \tilde{A}_u^v)$ accounts for the modification actions. Currently, the probabilities in Equation (2) are estimated from an annotated corpus by using Good-Turing estimation method (Good, 1953).

4. Two-stage Strategy to Find Potential Modification Actions

Theoretically, any modification actions can be applied to modify any phrase-levels of an ill-formed input. However, since the number of possible modifications is very large, it is infeasible to blindly try every one. Therefore, a two-stage strategy is proposed to find the

potential modifications. In the first stage, the forest of partial parses is first fitted into the S-productions, whose left-hand-side symbols are the "S" symbol (i.e., the start symbol). If the forest of partial parses cannot be well fitted into those S-productions, they are passed to the second stage to recover the part-of-speech errors. The details of these two stages are described in the following sections.

4.1 Fitting the partial parses

The errors of ill-formed inputs are two kinds: the isolated errors and the clingy errors. The isolated errors are those that do not hinder the parser from correctly parsing other phrases. Take the sentence "*Printer buffers are made by DRAM*" as an example. If the noun phrase "*Printer buffers*" is missed, the other words still can be parsed into a verb phrase. Therefore, such an error is an isolated error. On the contrary, missing the word "*made*" will hinder a parser from parsing the other three words "*are by DRAM*" into a verb phrase. Thus, it is a clingy error (clings to other words) in this example.

Isolated errors could be recovered by fitting the partial parses (Jensen, Miller, and Ravin, 1983). In the past, the fitting procedure is usually guided by heuristic rules, such as preferring some head phrases and preferring the widest phrase. Since acquiring those heuristic rules is expensive and maintaining the consistency of a large number of rules is difficult, the heuristic approach is hard to be scaled up. Besides, the heuristic rules are usually system-specific and hard to be reused by other systems. Therefore, we attempt to recover the isolated errors by fitting the partial parses according to probabilistic scores.

The forest of partial parses is fitted into the S-productions because most of the partial parses are constituents of S-productions. At most two modification actions are allowed to fit the forest of partial parses to the right-hand sides of the S-productions. Different modification actions can fit the forest into different S-productions and construct different full parse trees. These full trees are then ranked by the scoring function $S_{\tilde{T}}$ in Equation (2). If the forest of partial parses cannot be fitted into a full tree with one or two modification actions, it is assumed that the errors of this ill-formed input are not isolated. Then, the partial parses are passed to the second stage to fix the clingy errors.

4.2 Recovering errors of parts-of-speech

It is noticed that many clingy errors come from the second phrase-level (i.e., the phrase-

level of parts-of-speech). Therefore, in the second stage, attempts are made to recover the errors originated from the second phrase-level. Since enormous modifications can be applied to modify parts-of-speech, it is infeasible to try all of them. To be practical, currently, only the modifications with one insertion, deletion or substitution are permitted. Furthermore, since our training set of ill-formed inputs is rather limited, it cannot offer reliable statistical information to find the potential actions to modify the parts-of-speech. Therefore, the statistical information is acquired from well-formed training data via three scoring functions as described below.

Using the trigram formulation, the likelihood of a part-of-speech sequence, $c_1^n = c_1, c_2, \cdots, c_n$, can be approximated to be $\prod_{j=1}^{n} P(c_j \mid c_{j-2}, c_{j-1})$, where c_j denotes the j-th part-of-speech. Therefore, the score for inserting a part-of-speech "x" before the i-th part-of-speech is defined as:

$$S_{\text{INS}}(i, x; c_1^n) \equiv P(x \mid c_{i-2}, c_{i-1}) \times P(c_i \mid c_{i-1}, x) \times P(c_{i+1} \mid x, c_i) \times \prod_{\substack{j=1 \\ j \neq i, i+1}}^{n} P(c_j \mid c_{j-2}, c_{j-1})$$

With this scoring function, we can find the most probable modifications of one insertion action. Currently, only the top 5 insertion actions are applied to modify parts-of-speech. In the same way, the scores for deleting the i-th part-of-speech and substituting the i-th part-of-speech with "x" are respectively defined as:

$$S_{\text{DEL}}(i; c_1^n) \equiv P(c_{i+1} \mid c_{i-2}, c_{i-1}) \times P(c_{i+2} \mid c_{i-1}, c_{i+1}) \times \prod_{\substack{j=1 \\ j \neq i, i+1, i+2}}^{n} P(c_j \mid c_{j-2}, c_{j-1})$$

$$S_{\text{SUB}}(i, x; c_1^n) \equiv P(x \mid c_{i-2}, c_{i-1}) \times P(c_{i+1} \mid c_{i-1}, x) \times P(c_{i+2} \mid x, c_{i+1}) \times \prod_{\substack{j=1 \\ j \neq i, i+1, i+2}}^{n} P(c_j \mid c_{j-2}, c_{j-1})$$

Again, only the top 5 deletion actions and the top 5 substitution actions are applied to modify the parts-of-speech. These 15 modified phrase-levels of parts-of-speech are then parsed by building the modified phrase-levels and finally the full parse trees are ranked by the scoring function $S_{\tilde{T}}$ defined in Equation (2).

5. Experimental Results and Discussions

In our experiments, 8,727 well-formed sentences collected from computer manuals and their correct parse trees are used as the training data to estimate the parameters corresponding to the normal production actions. The average length of these sentences is about 13 words. A context-free grammar of 29 terminals, 140 nonterminals and 1,013 productions is used to parse input sentences. To estimate the parameters corresponding to the modification actions in Equation (2), another training set of 300 ill-formed sentences is collected and carefully parsed to full parse trees annotated with the required modification actions. Besides, the 200 ill-formed sentences in the testing set are also parsed to full parse trees with correct modification actions so that they can be used to test the performance of the proposed error recovery mechanism.

Table 1 lists the experimental results of parsing the 200 ill-formed testing sentences. The first row (PLB) corresponds to the performance of parsing without error recovery. The second row (ER1) gives the results of error recovery up to the first stage (i.e., fitting the partial parses). The last row (ER2) shows the results of error recovery up to the second stage (i.e., both fitting the partial parses and recovering errors of parts-of-speech).

The second row in Table 1 shows that, by fitting the partial parses into the S-productions, 25% of the ill-formed inputs can be correctly parsed and fitted to full parse trees. The last column indicates that 43% of the ill-formed inputs can be parsed to full parse trees by fitting their partial parses with one or two modification actions. In other words, 18% (resulted from subtracting 25% from 43%) of the ill-formed inputs are parsed to incorrect full parse trees.

The last row of Table 1 shows that, using the two-stage error recovery mechanism, 35% of the ill-formed inputs can be correctly parsed to the full parse trees. That is, 10% (resulted from subtracting 25% from 35%) of the ill-formed sentences are correctly parsed by recovering the errors of parts-of-speech. An ill-formed sentence correctly recovered in the

	Bracket and its label		Parse tree	
	Precision	Recall	Accuracy	Fitting rate
PLB	79.49%	68.49%	0.0%	0.0%
ER1	80.02%	70.59%	25.0%	43.0%
ER2	80.69%	76.60%	35.0%	76.0%

Table 1. Performances of parsing without and with error recovery

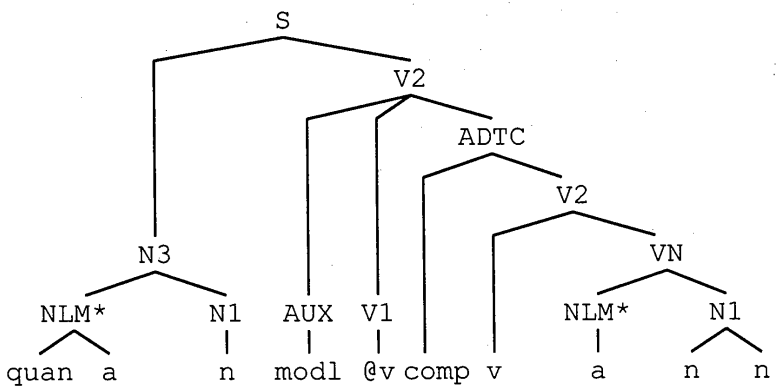


Figure 2. An ill-formed sentence correctly recovered in the second stage. "@V" indicates to replace the part-of-speech of the word "sufflce" with "v".

second stage is shown in Figure 2. In this example, the word "sufflce", which is a typo of the word "suffice", is regarded as an unknown word and, thus, is considered as a noun by default. After the second stage, 76% of the ill-formed sentences can be fitted into a well-formed syntactic structure.

By carefully inspecting the results, we find that, at the first stage, the improvement on the accuracy rate of parse trees is significant but the improvement on the bracket recall rate is little. The bracket recall rate is significantly improved in the second stage. This phenomenon is due to the fact that the errors recovered at the first stage are isolated errors. These errors do not hinder the parser from correctly parsing other words. On the contrary, the errors recovered at the second stage are not isolated. They seriously affect the partial parses of other words. Therefore, recovering such errors can significantly improve the bracket recall rate.

At both the first stage and the second stage, some ill-formed sentences are parsed to the incorrect full parse trees. Therefore, the numbers of incorrect brackets are increased at both stages. While computing the precision rate for brackets, these increased incorrect brackets compensate the increased correct brackets, which come from correctly parsing the ill-formed sentences. As a result, there are almost no improvement on the bracket precision rate at both the first stage and the second stage.

6. Error Analysis and Future Works

Although 35% of the ill-formed sentences can be recovered and correctly parsed to full parse trees, there are still many errors unresolved. By inspecting the remaining 65% unrecovered

sentences, three different types of errors are identified.

About 35% of the overall errors are caused by syntactic ambiguity. Most of these syntactic errors come from the problem of prepositional phrase attachment. To resolve such type of errors, purely syntactic information is not enough; higher level language knowledge, such as semantic knowledge, should be incorporated.

On the other hand, about 29% of the overall errors are due to incorrectly applying modification actions. Most of these errors occur at the second stage, where modification actions are applied to modify the parts-of-speech. About half of these errors are introduced because the correct modification actions are not included in those top 15 modification actions. To attack this problem, the discrimination power of the scoring function, which is used to select the potential modification actions, should be enhanced. Another half of the errors come from selecting the full parse tree derived from the undesired modification action. For example, the correct modification action for the ill-formed sentence "*An may appear in the display*" is to insert a noun after the word "*An*". However, the output tree is derived from the undesired modification action which replaces the part-of-speech of the word "*An*" with pronoun. This is because the incorrectly modified full tree is better than the correctly parsed full tree from the syntactic point of view. Therefore, such errors can be regarded as coming from "syntactic ambiguity" and the semantic knowledge should be incorporated to resolve them.

The last part of the unrecovered errors is caused by multiple errors in an ill-formed input. The portion of this part is 36%. For example, the sentence "*The tutorial explains how and why to use the tool*" is not covered by our grammar. Our grammar just covers the noun clause of only one question word (i.e., "*why*") followed by a infinitive (i.e., "*to use the tool*"). The desired modifications for this sentence is to delete the parts-of-speech of the words "how and". To recover such errors, multiple modification actions should be allowed to modify the parts-of-speech in the second stage. Since the number of different ways to modify a phrase-level of parts-of-speech with multiple modification actions is very large, a fast search method is necessary to find the likely combinations of modification actions. Besides, applying multiple modification actions will cause the numbers of parts-of-speech of the modified phrase-levels to be very different. However, a full parse tree with fewer parts-of-speech is usually more likely to have a higher score than a full tree with more parts-of-speech. Therefore, the normalization issue must be considered to fairly score the full parse trees with different numbers of parts-of-speech.

In summary, further improvements should be made in the following directions. First, semantic knowledge should be incorporated to resolve the errors coming from syntactic ambiguity and some of the errors caused by incorrectly applied modification actions. Second, the discrimination power and speed of the scoring function for searching potential modification actions should be enhanced. Third, the normalization issue should be considered to deal with the ill-formed sentences of multiple errors.

7. Conclusion

By incorporating wide-scope contextual information, the level-synchronous parsing mechanism (Lin and Su, 1997) showed its superiority in efficiency and accuracy for parsing ill-formed inputs. However, the proposed mechanism does not try to recover the errors. It only provides a forest of partial parses for an ill-formed input. In this paper, we generalize the level-synchronous parsing mechanism such that the errors can be fixed and a full parse can be provided. Besides, a two-stage strategy is also used to efficiently find the most probable modification actions to recover the errors in an ill-formed input. The experimental results show that the enhanced parser can correctly recover the errors in 35% ill-formed inputs. The recall of brackets is significantly improved from 68.49% to 76.60%, while the precision of brackets is slightly improved from 79.49% to 80.69%.

Acknowledgements

This paper is a partial result of both the Project 3P11200 conducted by ITRI under sponsorship of the Ministry of Economic Affairs, R.O.C. and the Project NSC-84-2221-E-007-013 sponsored by the National Science Council, R.O.C.

References

Abney, S. P., "Rapid Incremental Parsing with Repair," in *Proc. of the 6th New OED Conference: Electronic Text Research*, 1990, pp. 1–9.

Abney, S. P., "Parsing by Chunks," in *Principle-Based Parsing: Computation and Psycholinguistics*, Robert C. Berwick, Steven P. Abney, and Carol Tenny (Eds.), Kluwer Academic Publishers,1991, pp. 257–278.

Church, K. W., "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text," in *Proc. of ICASSP*, 1989, pp. 695–698.

Good, I. J., "The Population Frequencies of Species and the Estimation of Population Parameters," Biometrika, 40, 1953, pp. 237–264.

Jensen, K., G. E. Heidorn L. A. Miller, and Y. Ravin., "Parse Fitting and Prose Fixing: Getting a Hold on Ill-formedness," *American Journal of Computational Linguistics*, 9(3–4), 1983, pp. 147–160.

Kwasny, S. C. and N. K. Sondheimer, "Relaxation Techniques for Parsing Grammatically Ill-Formed Input in Natural Language Understanding Systems," *American Journal of Computational Linguistics*, 7(2), 1981, pp. 99–108.

Lee, K. J., C. J. Kweon, J. Seo, and G. C. Kim., "A Robust Parser Based on Syntactic Information," in *Proc. of the 7th Conference of European Chapter of the Association for Computational Linguistics*, 1995, pp. 223–228.

Lin, Y.-C., "A Level Synchronous Approach to Ill-Formed Sentence Parsing and Error Recovery," Ph.D. Thesis, Department of Electrical Engineering, National Tsing-Hua University, ROC, 1995.

Lin, Y.-C. and K.-Y. Su., "A Level Synchronous Approach to Ill-Formed Sentence Parsing," in *Proc. of the 10th R.O.C. Computational Linguistics International Conference*, 1997, pp. 89-108.

Lyon, G., "Syntax-Directed Least-Errors Analysis for Context-Free Languages," *Communications of the ACM*, 17(1), 1974, pp. 3–14.

Mellish, C. S., "Some Chart-Based Techniques for Parsing Ill-Formed Input," in *Proc. of the 27th Annual Meeting of the Association for Computational Linguistics*, 1989, pp. 102–109.

Seneff, S., "Robust Parsing for Spoken Language System," in *Proc. of ICASSP*, 1992, pp. 189–192.

Tomita, M., "An Efficient Augmented-Context-Free Parsing Algorithm," *Computational Linguistics*, 13(1-2), 1987, pp. 31-46.

Weischedel, R. M. and N. K. Sondheimer, "Meta-Rules as a Basis for Processing Ill-Formed Input," *American Journal of Computational Linguistics*, 9(3–4), 1983, pp. 161–177.

以語境判定中文未知詞詞類的方法

白明弘 陳超然 陳克健
中央研究院資訊科學研究所
e-mail: evan@iis.sinica.edu.tw, richard@iis.sinica.edu.tw, kchen@iis.sinica.edu.tw
Fax:(02)2788-1638

摘要

從中研院平衡語料庫估算，未知詞在實際的文章中約佔 3.51%，由於這些詞無法直接從辭典中獲得詞類訊息，所以必須以猜測的方式來獲得未知詞的詞類。在[Chen et al. 97]中曾經以詞首字及詞尾字和詞類的關係來猜測未知詞的詞類，其前三名猜測的覆蓋率可以達到約 96%，然而第一名召回的正確率只有 76%。本文將基於此一猜測方法，提出以語境規則來協助判定未知詞詞類的方法。在實際的測試中，第一名召回的正確率可以提升到 83.83%。

1. 緒論

在中文自然語言分析系統中，從辭典中查詢一個詞彙的詞類訊息是最基本的工作。然而，並不是所有的詞彙都可以在辭典中找到，這些辭典中找不到的詞稱爲未知詞。未知詞大部分是一些專有名詞或是複合詞等，因爲無法被窮舉，所以在一般的辭典中不會收錄。未知詞在實際文章中大約佔 3.51%，由於這些詞無法直接從辭典中獲得詞類訊息，而必須依賴詞彙本身的結構以及語境來猜測，所以詞彙的詞類猜測研究成爲不可避免的課題。在西方語言的研究當中，未知詞詞類的猜測方法十分依賴詞綴的構詞律[Mikheev 96]。但是在中文裡，詞彙之間沒有空白字元作區隔，而詞彙本身也沒有規則的詞尾變化，詞綴衍生詞的構詞律只能解決一小部分的問題而已。在[Chen et al. 97]中曾經提出以詞首字及詞尾字和詞類間的相互關係來猜測未知詞的詞類，這個方法猜測未知詞的效率，其前三名的覆蓋率可達約 96%，但是第一名召回的正確率卻降到 76%。爲了更準確地猜測未知詞的詞類，我們嘗試以這個猜測方法所猜測的前三名詞

類做爲候選詞類，而以語境規律從候選詞類中選擇最可能的未知詞詞類。

1.1 未知詞的型態

在中文未知詞的處理當中，不同構詞型態的未知詞，其處理的方式差異可能非常大。在目前所發表的論文當中，多半把不同型態的未知詞以不同的主題來研究。光是專有名詞一類就有專門探討中式人名識別[孫茂松等 94]、音譯人名識別[Lee 94]、以及組織名識別[陳信希等 94]等等的論文。我們從中央研究院平衡語料庫[Chen et al. 96]中，分析未知詞的構詞型態，歸納出未知詞最常發生的幾種類型，分列如下：

(a)略語：例如'中油(Nb)'，'台汽(Nb)'。略語的構詞律非常不規則，他們的詞首詞尾不大能夠反映出詞的意義[Huang 94]。

(b)專有名詞：例如'陳壽(Nb)'，'電機科(Nc)'，'香檳城(Nc)'，'微軟(Nb)'。專有名詞可以進一步分成三類，人名、地名以及組織名。在不同的類中，有不同的關鍵字可以識別。中國人名的姓氏的用字 90%集中在 114 個字裡[孫等 1994]，地區名常常以'市'、'鄉'等字爲結尾。而組織名則比較沒有規則，其構詞成分幾乎沒有任何限制。

(c)衍生詞：例如'電腦化(Vh)'。衍生詞有詞綴，是很好的識別指標。

(d)複合詞：例如'轉赴(VCL)'，'獲允(VE)'，'搜尋法(Na)'，'電腦桌(Na)'。複合詞的構詞比較複雜。

(e)數字型複合詞：例如'1986 年(Nd)'，'三千'，'19 巷(Nc)'。數字型複合詞的特點是他們都包含數字，例如日期、時間、電話號碼、地址、數字和定量式複合詞等等，都屬於這種型態。這類的複合詞比較規律，可以使用構詞律來識別。

然而不同類型之間的未知詞可能會有歧義或相同的構詞律，例如'陳年品'可能是一個普通名詞，也可能是人名。從構詞成分不容易區分，但是可能比較容易從未知詞出現的上下文中判別其正確的詞類。

1.2 語境和詞類的關係

在[Chen et al. 97]的研究中利用詞首和詞尾字與詞類間的關係，找出和詞首字與詞

尾字之間相關訊息量(mutual information)[Blahut 87, Su 96]及 Dice 測度(dice measure)[Su 96, Smajda 96]可以計算出相關性最強的詞類。前三名的猜測覆蓋率律可達 96%，然而第一名召回的正確率只有 76%。很顯然只靠詞首詞尾的訊息，不太容易正確的判定未知詞的詞類，但是可能很容易的去除大部分不可能的詞類。根據我們在語料庫中的觀察，未知詞的詞類和語境之間有某種程度的關係。例如'院長'和'所長'後面所接的未知詞 90%以上都是人名（Nb 類），'位於'後面所接的未知詞 85%以上都是地名（Nc 類），'民國'和'西元'後面所接的未知詞 99%以上都是時間名詞（Nd 類），類似的情形不勝枚舉，表一列出幾個例子。由於詞類和前後文語境之間有選擇性的關係，因此可以利用語境訊息，從未知詞的可能候選詞類中，進一步選出最可能的詞類，而不直接用候選詞類的第一名作爲未知詞的詞類猜測。採用語境規則進一步選擇詞類的方法，將遭遇兩個困難，第一個困難是如何找出有用的語境規則，由於語境的多樣性，不同的未知詞和語境之間有太多太複雜的搭配關係，不容易以人爲的方式歸納出規則，第二個困難是語境規則和候選詞類的原始權重如何調整，以找到最佳的平衡點。本論文提出一個從語料庫中自動抽取語境規則的方法，並且利用這些語境規則來協助判定未知詞的詞類。

語境與未知詞詞類	語境出現頻率	詞類相符的機率
公共 -> (Na)	12	0.916667
缺乏 -> (Na)	14	0.928571
主任 -> (Nb)	144	0.993056
市長 -> (Nb)	130	0.953846
位於 -> (Nc)	84	0.916667
民國 -> (Nd)	594	1.000000
西元 -> (Nd)	149	1.000000
(A) <- 麻痺	16	0.937500
(Na) <- 肝炎	106	0.990566
(Na)<- 們	37	0.972973
(Nb) <- 小姐	87	0.977011
(Nb) <- 將軍	54	0.981481
(Nb) <- 教授	155	0.851613

表一、語境與未知詞詞類之間的相關性

2. 詞類判定規則的取得

在本文中我們嘗試從語料庫中學習由語境來判定詞類的規則。訓練語料庫是使用中央研究院平衡語料庫 2.0 版，其中包含了 350 萬詞。語料庫中的每一個句子都已經經過分詞的處理，以空白字元隔開每一個詞，並且在每一個詞的後面有詞類標記。我們將語料庫分成兩個部分，其中的 300 萬詞當作規則的訓練資料，另外的 50 萬詞則當作測試資料。

2.1 規則的抽取

語境規則描述了一個未知詞詞類在實際語料庫中，與語境之間的關係。例如規則'所長->Nb'，說明了在語料庫中，所長一詞後面所接的未知詞應該是 Nb 類。本實驗抽取語境規則的方法是類似於[Brill 95, Chen 97]所使用的錯誤驅動學習法(error-driven learning method)，不同於 Brill 方法的地方在於，此一研究是用未知詞驅動而非錯誤驅動。在抽取規則之前，必須先設定規則的基本型態。本實驗總共使用了九種基本的型態。這九種型態如下所示：

==========

a. $word_{-1}$ -> category, 例如：'所長->Nb'，'很->VH'。
b. $word_{+1}$ -> category, 例如：'先生->Nb'，'警局->Nc'。
c. $category_{-2}$, $category_{-1}$ -> category, 例如：'A,Caa->A'，'Nb,Caa->Nb'。
d. $category_{+1}$, $category_{+2}$ -> category, 例如：'Caa,A->A'，'Caa,Nb->Nb'。
e. $category_{-1}$, $category_{+1}$ -> category, 例如：'VJ,VJ->Na'，'A,D->Na'。
f. $word_{-2}$, $category_{-1}$ -> category, 例如：'女,Na->Nb'，'新任,Na->Nb'。
g. $category_{+1}$, $word_{+2}$ -> category, 例如：'Cab,人->Nb'，'D,組成->Na'。
h. $word_{-2}$ -> category, 例如：'有效->Na'，'位於->Nc'。
i. $word_{+2}$ -> category, 例如：'報導->Nb'，'大小->Na'。

==========

其中 $word_{-i}$ 表示未知詞的前第 i 個詞，$word_{+i}$ 表示未知詞的後第 i 個詞，同樣的 $category_{-i}$ 表示未知詞的前第 i 個詞的詞類，$category_{+i}$ 表示未知詞後第 i 個詞的詞類。

在抽取規則的方法上，使用已經標記好的語料庫爲本，抽取的程序如下：

規則抽取程序：

1. 對已經標記好的訓練語料庫中的每一個詞，如果有辭典中找不到的詞
   1.1 依據該詞的前後文，以及該詞所標記的詞類，產生 9 類的規則

以下面的句子爲例：

職位(Na) 低(VH) 的(DE) 不(D) 具(VJ) 裁決權(Na) ，(COMMACATEGORY)

'裁決權(Na)'一詞沒有收錄在辭典中，所以依據裁決權的前後文以及其詞類，產生規則：

a . 具->Na
b. ，->Na
c. D,VJ->Na
d. --
e. VJ,COMMACATEGORY->Na
f. 不,VJ->Na
g. --
h.不->Na
i. --

其中 d, g, i, 類型由於條件不足，不產生規則。

2.2 規則的評分

並不是每一條從語料庫中抽出來的規則都具有相同的判斷能力，語境規則只能判斷一個未知詞詞類在某個語境狀態下的可能程度。例如'院長'一詞後面所接的未知詞93.33%是 Nb 類，仍有 6.66%可能是其他的詞類。因此，爲了給每一條規則評分，我們必須對抽出的規則做了一些統計，計算對於每--條規則在訓練語料庫中匹配到未知詞的語境之次數，以及計算其正確判斷詞類之次數。評分的公式如下所示：

Score-of-Rule(r,cate) = 規則 r 在語料庫中正確匹配 *cate* 的次數 / 規則 r 在語料庫中匹配的次數

每一條語境與未知詞詞類關係的規則的分數，代表在某一語境之下，對於某一未知詞詞類的支持度。

3. 未知詞詞類的判別

在判別未知詞的程序上，主要分成兩個步驟：第一個步驟是以(詞首字,詞類)及(詞尾字,詞類)的相關訊息來猜測未知詞詞類，並取其前三名的猜測作爲候選詞類，第二個步驟是使用語境與未知詞詞類關係的規則，從候選詞類中選出比較可靠的詞類。本文的焦點主要在探討第二個步驟，第一個步驟請參考[Chen 97]。一個未知詞在匹配語

境與未知詞詞類關係規則的時候，一般在每一類型的規則中，都會匹配到一條規則，而獲得一個分數。在方法上，我們是以獲得分數最高的的詞類爲未知詞的詞類。在實驗中，我們一共使用了如表三中所列的 9 種類型的規則，所以一個未知詞將會得到 9 個分數，再加上第一個步驟猜測時獲得的分數，則一共有 10 個分數(註：如果未知詞的語境在某一種類型的規則中並未出現，則所得分數爲零)。由於第一個步驟和第二個步驟的評分標準不同，並且 9 種規則對於詞類判斷的能力也不盡相同。所以在加總分數的時候，每一個分數都有一個權重。如何調整權重以找到最佳的平衡點，是實驗中的一項難題。

3.1 權重的調整

一個未知詞詞類和它的語境在一種類型的語境規律中，只會匹配到一條規則。所以對於未知詞的詞類 *cate* 而言，在語境規律類型 *j* 中只會匹配到一條規則假設爲 $Rule_n$，則語境規律類型 *j* 對於 *cate* 的評分可以表示成：

$$Score\text{-}of\text{-}Rule\text{-}Type_j(\text{cate}) = Score\text{-}of\text{-}Rule(Rule_n, \text{cate})$$

，如果將每一個分數乘上一個比重來求得總分，則一個未知詞總分的計算式可以表示爲：

$$score(cate) = MI(cate) + \sum_{j=1}^{9} W_j \cdot Score\text{-}of\text{-}Rule\text{-}Type_j(cate)$$

其中 *cate* 表示未知詞的候選詞類，MI 表示以(詞首字,詞類)及(詞尾字,詞類)相關訊息量賦予詞類 *cate* 的分數，W_j 表示第 *j* 類型規則的權重，$Score\text{-}of\text{-}Rule\text{-}Type_j(cate)$表示詞類 *cate* 在第 *j* 類型規則中所匹配到規則的分數，。在權重調整上，是以貪婪法(greedy method)來取得最佳值。首先，給每一個權重一個初始值，然後在訓練語料庫中測試，得到一個召回正確率值。其次，調整 W_1 使召回正確率增加，直到召回正確率不再有明顯的增加爲止。接著用同樣的方法調整 $W_2 \ldots W_9$，調整完 W_9 之後再從頭調整一次，調整後，如果召回正確率還有明顯的增加，就再從頭調整，如此重複調整，直到召回正確率沒有明顯的增加爲止。

3.2 判別的演算法

在本章的開頭已經說明了判別未知詞的主要步驟，在觀念上十分單純。然而在判別未知詞的過程中還將遇到一個難題，語境中如果包含了未知詞將使得語境的匹配變的複雜許多。下面是一個未知詞的語境中包含未知詞的例子：

另(Nes) 建議(VE) 由(P) 政風室(Nc,0.482;Na,0.223;VA,0.162) 閻琴南(Nb,0.427;Nc,0.259;Na,0.148) 負責(VL) 推動(VC) 小組(Na) 成立(VC)[+nom] 事宜(Na) 。

在上例中要以語境規則判斷未知詞'政風室'的詞類，必須使用到'閻琴南'的詞類，不幸的是'閻琴南'本身也是一個未知詞。

如果每一個詞都是未知詞，要真正找到最佳詞類猜測，必須考慮所有的詞類組合，即使採用 dynamic programming 的方法，依然有太多及複雜的計算，因此我們只用從左到右的依序處理方式，只以區域最佳解(local maximal)爲滿足，不追求真正的全域最佳解(global maximal)。由於在實驗中，語境只使用到前面兩個詞以及後面兩個詞，所以，在處理時，可以假設一個包含 5 個詞的詞窗：

$word_{-2}(cate_{-2})\ word_{-1}(cate_{-1})uword(cate_{11}..cate_{1i})word_{1}(cate_{11}..cate_{1j})\ word_{2}(cate_{21}..cate_{2k})$

假設 uword 爲即將處理的未知詞，由於處理未知詞的方向是由最左邊的詞一個接著一個向右處理，所以 $word_{-2}$ $word_{-1}$ 已經處理過，可以視爲已知詞。而 $word_1$, $word_2$ 都還沒有處理過，將其視爲具有 i 個及 j 個候選詞類的詞，如此一來，對於已知詞而言，其候選詞類的個數爲 1，對於未知詞而言其候選詞類的個數爲 3。於是我們可以把問題看成是一個最佳路徑選擇的問題。假設 $score_{m,n}(cate_l)$表示未知詞的候選詞類 l，在語境 $word_1$ 的詞類爲 $cate_m$， $word_2$ 的詞類爲 $cate_n$ 時，所獲得的分數，則 $cate_l$ 的最佳分數可以簡化爲：

$$score_{opt}(cate_l) = \underset{m,n}{MAX}\{score_{m,n}(cate_l)\}$$

最後在比較每個候選詞類的最佳分數，然後選擇分數最高的候選詞類爲未知詞的猜測詞類。

以上面的句子爲例，要處理未知詞'政風室'的時候，詞窗爲：

建議(VE) 由(P) 政風室(Nc,0.482;Na,0.223;VA,0.162) 閻琴南(Nb,0.427;Nc,0.259;Na,0.148) 負責(VL)

對於'政風室'的候選詞類 Nc 而言，分別假設'閻琴南'的詞類爲 Nb, Nc, Na 三種路徑做處理。

路徑 1. '建議(VE) 由(P) Nc 閻琴南(Nb) 負責(VL)' 得到 $score_{1,1}$(Nc)

路徑 2. '建議(VE) 由(P) Nc 閻琴南(Nc) '負責(VL)' 得到 $score_{2,1}$(Nc)

路徑 3. '建議(VE) 由(P) Nc 閻琴南(Na) 負責(VL)' 得到 $score_{3,1}$(Nc)

所以'政風室'候選詞類 Nc 的最佳分數爲：

$score_{opt}$(Nc) = $MAX\{score_{1,1}$(Nc), $score_{2,1}$(Nc), $score_{3,1}$(Nc)$\}$

同樣的方法可以得到'政風室'候選詞類 Na 及 VA 的分數 $score_{opt}$(Na)及 $score_{opt}$(VA)。最後比較 Nc, Na, VA 三個候選詞類的最佳分數，假設 Nc 的最佳分數最高，則選擇 Nc 爲'政風室'的詞類。

4. 實驗結果

本實驗的測試資料是使用中央研究院語料庫，中研院語料庫 2.0 版總共有 350 萬詞。其中的 300 萬詞當成訓練語料庫，另外 50 萬詞當做測試語料庫。而其中出現在 CKIP 辭典中的詞被視爲已知詞。目前 CKIP 辭典一共收錄了大約八萬目詞，每一個詞項都包含他的語法類別以及文法訊息。一個沒有收錄在 CKIP 辭典中的詞，如果也沒有被識別爲外來語(通常是文章中夾雜的英文字)的話，則被視爲是一個未知詞。在中央研究院語料庫中，總共有 52 種不同的詞類標記。而其中只有 14 種詞類具有較高的滋生力，其他的詞類通常是虛詞或是低滋生力的詞類。因此，我們的實驗只針對 14 種高滋生力的詞類。在訓練語料庫中，一共有 135896 個未知詞，而在測試語料庫中則有 21588 個未

知詞。表二列出 14 此種詞類，以及未知詞在各種詞類的頻率分佈情形。

Category	Training	Testing	Meaning of the Categories
A	1911	285	/*non-predictive adjective*/
Na	37646	5641	/*common noun*/
Nb	42853	6619	/*proper noun*/
Nc	16346	2242	/*location noun*/
Nd	11845	2037	/*time noun*/
VA	3985	656	/*active intransitive verb*/
VC	8757	1663	/*active transitive verb*/
VCL	1484	307	/*active transitive verb with locative object*/
VD	642	134	/*ditransitive verb*/
VE	991	257	/*active transitive verb with sentential object*/
VG	1675	295	/*classificatory verb*/
VH	5437	1073	/*stative intransitive verb*/
VHC	683	88	/*stative causative verb*/
VJ	1641	291	/*stative transitive verb*/

表二、未知詞在 14 種高滋生力詞類中的分佈情形

4.1 規則抽取的結果

我們從語料庫中自動抽取了 9 種不同型態的語境—未知詞詞類關係的規則，並且計算每一條規則在語料庫中匹配到未知詞語境的頻率，以及正確匹配的頻率。例如 word$_{-1}$ -> category 類的規則中，'院長->Nb'一共匹配了 45 次，其中有 42 次是正確的匹配，亦即在訓練語料庫中，'院長'一詞後面接未知詞一共出現了 45 次，其中有 42 次所接的未知詞是 Nb 類。在附錄一中所列的是一些規則的樣本。在 300 萬詞的訓練語料庫中，去除匹配次數小於 3 次的規則之後，一共抽得 113327 條規則，各類型的規則數量分佈如表三所示。

規則類型	規則條數
a. $word_{-1}$ -> category	13450
b. $word_{+1}$ -> category	13572
c. $category_{-2}$ $category_{-1}$ -> category	7238
d. $category_{+1}$ $category_{+2}$ -> category	6802
e. $category_{-1}$ $category_{+1}$ -> category	8513
f. $word_{-2}$ $category_{-1}$ -> category	15943
g. $category_{+1}$ $word_{+2}$ -> category	15027
h. $word_{-2}$ -> category	16125
i. $word_{+2}$ -> category	16657
規則總數	113327

表三、各類型語境與未知詞詞類關係規則的數量分佈

4.2 詞類判別的結果

在探討結果以前先定義實驗的召回率、精確率以及覆蓋率：

召回率 = 未知詞詞類爲 cat 且被正確猜測的詞數 / 未知詞詞類爲 cat 的總詞數

精確率 = 未知詞詞類爲 cat 且被正確猜測的詞數 / 被猜測爲詞類 cat 的總詞數

前 n 名覆蓋率 = 未知詞詞類爲 cat 且正確詞類包含在猜測的前 n 名內 / 未知詞詞類爲 cat 的總詞數

表四是以 300 萬詞的訓練語料庫做內部測試(inside test)的結果。其中第三欄 MI(1)的召回率是表示，只以詞首字—詞類及詞尾字—詞類關係猜測未知詞詞類，取其第一名爲未知詞詞類的召回率。第六欄的召回率則是以語境與未知詞詞類關係規則輔助猜測的召回率。

類別	未知詞數	MI(1)召回率	MI(1)精確率	MI(3)覆蓋率	召回率	精確率
A	1911	89.80%	34.74%	98.90%	78.23%	73.83%
Na	37646	69.77%	89.62%	97.20%	90.62%	87.87%
Nb	42853	85.59%	91.83%	97.57%	93.83%	92.06%
Nc	16346	85.05%	79.90%	97.92%	81.23%	92.43%
Nd	11845	97.89%	91.48%	99.21%	97.89%	96.65%
N	108690	90.45%	98.98%	99.21%	98.06%	97.91%
VA	3985	65.04%	43.78%	94.93%	60.43%	76.91%
VC	8757	67.08%	80.48%	98.16%	86.64%	85.91%
VCL	1484	95.89%	44.99%	99.53%	93.19%	68.94%
VD	642	95.17%	56.37%	99.69%	94.39%	84.17%
VE	991	92.23%	44.30%	98.99%	84.66%	67.28%
VG	1675	98.99%	82.24%	99.82%	98.69%	88.40%
VH	5437	71.20%	63.56%	94.74%	71.11%	87.51%
VHC	683	98.24%	57.84%	99.71%	96.78%	75.80%
VJ	1641	88.42%	50.50%	98.54%	85.74%	73.55%
V	25295	94.84%	75.75%	99.68%	91.73%	92.78%
總計	135896	80.37%		97.61%	89.11%	

表四、內部測試的結果

從表四的召回正確率來看，詞首—詞類及詞尾—詞類關係猜測的召回正確率爲80.37%，經過語境規則從前三名中輔助判定詞類召回正確率達到 89.11%，召回正確率提高了 8~9 個百分點。而其中 Na 類、Nb 類及 VC 類的召回率提升了，但是其他類的詞類召回率反而下降了。追究其原因發現 這些召回率下降的詞類，他們原本猜測的精確率都比語境規則判斷的精確率低很多，也就是說，用語境規則的猜測方式提高了各類詞類猜測的精確率，並且提高了整體的召回正確率。

類別	未知詞數	MI(1)召回率	MI(1)精確率	MI(3)覆蓋率	召回率	精確率
A	285	81.75%	27.74%	91.93%	63.16%	52.33%
Na	5641	65.48%	85.55%	96.10%	86.23%	82.52%
Nb	6619	82.14%	91.64%	96.90%	90.06%	90.46%
Nc	2242	83.99%	77.52%	96.16%	75.69%	86.54%
Nd	2037	96.02%	86.51%	97.89%	95.19%	92.55%
N	16539	88.87%	98.37%	98.85%	96.77%	96.76%
VA	656	53.96%	38.52%	89.94%	47.71%	64.94%
VC	1663	60.61%	76.60%	96.39%	79.49%	80.46%
VCL	307	92.83%	47.42%	98.37%	83.06%	63.43%
VD	134	94.03%	56.00%	96.27%	90.30%	79.61%
VE	257	86.77%	49.78%	97.28%	78.21%	67.91%
VG	295	95.59%	73.63%	99.32%	95.59%	79.21%
VH	1073	66.73%	59.42%	92.17%	62.44%	77.19%
VHC	88	89.77%	41.58%	94.32%	84.09%	51.39%
VJ	291	84.19%	47.12%	96.22%	75.26%	60.66%
V	4764	93.70%	76.87%	99.71%	89.67%	90.82%
總計	21588	76.53%		96.19%	83.83%	

表五、外部測試的結果

表五是外部測試的結果，以詞首—詞類及詞尾—詞類關係猜測的召回正確率爲76.53%，經過語境規則從前三名的猜測中判斷詞類，召回正確率提升到 83.83%，提升了約 7~8 個百分點。我們從表中發現跟內部測試類似的結果，Na 類, Nb 類,及 VC 類的召回率提升了，而其他詞類的召回率下降了，但是召回率下降的詞類，其精確率都有非常顯著的提升。

4.3 與二連詞(bigram)詞類標記模型的比較

在實驗中，我們也嘗試使用二連詞(bigram)的統計機率模型[Church 1993, Su 1996]，來輔助預測未知詞的詞類。二連機率模型的公式如下：

$$cat'_i = \arg\max_{cat_i} P(cat_i \mid cat_{i-1}) * P(word_i \mid cat_i)$$

二連機率 $P(cat_i|cat_{i-1})$

$P(cat_i|cat_{i-1})$和 P(cat|word)同樣是以中央研究院平衡語料庫的 300 萬詞估算而得。由於每一個未知詞的 P(word|cat)機率值無法從語料庫中估算，因此我們以 P(cat|word)取代。而 P(cat|word)的值是以不同詞類猜測值爲權重分配而得。例如未知詞'陳年品'以 Dice 測度前三名猜測依序爲 Nb、Na、以及 VC 類，其猜測值分別爲：

猜測值(Nb) = 8.48

猜測值(Na) = 7.327

猜測值(VC) = 2.956

因此我們假設：

P(Nb|陳年品) = 猜測值(Nb) / 總分 = 0.452

P(Na|陳年品) = 猜測值(Na) / 總分 = 0.390

P(VC|陳年品) = 猜測值(VC) / 總分 = 0.158

在實際演算過程中，我們是以 *W**log(P(cat|word))來取代未知詞 *word* 的 log(P(word|cat))值，*W* 爲調整的權重。對於未知詞而言，由於 $P(cat_i|cat_{i-1})$和 P(cat|word)兩個值的來源不同，在合併的時候，以 *W* 爲權重調整兩個值的比重以得到最佳的結果。實驗結果，同樣以中央研究院平衡語料庫另外的 50 萬詞爲測試語料庫，召回正確率從 76.53%提升到 79.97%，提升的幅度不如我們所提出的方法。

5. 結論與未來的研究

從語料庫自動產生判別規律的方法，經實際驗證是一種非常有效的方法，不但產生容易，而且有較佳的覆蓋率可以照顧到許多不同的類型。以未知詞的詞類判別而言，可以從語料庫中產生超過百萬條不同的規律，只是大部分是沒有什麼效果的。出現的頻率太低的規則可以被忽略而不會降低召回正確率。至於有些規則有包含關係，例如兩個語境相關，就可能包含於一個語境相關的規則中，只是它們各有不同的權重，因此並未被刪除。以語境與未知詞詞類關係的規則來輔助猜測詞類，大約可以提升 7~8

個百分點的召回正確率。距離前三名的覆蓋率還有一段距離，應該還有很大的提升空間。我們觀察所抽取的規則發現，有 50%的規則在語料庫中出現的頻率少於十次，並且在訓練語料庫中大部分的未知詞都是屬於 Na 類及 Nb 類，其他詞類的未知詞樣本太少，訓練出來的規則所具有的代表性不足，對於判斷的正確性有很大的影響。如何針對不同詞類調整權重，也是未來可能的研究。事實上，本研究所提出的方法，不僅適用於未知詞，也適用於任何具有多重詞類的已知詞的詞類判定上。因此，可以應用在詞類標記的工作上。此一方法和 Brill 所提出的方法，最大的不同是 Brill 的規律在給分時只有 0 和 1，而本文提出的方法，每一種規則有不同的權重，而且每一條規則有不同的給分。

參考文獻

陳信希、李振昌, 1994, "中文文本組知名之辨識" *Communications of COLIPS, VOL 4, NO 2, Page 131-142.*

孫茂松、黃昌寧、高海燕、方捷, 1994, "中文姓名的自動辨識" *Communications of COLIPS, VOL 4, NO 2, Page 143-149.*

Blahut, Richard E., 1987, Principles and Practice of Information Theory, *Addison-Wesley Publishing Company.*

Brill, Eric, 1995, "Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging." *Computational Linguistics Vol. 21, No. 4, pp. 543-566.*

Chen, C. J., M. H. Bai, K. J. Chen, 1997, "Category Guessing for Chinese Unknown Words." *Proceedings of the Natural Language Processing Pacific Rim Symposium 1997: pp. 35-40. NLPRS '97 Thailand.*

Chen, H. H. and J. C. Lee, 1994, "The Identification of Organization Names in Chinese Texts." *Computer Processing of Chinese and Oriental Languages, Vol. 8, No. 1, June*

1994, pp. 75-85.

Chen, K. J., C. R. Huang, L. P. Chang & H.L. Hsu, 1996, "SINICA CORPUS: Design Methodology for Balanced Corpora." *Proceedings of PACLIC 11th Conference, pp. 167-176*

Chen, Keh-Jiann, Ming-Hong Bai, 1997, "Unknown Word Detection for Chinese by a Corpus-based Learning Method." *Proceedings of the 10th Research on Computational Linguistics International Conference, pp159-174.*

Huang, Chu-Ren, Wei-Mei Hong, and Keh-jiann Chen, 1994, "An Information Based Lexical Rule of Abbreviation." *the Proceedings of the Second Pacific Asia Conference on Formal and Computational.*

Church, K. W., & R. L. Merser, 1993, "Introduction to the Special Issue on Computational Linguistics Using Large Corpora." *Computational Linguistics, Vol. 19, #1, pp. 1-24*

Lee, J.C. , Y.S. Lee and H.H. Chen, 1994, "Identification of Personal Names in Chinese Texts." *Proceedings of 7th ROC Computational Linguistics Conference.*

Mikheev, A., 1996,"Unsupervised Learning of Word-Category Guessing Rules." *Proceedings of ACL-96.*

Smadja, F.A., K.R. McKeown, and V. Hatzivassiloglou, 1996, " Translating Collocations for Bilingual Lexicon: A Statistical Approach." *Computational Linguistics, Vol. 22, No. 1.*

Su, Keh-Yih, Tung-Hui Chiang, & Jing-Shin Chang, 1996," An Overview of Corpus-Based Statistics-Oriented Techniques for Natural Language Processing." *Computational Linguistics and Chinese Language Processing, vol. 1, no. 1, pp. 101-157.*

附錄一、語境—未知詞詞類關係規則的一些例子

a. $word_{-1}$ -> category
主任->Nb 144 0.993056
位於->Nc 84 0.916667
積極->VC 10 1.000000

b. $word_{+1}$ -> category
肝炎->Na 106 0.990566
般->Na 24 0.916667
女士->Nb 64 1.000000

c. $category_{-2}$, $category_{-1}$ -> category
Nb, PAUSECATEGORY->Nb 1267 0.930545
Nd, Caa->Nd 253 0.964427
Nh, Caa->Nb 84 0.916667

d. $category_{+1}$, $category_{+2}$ -> category
Caa, Nb->Nb 732 0.939891
Caa, Nd->Nd 313 0.913738
DASHCATEGORY, Nb->Na 191 0.989529
Nc, VE->Nb 583 0.914237

e. $category_{-1}$, $category_{+1}$ -> category
A, D->Na 82 0.975610
Neu, VG->Na 63 0.888889
A, P->Na 21 0.952381
Nb, VI->Na 13 0.923077

f. $word_{-2}$, $category_{-1}$ -> category
下午, Nd->Nd 72 0.972222
女, Na->Nb 32 0.937500
中心, Na->Nb 37 0.945946
平行式, Caa->A 4 1.000000

g. $category_{+1}$, $word_{+2}$ -> category
跟->Nb 16 0.937500
高興->Nb 13 0.923077
表示->Na 55 0.963636

h. $word_{-2}$ -> category
位於->Nc 59 0.915254
前任->Nb 14 0.928571
訂->Nd 21 0.952381

i. $word_{+2}$ -> category
掩埋場->Nc 20 0.950000
現任->Nb 13 0.923077

附錄二、測試結果的例子

第一行是經過自動斷詞與自動詞類標記後的結果，其中詞類欄內爲'?'的詞表示爲未知詞。第二行是經過詞首—詞類及詞尾—詞類關係猜測後的結果，每一個未知詞後面都有3個候選詞類，每一個候選詞類後面都有一個值，是猜測時所給的分數。第三行是經過語境—未知詞詞類關係規則判定詞類後的結果。

****

珍貴(VH) 的(DE) 古(VH) 陶壺(?) 並(D) 被(P) 視爲(VG) 貴族(Na) 的(DE) 傳家(VI) 之(DE) 寶(Na) ，

珍貴(VH) 的(DE) 古(VH) 陶壺(Na,0.421;Nb,0.215;VJ,0.159) 並(D) 被(P) 視爲(VG) 貴族(Na) 的(DE) 傳家(VI) 之(DE) 寶(Na) ，

珍貴(VH) 的(DE) 古(VH) 陶壺(Na) 並(D) 被(P) 視爲(VG) 貴族(Na) 的(DE) 傳家(VI) 之(DE) 寶(Na) ，

****

清代(Nd) 臺灣(Nc) 早期(Nd) 的(DE) 移墾(?) 者(Na) ,由於(Cbb) 其(Nep) 祖籍(Na) 之(DE) 不同(VH) ，

清代(Nd) 臺灣(Nc) 早期(Nd) 的(DE) 移墾(VCL,0.363; VC,0.287; VA,0.207) 者(Na) 由於(Cbb) 其(Nep) 祖籍(Na) 之(DE) 不同(VH) ，

清代(Nd) 臺灣(Nc) 早期(Nd) 的(DE) 移墾(VC) 者(Na) 由於(Cbb) 其(Nep) 祖籍(Na) 之(DE) 不同(VH) ，

****

另(Nes) 建議(VE) 由(P) 政風室(?) 閻琴南(?) 負責(VL) 推動(VC) 小組(Na) 成立(VC)[+nom] 事宜(Na) 。

另(Nes) 建議(VE) 由(P) 政風室(Nc,0.482;Na,0.223;VA,0.162) 閻琴南(Nb,0.427;Nc,0.259;Na,0.148) 負責(VL) 推動(VC) 小組(Na) 成立(VC)[+nom] 事宜(Na) 。

另(Nes) 建議(VE) 由(P) 政風室(Nc) 閻琴南(Nb) 負責(VL) 推動(VC) 小組(Na) 成立(VC)[+nom] 事宜(Na) 。

****

也(D) 肩負起(?) 更(D) 重大(VH) 的(DE) 任務(Na) 。

也(D) 肩負起(VJ,0.325;VC,0.293;VA,0.193) 更(D) 重大(VH) 的(DE) 任務(Na) 。

也(D) 肩負起(VC) 更(D) 重大(VH) 的(DE) 任務(Na) 。

****

應用動、靜態詞典以加速鍵盤輸入中文之方法
A Dynamic-and-Static-Dictionaries Based Method for Accelerating Chinese-Character Inputting with Keyboard

古鴻炎　　楊仲捷
Hung-yan Gu　　Chung-Chieh Yang
國立台灣科技大學電機系
Department of Electrical Engineering
National Taiwan University of Science and Technology
E-mail: root@guhy.ee.ntust.edu.tw　Fax: 886-2-27376699

摘要

在中文輸入上，鍵盤是一種普遍被使用的工具。為了加快以鍵盤輸入中文的速度，本文研究了以詞為單位來輸入的觀念、並提出實踐的方法，而將舊有的以字為單位來輸入的觀念加以擴充了。關於詞輸入觀念的實踐，我們研究了兩個重要的課題，第一個是，可減少按鍵次數之多字詞按鍵規則的設計，我們以注音輸入法為例，提出了一種設計方法，可據以獲得簡單且相容的按鍵規則；第二個是，可動態地記錄專有名詞、及支援短詞組合成長詞之動態詞典的設計，我們提出了一種基於赫序表的資料結構，使得動態詞典的功能、及查尋速度能夠符合實際需求。除了研究相關的問題，我們也實際去製作出一個整合了單字注音連續轉換、及多字詞直接輸入的原型系統，而驗證了所提出方法的可行性。

1. 導言

關於中文輸入之研究，在許多方向上都有人投入心力，如光學文字辨識(Chuang 1995, Huang 1997)、線上手寫輸入(蔡 1997)、國語聽寫機(Wang 1997)、鍵盤輸入(古 1992)等，其中以鍵盤輸入之方向較容易實行，也擁有最多的使用者，但這並不表示以鍵盤爲基礎的中文輸入方法(如注音類的漢音、自然、忘形，字根類的倉頡、大易)，就無再改進之空間，或者將窮途末路而不再需要研究，相反地，基於省力性(手寫輸入易疲勞)、隱密性、錯字更正之方便性(只用語音聽寫不易操作錯字之更正)、費用、輸入速率等等因素之考慮，我們覺得近期內鍵盤仍將是中文輸入上一種普遍被使用的工具(包括滑鼠按的圖像式鍵盤)。因此，若有一種能夠改進鍵盤中文輸入的效率(輸入一個中文字所需之按鍵數)、或速率(單位時間內輸入的字數)的方法被提出，那將讓爲數眾多的中文輸入

者受益，基於此一信念，我們遂投入於改進鍵盤中文輸入之研究，如今也獲得了幾許成果而可供參考。

本文提出的中文輸入之加速方法，其所依據的基本觀念是，將過去只能以中文字爲單位來輸入的作法擴充成能夠以詞爲單位來輸入，這裡所謂的詞其含意是廣義的，如"做了再說"、"中看不中用"都可被當成是詞，且詞的長度可以從 1 到一個預先設定的最大值(如 13)，說以片語爲輸入單位應較恰當，不過本文內仍將稱之爲詞。我們所以會想到以詞爲單位來輸入，是考慮到原始的注音輸入法裡，有同音字選字的問題，而當以二字詞或更多字之詞的注音爲單位來看時，同音詞的問題就小很多了，因此，我們想到進一步去爲二字以上之多字詞來定義一些能夠減少按鍵次數的按鍵規則，以加速中文之輸入或達到節省氣力之目的。這裡所說的以詞爲單位來輸入的觀念，和以往注音輸入法裡的連續注音轉換之觀念(Chen 1987)並不相衝突，因爲本文所說的詞是包含字的，當輸入的是單字的注音時，仍可去使用連續注音轉換的機制，事實上我們已製作出一個整合多字詞直接輸入、和連續單字注音轉換的原型系統。除了注音輸入因爲有同音字而需要連續轉換之處理，其實字根類的一些簡化的輸入法也需要類似的連續轉換處理(Fan 1991)。

雖然說以詞爲單位來輸入中文的觀念並不難，並且這樣的觀念對注音類或字根類的輸入法都可應用，但是，如何去實行就不是一件容易的事了，因爲有兩個實際的問題需要解決，一個是按鍵規則的設計與實行，而另一個是動態詞典的設計與製作。輸入二字以上多字詞的按鍵規則的設計，除了要減少按鍵的次數以達到加速輸入中文的目的外，各種長度的詞的按鍵規則要能夠相互相容，即不需額外按鍵來選擇接著要輸入之詞的詞長，這樣才可讓使用者隨心所欲，想輸入 N 字詞就直接用 N 字詞的按鍵規則， N 的值可以是傳統上所用的 N=1(即以字爲單位)，或新加入的 N=2,3, ..., 13(本文設定之詞長最大值)；除了考慮按鍵規則間的相容性之外，也要考慮如何設計一組既容易理解、又幾乎不需背誦的按鍵規則，不然按鍵規則又多又複雜而令人生畏，那將引不起使用者的使用意願，也就讓所訂的按鍵規則變得沒有意義了。

考慮以詞爲單位來處理使用者的按鍵輸入時，由於本文所謂的詞也含蓋了傳統上的字的單位，因此，在一般注音輸入法裡提供的連續注音轉換之功能，可說是基本且必備的功能，而此功能內所用到的一個支援注音到詞語轉換、查尋的詞典，我們也可再拿來

利用，以支援作多字詞之輸入按鍵轉換成詞語的查詢動作，這樣一個預先準備好的、用以儲存一般常用詞語的詞典，本文稱之爲靜態詞典(static dictionary)。若僅使用靜態詞典來查詢多字詞按鍵輸入可能對應到的語詞，將會面臨如下所說的困難，第一個是，對於靜態詞典裡未收錄的詞彙，如專業術語、專有名詞、及個人之慣用語，如何在輸入過一次後就能操作多字詞的按鍵規則來輸入？若規定使用者要自己去爲每一個新詞彙操作一次登錄的程序，很明顯地會造成使用者的不便，一者是費力氣，二者是要停頓思慮去想想是否需將碰到的新詞登錄到詞典；第二個困難是，一般的靜態詞典並不支援短詞自動組合成長詞的功能，使得一個長詞如"後天免疫不全症候群"，即使已經被輸入過數次，下次要輸入時，還是得依序逐一輸入"後天"、"免疫"、"不"、"全"、"症候群"等詞語(假設未操作過登錄的程序)，而不能夠直接(不用操作登錄之程序)就去操作 9 字詞之按鍵規則。請不必擔心 9 字詞的按鍵規則是否很複雜，其實我們設計的按鍵規則簡單得一看就記住了，且不必理會欲輸入的長詞(字數大於 3 者)究竟有幾個字。

雖然過去已有不少人研究了連續注音轉換成中文字之問題(Lin 1987, Gu 1991, Hsu 1994, Kuo 1995, Ho 1997)，但是對於前面提到的兩個關於靜態詞典的缺點，則尚未看到有人提出實際且有效的解決方法。因此，我們便思考此一問題，而想到引進動態詞典(dynamic dictionary)的觀念，並去研究動態詞典功能的實行方法，以便用動態詞典來自動記錄使者最近輸入進來的文句或詞彙，這樣，使用者就完全不需要去操作新詞登錄的程序，而且任何一個新詞彙不管其長度多長(在最大長度的限定內)，只要被輸入過一次，就確定可在動態詞典裡尋找到，如此當要再次輸入該新詞時，就可直接操作多字詞之按鍵規則了。例如輸入過一次"台北市基隆路四段"後，以後想要輸入"台北市基隆路"之詞時，就可直接操作 6 字詞之按鍵規則。

由前面的說明可知，本文倡議在原有的以字爲單位的輸入法上，再擴充支援以詞爲單位的中文輸入方式，藉由訂定減少按鍵次數的多字詞輸入之按鍵規則來加速鍵盤中文之輸入，這樣的想法可被應用到許多現有的、不管是注音類或是字根類的中文輸入法上，雖然本文裡的說明都拿注音輸入法作例子，但這並不表示只有注音輸入法才適用。要實踐以詞爲單位來輸入中文，動態詞典之觀念及製作是很重要的，所以在第二節就先說明我們對於動態詞典製作之研究成果；關於多字詞之按鍵規則的設計，在第三節裡我

們以注音輸入法爲例提出了一種相容、且簡單可行的設計；在第四節裡，我們則對同鍵詞(具有相同輸入按鍵的詞語)的問題加以分析；第五節裡則說明原型系統製作時會面臨的問題，及我們採取的作法；最後一節是結語。

2. 動態詞典設計

2.1 動態詞典的功用

回顧過去，可知動態詞典的觀念很早就被資料壓縮領域所應用(Bell 1990)，即調適型(adaptive)詞典式編碼方法，此外，建造統計式靜態詞典的問題也有不錯的方法可用。動態詞典的觀念簡單說來是，利用一個貯列(queue)之資料結構來儲存最近被處理過之資料，當新的一筆被處理過之資料要放進去時，就固定從貯列的一端放入，而當空間不夠時，就從貯列的另一端將最舊的資料移出。動態詞典之所以能夠發揮功用是因爲，一序列等待被處理的資料中，常常有反覆出現的資料片段存在，使得處理之軟體常常可從動態詞典中找到對應的已被處理過的、與等待被處理的相同資料片段，而達成特定的目標。

動態詞典爲什麼對中文之輸入有幫助？因爲我們可用大容量之動態詞典來記錄相鄰中文字的共出現(co-occurrence)關係，例如一'枝'花、一'隻'狗、一'支'筆等名詞與量詞之間的共出現關係，如此，當使用者再次輸入音節注音／ㄓ ㄍㄡˇ／時，"隻狗"之詞就會被查出；再者，動態詞典可用來記錄靜態詞典中未收錄的專有名詞、術語，如"紫杉醇"、"低鈉鹽"、"虛擬實境"，如此，當使用者要再次輸入曾被輸入過的專有名詞時，就可直接操作多字詞的按鍵規則來加速輸入；此外，還可利用動態詞典來進行短詞組合成長詞之處理，詳細作法在第五節裡說明，這樣，使用者只要輸入過一次某一長片語，以後就可直接操作多字詞的按鍵規則來快速地輸入那個長片語。由前面的說明可知，我們只使用了一個共用的動態詞典，並且爲了能夠記錄得愈多愈詳盡，詞典的大小是愈大愈好，不過，也不能大到使得搜尋速度變慢得不能接受，再者，動態詞典裡的資料在程式結束執行後也要繼續維持下去(即存檔)，這樣才能讓所記錄的資訊累積下去。

2.2 動態詞典之結構設計

由於我們要利用動態詞典來提供短詞(含單字詞)組合成長詞之功能，並且，這個功能除了支援以詞(二字以上)爲單位來輸入的方式之外，也要支援以字爲單位來輸入時用馬可夫中文模型來進行連續注音轉換之處理方式(古 1995, 古 1997)，因此，只採用簡單的貯列結構來實現動態詞典在實作上是不可行的，因爲在馬可夫模型中，一個音節若對應到10個同音字，則 5個連續音節可能組合出的五字詞就有 100,000 個，不用說6、7或更多字組成的詞了，再加上我們希望動態詞典愈大愈好，則字串比對的運算量及所需的時間是可想而知的，因此，我們就決定以赫序表(hash table)結合貯列的複合結構之設計來實現動態詞典，以赫序表的功能來加快搜尋的速度，而以貯列的功能來控制資料的存入與刪除。

爲了配合赫序表的處理，實作上就必需把一個中文句子的組成字(如: ABCDE)，拆成相鄰字組成的雙字詞序列(如: AB, BC, CD, DE)，然後將這些雙字詞(含注音資料)插入赫序表，可是如此做會伴隨產生兩個問題，第一個問題是無法迅速地確定雙字詞之間的時間次序、及存入時間，例如由兩音節/ㄓ ㄏㄨㄚ/去查出"支花"與"枝花"都在赫序表裡時，如何知道那一個是較晚存入的、及多久之前存入的，以便由存入時間來推測某一詞語再次被用到的機率，我們採用的一個解決辦法是應用郵戳(time stamp)的觀念，即每次要將一個雙字詞存入赫序表時(不管是新插入的或是更新的)，就將目前時間計數器的值取出一起存入到赫序表裡，然後將時間計數器加一。第二個問題是會發生張冠李戴的現象，假設先前輸入的語句 "一隻黑貓帶著兩支黑拐杖並撐著一支花雨傘"，已被存入動態詞典裡，則接著輸入 "一隻黑貓" 的四個音節注音時，會轉換出來的是"一支黑貓"，這是因爲 "一支" 的郵戳比 "一隻" 晚，而 "支黑" 比 "隻黑" 晚，乍看之下，也許會認爲何不依據郵戳值是否連續去判斷原先是否是接在一塊的，不過，這樣做是不可行的，例如若輸入 "一隻黑狗和二隻黑貓" 後，那是不是使得"隻黑" 與 "黑狗"不被認爲是連在一塊的，因爲"隻黑"的郵戳值會被第二次出現的"隻黑"所更新，對於這樣的問題，我們的解決方法是再增加一個赫序表，用以儲存中文句子裡相鄰三字所形成之三字詞，例如要將中文句子 ABCDE 存入動態詞典，就相當於把 AB,BC,CD,DE 等雙字詞存入第一個赫序表，而把 ABC, BCD, CDE 等三字詞存入第二個赫序表。如此，當要查尋一個四字詞 WXYZ 存不存在於動態詞典裡時，就先到第一個赫序表去看 WX,XY,YZ

是否都能比對成功，包括注音資料之比對，這樣才不會因破音字而發生如／ㄕㄥ ㄐㄧㄢˋ／被轉成"時間"而非"實踐"之現象；然後，再到第二個赫序表去看 WXY 與 XYZ 存不存在。如果都比對成功，才算是存在於動態詞典裡，如此，前述例子"一隻黑貓"的音節注音就不會轉換錯了。至於 5 、 6 、...等多字詞的檢查可依此類推。

3. 按鍵規則設計

3.1 原則與方法

前面我們曾提到，設計多字詞的按鍵規則時，要在兩個基本原則之下去考慮減少按鍵的次數，一個原則是按鍵規則間要兩兩相容，這樣才不需額外按鍵來選擇欲輸入之詞的詞長，另一個原則是按鍵規則要簡單易理解、而幾乎不需背誦。基於這兩個原則，我們爲普通注音鍵盤設計了一套多字詞的按鍵規則，詳細情形如下面的說明，雖然這裡只以注音輸入法爲例，提出對應的多字詞按鍵規則之設計，但我們認爲同樣的原則、同樣的設計方法也可應用到字根類中文輸入法裡，去設計對應的多字詞按鍵規則。

在設計注音輸入法的多字詞按鍵規則時，我們注意到注音鍵盤上的注音符號的排列方式已有許多種被提出，一種使用傳統排列方式的我們稱爲普通注音鍵盤，也就是我們要據以設計多字詞按鍵規則的鍵盤，除此之外還有如宜韻注音鍵盤(古 1992)、許氏注音鍵盤等等的排列法。基本上，注音符號的排列方式和詞爲單位來輸入的觀念並不相衝突，不過，某些排列方式的確會對詞輸入觀念之實行造成麻煩，因爲在原先字爲單位的輸入觀念裡，雖可讓二個或多個注音符號共用一個按鍵而不會有分辨不清(ambiguous)的情形，但是當考慮要設計多字詞輸入之按鍵規則時，就會發現有分辨不清的情形存在，而增加了中文輸入軟體製作上的複雜度，並增加了操作同鍵詞選詞動作的機會，因此，字單位的輸入觀念裡認爲不錯的注音鍵盤設計，不見得就一定適合在以詞爲單位來輸入的觀念裡使用，所以本文也給注音鍵盤之設計導入了一個新的考慮因素。

由於本文所謂的詞單位是包含字的單位的，所以在設計多字詞的按鍵規則時，第一個考慮到的便是如何區分使用者目前要輸入的是單字還是多字詞，爲了解決這個問題，我們就訂定如下的規則：

(R1) 多字詞的按鍵規則裡，不可用到聲調按鍵。

因爲輸入一個單字的注音時，最後一定要輸入一個聲調的按鍵，而當輸入軟體接收到一個聲調按鍵時，就可依規則(R1)確定使用者不是在輸入多字詞。這樣的規則也可應用在字根類的輸入法裡，例如倉頡輸入法裡輸入一個單字的最後一個按鍵固定是空白鍵。

能夠區分目前輸入的是單字還是多字詞之後，接著我們考慮，使用者欲輸入之多字詞的長度可能從 2 變化到 13 ，那麼輸入軟體如何知道目前被輸入之多字詞的詞長？相關的一個較基本的問題是，是否一定要知道詞長才能夠作處理？如果從軟體製作者的觀點來看，當然是希望使用者明確告知欲輸入之多字詞的長度，當反映在多字詞的按鍵規則上，可能就是要求使用者多輸入一個暗示詞長的按鍵，這樣的作法不僅會增加按鍵的次數，而使輸入速度變慢，最主要的缺點還是造成使用者的不便，因爲要記住多輸入一個暗示詞長的按鍵這件事(而輸入單字時不用)，及輸入之前還得自己先去算詞的長度。因此，我們設計多字詞的按鍵規則時，是在實作上可行的條件下儘量考慮讓使用者方便，實際上我們的規劃是，將多字詞依詞長分成三類，即二字詞、三字詞、與四字以上之多字詞，然後各別去設計各類詞的按鍵規則，以便兼顧不同詞長之詞的特性及簡化按鍵規則，其實從所設計的按鍵規則來看，可以說只有兩類詞，因爲三字詞可看成是四字詞的特例，也就是說使用者大約只需分辨欲輸入之詞是二字詞還是二字以上之詞。關於作分類的考慮因素，前面提到的一個是不同詞長之詞的特性，我們指的是，詞的長度愈長(如四字以上之詞)，則只需較小的按鍵率(按鍵次數與詞長之比率)就能夠互相區分而少有同鍵詞出現，而二字詞需要較大的按鍵率，不然發生同鍵詞的機率會很大；此外，我們把四字以上之多字詞歸爲一類，並爲它們設計通用的按鍵規則，目的是讓使用者不用去算欲輸入之詞的詞長，而這樣也讓需要記憶的按鍵規則的數目減少了。

3.2 多字詞按鍵規則

詳細說來，我們設計的雙字詞按鍵規則是:

(R2) 欲輸入雙字詞 XY 時，先按 X 的頭尾兩注音符號，再按 Y 的頭尾兩注音符號。

例如要輸入雙字詞"電腦"則按ㄉㄢㄋㄠ等四鍵，而雙字詞"實際"要按ㄕ*(Enter)*ㄐㄧ等四

鍵，也就是說注音符號不夠用時要以*(Enter)*鍵替補，以湊成四鍵。我們所以選用頭、尾而不用頭、中之兩注音符號，是因爲確定頭、尾後，中間能夠被插入的注音符號只有含不含介音ㄧㄨㄩ等四種可能，不像確定頭、中後，尾部可能插入的韻母注音符號的數量有 12 個之多，而發生更嚴重的混淆不清之情況。類似(R2)之按鍵規則，也可用在字形類的中文輸入法裡，以輸入雙字詞。接者，我們爲三字詞設計的按鍵規則是：

(R3) 欲輸入三字詞時，依序按各字最前之注音符號，然後按〕鍵。

例如要輸入三字詞"王陽明"則按 ㄨㄧㄇ〕 等四鍵。規定多按一個注音符號以外的 〕鍵，是要使按鍵數湊成四，以消除 ambiguous 之情況，因爲若不加按一個〕鍵，就不能確定使用者是要輸入單字詞、二字詞、三字詞、還是四字詞，而只有當確切知道使用者要輸入幾字詞時，若從詞典中只查到一個對應的詞語，就可立刻將該詞語送出，以免除使用者需要再按一鍵來選取所要之詞語的步驟。關於四字詞的按鍵規則，我們的設計是：

(R4) 欲輸入四字詞時，依序按各字最前之注音符號。

例如要輸入四字詞"一帆風順"則按 ㄧㄈㄈㄕ 等四鍵。由規則(R4)可知，爲什麼我們說三字詞的按鍵規則可看成是四字詞規則的一個特例。此外，應該不難看出規則(R4)與規則(R2)是有衝突存在的，因爲使用者輸入一個二字詞的四個按鍵時，可能由規則(R4)找到一個對應的四字詞，例如欲輸入"距離"時，按ㄐㄩㄌㄧ等四鍵，依據規則(R4)會對應到"金玉良言"。基本上發生這種衝突的機率很小，靜態詞典裡的二字詞經由分析只發現到 15 個會有這種衝突，因此我們依長詞優先的原則來設定規則(R4)比(R2)具有較高的套用權，即當使用者按了四次注音符號按鍵後，先試圖套用規則(R4)，如果詞典中未能找到對應的詞語，再去嘗試規則(R2)。雖然我們訂定規則(R4)具有較高的套用權，但是使用者想輸入的可能是發生衝突的那個二字詞，再者還要考慮如何銜接到五字以上多字詞之輸入，因此我們不能在套用規則(R4)成功後，看到只有一個對應的詞語就立刻將該詞語送出，而必需令使用者再按一數字鍵來選取所要的詞語，此時使用者按的鍵若是特殊的*(Esc)*鍵，就可確認他想輸入的是二字詞，然後改成去套用規則(R2)。

關於五字以上至十三字之多字詞的按鍵規則，我們的設計是：

(R5) 欲輸入五字以上之多字詞時，依序按各字最前之注音符號。

例如要輸入六字詞"台北市民政局"(假設曾輸入過此片語)則按 ㄊㄅㄕㄇㄓㄐ 等六鍵。比較規則(R4)與(R5)可知，事實上兩規則並無不同，這也就證實了我們在前面所說的，雖然詞的長度有長有短且愈長愈難算清楚，但是依據本文設計的按鍵規則，使用者不需去算所輸入長詞的字數，也無需掛慮會分不清楚該用那一條規則，而只需分辨欲輸入之詞是二字詞、三字詞、還是四字以上之詞。另一方面，從實際製作的觀點來看，規則(R4)與(R5)在實行上並無衝突或不可行的地方，當使用者輸入四個注音符號後，就先去嘗試套用規則(R4)，若詞典中找不到對應的詞語，就再去嘗試套用規則(R2)，若能夠找到對應之四字詞，就將找到的詞語顯示於螢幕上，然後等待使用者的下一個按鍵，接著，將讀到的按鍵分成三類，第一類如果讀到的是*(Esc)*鍵，則表示之前鍵入的四個按鍵是要輸入二字詞的；第二類如果讀到的是注音符號之按鍵，就可確認是，使用者想輸入比四字更長的詞，因此就套用規則(R5)，此時若未能從詞典找到對應的詞語，就將之前找到的詞語中的一個送出，若能找到對應的詞語，就再將找到的詞語顯示於螢幕上，然後再等待使用者的下一個按鍵；第三類如果讀到的是數字按鍵，就表示使用者要選取目前顯示出的詞語中的一個。在普通注音鍵盤上，由於部分的數字鍵(如 1,2,5,8,9)也是注音符號之按鍵，而產生四字以上多字詞輸入時的混淆不清情況，這樣的情況可採用前述解決規則(R2)與(R4)衝突的相同作法來化解，也就是令注音符號有較高的優先權，而先嘗試從詞典找對應的增長一字的詞語，第一種情況若找不到，就將剛輸入的按鍵當成是數字鍵，然後去選取前一次顯示出的詞語；第二種情況若能找到對應的詞語，就將找到的詞語顯示於螢幕上，然後等待使用者的下一個按鍵，此時使用者可按*(Esc)*鍵，以強迫將前一次的按鍵當成數字鍵並選取前一次顯示出的詞語。

4. 同鍵詞分析

前一節裡我們提出了一套輸入多字詞的按鍵規則，初步看來應可說是簡單易懂，但相對的一個考慮是，簡單的按鍵規則會不會導致嚴重的同鍵詞問題，即依據按鍵規則來輸入一個詞時，卻從詞典裡查出許多的詞語都可對應到所輸入的按鍵，而需操作同鍵詞

選詞的步驟，類似於傳統注音輸入法裡的同音字選字的操作，例如輸入二字詞"美妙"的 4 個按鍵ㄇㄟˇㄇㄧㄠ時，會查出"美妙"，"美貌"，"眉毛"等詞語。爲了檢視同鍵詞的發生機會及嚴重情形，我們就以原型系統裡所用的靜態詞典來進行分析，該詞典裡含有 33,275 個二字詞、8,083 個三字詞、10,448 個四字詞，詞語的收錄參考了中研院詞典(中研院 1994)的二至四字詞，經過分析後，我們得到表 1 至表 3 的數據，在這三個表裡，

表 1　二字詞之同鍵詞分析

同鍵範圍	1	2	3	4	5	6	7	8	9	10	11	12	13	14
詞個數	14,185	8,646	4,644	2,472	1,425	864	406	192	81	70	66	36	13	28
比率%	42.6	26.0	14.0	7.4	4.3	2.6	1.2	0.6	0.2	0.2	0.2	0.1	0.0	0.0

表 2　三字詞之同鍵詞分析

同鍵範圍	1	2	3	4	5	6	7	8	9
詞個數	3,418	2,188	1,314	616	280	114	70	40	18
比率%	42.3	27.1	16.3	7.6	3.5	1.4	0.9	0.5	0.2

表 3　四字詞之同鍵詞分析

同鍵範圍	1	2	3	4
詞個數	9,481	848	81	12
比率%	90.7	8.1	0.8	0.1

同鍵範圍表示輸入一個詞的按鍵後會查到的同鍵詞的個數，詞個數表示靜態詞典中有多少個詞語具有某一個特定的同鍵範圍值，例如從靜態詞典中找輸入按鍵爲ㄇㄢㄏㄚ的二字詞，只能找到詞語"漫畫"與"棉花"，因此這兩個詞的同鍵範圍值都定義爲 2 ，且都要算在同鍵範圍值爲 2 的詞個數之計數裡。由表 1 可知，靜態詞典裡有 42.6%之二字詞的同鍵範圍值爲 1 ，也就是沒有同鍵詞，而可在輸入四個注音按鍵後即刻送出對應之詞，另外 56.7%的二字詞(同鍵範圍值爲 2 至 9)需要多按一個數字鍵來選取所欲輸入之詞，至於同鍵範圍值到達 10 以上的二字詞有 213 個。接著，由表 2 可得知， 42.3%的三字詞沒有同鍵詞，而可在輸入四個按鍵後即刻送出對應之詞，其餘的需多按一個數字鍵來選取所欲輸入之詞。最後由表 3 可知， 90.7%的四字詞沒有同鍵詞，這個比率比起表 1 與表 2 裡的都高許多，因此我們可推測五字以上之多字詞的同鍵詞問題會更小。不過，對於四字以上之多字詞來說，即使輸入之按鍵只對應到一個詞，使用者還是要加按

一個數字鍵來選取欲輸入之詞，因爲輸入軟體依據前一節裡的按鍵規則(R5)會無法判斷使用者是否要輸入更長的詞，然而在需加按一個數字鍵的情況下，四字詞的輸入效率尙可達到 1.25 鍵每字。

由於表 1 、表 2 顯示，輸入二字詞或三字詞時，發生同鍵詞選詞的機率(1 - 42.3%)並不小，而將造成使用者需要常常注視螢幕的情況，因此，我們便思考可能的改進方法，後來想到一種可行的作法是，以聲調來分辨、選取同鍵詞群中的一個詞，例如要輸入二字詞"市長"時，按 尸(Enter)业尢 等四鍵，結果螢幕上會出現"師長"、"師丈"、"時裝"、"市長"等詞語，並聽到電腦發出"嗶"一聲來通知有同鍵詞之情況，此時，使用者可不必看螢幕，而直接在心裡想"市"、"長"的聲調分別是 4 聲與 3 聲，然後就去按聲調 43 所對應的按鍵而選到"市長"之詞語。這樣的方法需要 5*5 = 25 個表示聲調組合之按鍵，而數字鍵之外的按鍵數量多於 25 ，所以實行上並無困難。依據前述的按聲調鍵的方法，再去對靜態詞典中的二字詞、三字詞作分析，其中三字詞之聲調鍵由詞首兩字決定，結果我們得到表 4 與表 5 之數據，在此二表裡，同鍵範圍值爲 1 的詞個數表示，不用按聲調鍵就已無同鍵詞情形的詞語個數，加上按聲調鍵後才變成無同鍵詞情形的詞語個數，而其它的同鍵範圍值下的數值都是在加按聲調鍵後的統計。由表 4 可知，透過聲調鍵的篩選，在輸入二字詞時可有 82.1%的機會不用注視螢幕，並且最大的同鍵範圍值也由 14 降到 6 了，即減低了同鍵詞問題之複雜度；此外由表 5 也可看到類似的改進，透過聲調鍵的篩選後，輸入三字詞時可有 85.7%的機會不用注視螢幕，並且最大的同鍵範圍值會由 9 降到 5 。

表 4　加聲調鍵之二字詞同鍵詞分析

同鍵範圍	1	2	3	4	5	6
詞個數	27,329	4,882	804	188	65	6
比率%	82.1	14.7	2.4	0.6	0.2	0.0

表 5　加聲調鍵之三字詞同鍵詞分析

同鍵範圍	1	2	3	4	5
詞個數	6,924	982	135	24	5
比率%	85.7	12.1	1.7	0.3	0.1

5. 原型系統製作

爲了驗證本文提出的多字詞輸入之按鍵規則的可行性，我們遂實際去製作一個可以詞爲單位來輸入中文的原型軟體系統，由於這個系統也必需能夠接受字爲單位的注音輸入，因此我們決定拿過去建立的字爲單位來輸入、且以馬可夫語言模型來進行連續注音轉換之中文輸入系統[14]來作基礎，然後加以擴充使它能夠處理多字詞之按鍵輸入。

由第 3 節裡的按鍵規則可知，多字詞的輸入按鍵只提供了音節注音的部分資訊，不像單字的輸入按鍵提供了整個音節的注音資訊，因此，到詞典去查多字詞輸入按鍵可能對應的詞語時，注音資料的比對就必需改變成一種非精確比對的方式，實作上我們以空白符號(即 ASCII 碼 32)來填充缺少的注音資料，然後令空白符號可和任何注音符號比對成功。此外，原型系統裡的詞典是分成靜態與動態兩種詞典的，靜態詞典裡的詞語長度爲 1 至 4 ，所以只能支援按鍵規則(R2)、(R3)、(R4)，用以查詢一至四字詞的注音輸入會對應到的詞語，並且不能用以查詢新增的、未登錄過的詞語；相反地，動態詞典支援的按鍵規則除了(R2)、(R3)、(R4)之外，還增加了(R5)，即接受查詢的詞語長度可由 2 變化到 13 ，並且可用以查出先前曾輸入過的專有名詞、慣用語(在不需主動登錄的條件下)。這樣的靜、動態詞典的功能差異，意味著在原型系統裡製作這兩種詞典時，需要採取不同的資料結構、不同的處理方式。

關於動態詞典的一個重要的、尚未說明的功用是，支援短詞組合成長詞的處理。例如，使用者曾各別輸入"電腦"與"文盲"兩詞語來串成"電腦文盲"之詞，然後按*(Enter)*鍵以送出該詞並將它存入到動態詞典裡，則下次使用者依按鍵規則(R4)來按 ㄉㄋㄨㄇ 等四鍵時，輸入軟體要能夠從動態詞典中找出"電腦文盲"之詞。依據第二節裡的動態詞典的結構設計，我們達成短詞組合成長詞的作法是，先用前三字的部分注音(如前述的ㄉㄋㄨ)到第二個赫序表去作非精確比對，以查出可能和它對應的三字詞 $U_nV_nW_n$, n = 1, 2,...,N ，再用後三字的部分注音(如前述的ㄋㄨㄇ)去作非精確比對，以查出可能對應的三字詞 $X_kY_kZ_k$, k=1,2, ...,K ，然後對所有可能的 n, k 組合下的 V_nW_n 與 X_kY_k 作比對，以找出當 V_nW_n 相同於 X_kY_k 時所組合出的四字詞 $U_nV_nW_nZ_k$ 。

此外，若使用者要依按鍵規則(R5)來輸入六字詞的注音按鍵 $P_1P_2P_3$ $P_4P_5P_6$ ，則

當他輸入到 P_4 時，就先以 $P_1P_2P_3P_4$ 去第二個赫序表查出可能對應的四字詞 $A_n B_n C_n D_n$ ，而當他輸入到 P_5 時，再以 $P_3P_4P_5$ 去查出可能對應的三字詞 $E_kF_kG_k$ ，然後對所有可能的 n, k 組合下的 C_nD_n 與 E_kF_k 作比對，以找出當兩者相同時所組合出的五字詞 $A_n B_n C_n D_n G_k$ ，令找出的五字詞重新排定下標後以 $A'_j B'_j C'_j D'_j G'_j$, j=1,2,...,J, 表示；接著當輸入 P_6 後，就以 $P_4P_5P_6$ 去查詢可能對應的三字詞 $Q_iR_iS_i$ ，然後對所有可能的 j, i 組合下的 $D'_j G'_j$ 與 Q_iR_i 作比對，以找出當兩者相同時所組合出的六字詞 $A'_j B'_j C'_j D'_j G'_j S_i$ 。對於其它字數(大於六字)的多字詞按鍵，只需將前述的作法延續下去即可。不過，檢視前述的作法可看出，此種組合成長詞的作法有時會組合出未曾輸入過的詞語，例如使用者曾輸入過"小獅子"與"獅子頭"之詞，則依按鍵規則(R4)按ㄒㄕㄗㄊ等四鍵時，會找出未曾輸入過的"小獅子頭"之詞，然而這種情形在很多時候都是可接受的。

6. 結語

我們認爲在近期內，鍵盤仍將是一種普遍被使用的中文輸入工具，而本文倡議的加快鍵盤輸入中文之方法，包含：(1)詞爲單位之輸入方式、(2)多字詞按鍵規則、(3)支援短詞組合成長詞並可動態記錄新詞之動態詞典等功能，如果能夠推廣至原已存在的中文輸入法上，深信可帶給爲數眾多的中文輸入者一定程度的助益。

爲了實踐以詞爲單位來輸入的觀念，本文研究了兩個重要的課題：動態詞典結構設計與多字詞按鍵規則設計。在動態詞典方面，由於動態詞典不僅要支援以詞爲單位來輸入之方式，也要支援以字爲單位來輸入時的連續轉換處理，因此我們提出一種包含兩個赫序表和一個貯列的複合結構設計，以滿足二者之快速查尋的需求；在多字詞按鍵規則方面，我們以注音輸入法爲例，設計了一組簡單易記、且相容的多字詞按鍵規則，而同樣的設計原則可被用來爲其它的中文輸入法設計多字詞按鍵規則。依據所設計的按鍵規則去對靜態詞典裡的二、三、四字詞作同鍵詞分析，我們發現各有 42.6%, 42.3%, 90.7% 的詞語沒有同鍵詞，而如果再以聲調來篩選二、三字詞，則可各別讓高達 82.1%與 85.7% 的詞語沒有同鍵詞，而減少了需要注視螢幕的機會。此外，我們也實際地去製作出一個整合了單字注音連續轉換、及多字詞直接輸入的原型系統，而驗證了前述方法的可行

性。

參考文獻

Bell, T. C., J. G. Cleary and I. H. Witten, Text Compression, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1990.

Chen, S. I., et al., "The Continuous Conversion Algorithm of Chinese Character's Phonetic Symbols to Chinese Character", Proceedings of National Computer Symposium (Taipei), pp. 437-442, 1987.

Chuang, C. T. and L. Y. Tseng, "A Heuristic Algorithm for the Recognition of Printed Chinese Characters", IEEE trans. on Systems, Man, and Cybernetics, Vol. 25, No. 4, pp. 710-718, April 1995.

Fan, C. K. and W. H. Tsai, "Reduction of Key Stroke Numbers in Chinese Input by Relaxation Based Word Identification", Proc. of Int. Conf. on Computer Processing of Chinese and Oriental Languages(Taipei), pp. 37-44, 1991.

Gu, H. Y., C. Y. Tseng and L. S. Lee, "Markov Modeling of Mandarin Chinese for Decoding the Phonetic Sequence into Chinese Characters", Computer Speech and Language, Vol. 5, No. 4, pp. 363-377, 1991.

Ho, T. H., el al., "Integrating Long-Distance Language Modeling to Phoneme-to-text Convertion", Proc. of Int. Conf. on Computer Processing of Oriental Languages (Taipei), pp. 287-299, 1997.

Hsu , W. L., "Chinese Parsing in a Phoneme-to-Character Conversion System Based on Semantic Pattern Matching", Computer Processing of Chinese and Oriental Languages, Vol. 8, No. 2, pp. 227-236, Dec. 1994.

Huang, K. Y., H. T. Yen and C. S. Han, "Neural Networks for Robust Recognition of Printed Chinese Characters", Computer Processing of Oriental Languages, Vol. 10, No. 4, pp. 425-442, April 1997.

Kuo, J. J., "Phonetic-input-to-character Conversion System for Chinese Using Syntactic Connection Table and Semantic Distance", Computer Processing of Chinese and Oriental Languages, Vol. 10, No. 2, pp. 195-210, Oct. 1995.

Lin, M. Y. and W. H. Tsai, "Removing the Ambiguity of Phonetic Chinese Input by the Relaxation Technique", Computer Processing of Chinese and Oriental Languages, pp. 1-24, 1987.

Wang, H. M., et al., "Complete Recognition of Continuous Mandarin Speech for Chinese Language with Very Large Vocabulary Using Limited Training Data", IEEE trans. Speech and Audio Processing, Vol. 5, No. 2, pp. 195-200, March 1997.

中研院中文詞知識庫小組，中文書面語頻率詞典， 1994 。

古鴻炎，"一個同時考慮鍵盤效率、人體工學原則、及符鍵對應規律性之國語注音輸入鍵盤的設計"，電工雙月刊，第 35 卷，第 2 期，第 123-132 頁， 1992 。

古鴻炎、陳志耀，"使用新式注音鍵盤及複合馬可夫語言模型之中文輸入系統"，中華民國電腦學會電腦學刊，第 7 卷，第 3 期，第 1-9 頁， 1995 年。

古鴻炎，"動態詞典及其與馬可夫中文語言模型之整合"，全國計算機會議論文集(台中)，第 D1-D7 頁， 1997 。

蔡奇峰、馬自恆，"樣本比對之中文手寫辨識系統"，全國計算機會議論文集(台中)，第 B39-B44 頁， 1997 。

Quantitative Criteria for Computational Chinese Lexicography

A Study Based on a Standard Reference Lexicon for Chinese NLP

Topic Areas: (d) electronic dictionaries, (h) large corpora

Chu-Ren Huang, Zhoa-ming Gao, Claude C.C. Shen, and Keh-jian Chen

Academia Sinica

Email:hschuren@ccvax.sinica.edu.tw

Fax:(02)2788-1638

Abstract

The construction of a standard reference lexicon for Chinese NLP involves two fundamental issues in computational linguistics: the definition of a word and the principled delimitation of the lexicon. We argued that such reference lexicons must be judged by their cross-domain portability, expressive adequacy, and reusability. Thus principles for lexical selection must also be driven these criteria.

This paper reports the approach and result of our construction of a standard reference lexicon for Chinese NLP, which also serves as the empirical basis for a segmentation standard. Our approach uses a mixture if stochastic and heuristic steps. First, a reference corpus is selected and lexical entries are automatically extracted from it based on statistically significant threshold. Second, the coverage of the automatically extracted lexicon is enhanced by conceptual primes as well as by comparative studies of MRD's from different Chinese speaking communities. We show the satisfactory coverage of the resultant lexicon by testing it with randomly accessed texts from the web.

1 Introduction

Since words are not conventionally marked in Chinese texts, segmentation is a pre-requisite step for Chinese NLP and setting a standard to define and measure segmentation results becomes necessary both for evaluation and for resource-sharing. However, as noted by standard-setters both in Mainland China (Liu et al. 1994) and in Taiwan (Huang et al. 1996), no segmentation standard can be successfully implemented and evaluated until it is accompanied by a wide-coverage reference lexicon. Huang et al. (1996) argued that segmentation standards must include a sharable and adaptable lexicon in order to apply across variations such as domain, genre, and time. On the other hand, for data-sharing, a standard lexicon is essential to ensure that texts from different sources can be uniformly tokenized. Thus, there is

a consensus that an empirically compiled reference lexicon is an indispensable part of a Chinese segmentation standard (e.g. Lin and Miao 1997, Sun and Zhang 1997). An additional benefit of such a lexicon is that it can be shared without much additional cost and thus saves NLP researchers the time and cost in building this essential infrastructure. To serve the above dual purposes, this standard lexicon must be selected in a principled way in order to best test validity and usefulness.

However, even though computational lexicography offers a rich literature on the structure and content of a lexical entry, there is hardly any discussion of a principled way of lexical entry selection (Armstrong-Warwick 1995). We only see discussion in the content of a terminology lexicon (Nagao 1994) or a reference segmentation lexicon (Liu et al. 1992). Both assume that the lexicon is built from scratch. We suggest that there are three criteria to judge the merit of a lexicon: reusability, expressive adequacy, and domain-portability. Based on these three criteria, we propose a principled way to construct a standard reference lexicon for Chinese NLP.

2 Criteria for Selection of Lexical Entries

2.1 Word, Segmentation Unit and Lexical Entry

Determining whether a string is a (new) word is trivial in many languages such as English. However, it is not easy in Chinese because of the lack of conventional demarcation and of native speakers' consensus of what is a word. Once a string is identified as a unit, a further decision needs to be made as to whether it should be listed in the lexicon (e.g. Wang et al. 1994).

In this study, we stipulate that all entries must be segmentation units defined in

Huang et al. (1997a & b). Notice that even though Huang et al. Propose to take the notion of linguistic word as the theoretical foundation of the definition of segmentation unit, it is obvious that certain non-words, such as (derivational) affixes, must also be treated as segmentation units. Thus they must be listed in the reference lexicon for segmentation.

The motivation of such a stipulate is two fold. First, it ensures uniformity of the segmentation criteria and the reference database within the segmentation standard. Second, this allows the reference lexicon to list non-words such as derivational affixes, and thus will provide crucial information to account for the productive morpho-lexical processes.

2.2 Reusability: Corpus Base of Lexical Selection

That the corpus is the best source of lexical entries has been the cornerstone of recent developments of corpus linguistics (e.g. Sinclair 1987). Making a balanced corpus as the basis of a standard reference lexicon also makes it possible to automatically update the lexicon for different domain or for language changes. Either a monitor corpus will be maintained to indicate any change in the language, or comparable corpus from separate domains can be maintained, any new entries can be extracted by the same automatic procedure to augment and revise the standard set. Our current lexicon is based on the Sinica Corpus (Chen et al. 1996), a tagged balanced corpus of Taiwan Mandarin Chinese containing 5 million words. 146,876 different words appear in the corpus. The number of lexical entries defined by frequency threshold of 1 to 10 are as follows:

(1) Number of Lexical Entries as Defined by Sinica Corpus Frequency

frequency threshold	**number of word types**
1	146,876
2	*84,309*
3	*63,421*
4	*52,571*
5	*45,443*
6	*40,395*
7	*36,392*
8	*33,301*
9	*30,701*
10	*28,564*

Our data shows that the frequency thresholds of 10 and 4 correspond to sharp increase in the number of lexical entries. Thus these are the two thresholds that will be adopted later in this work.

2.3 Expressive Adequacy: Conceptual Primes and Lexical Selection

Selecting lexical entries by frequency threshold based on corpus calculation is a dependable way to ensure relatively high coverage of the lexicon. However, since lexical information is nit available in NLP unless it is encoded in the lexicon, high coverage does not necessarily translate into successful application if conceptually crucial items are missing. Thus, we propose that a standard reference lexicon must achieve expressive adequacy. Our hypothesis is that such adequacy can be ensured when entries representing conceptual primes are exhaustively included.

The conceptual primes that we adopt are the 3,922 covering terms of Tongyici Cilin (Mei et al. 1983, CILIN hereafter), the most widely used thesaurus in Chinese NLP. We treat them as if they are covering terms in semantic fields, assuming these terms alone will be adequate to express concepts represented by embedded terms in their fields. Thus a lexicon containing all these terms will be expressively adequate.

A possible objective to adopting such an heuristic method independent of corpus – based stochastic approaches is that the same goal could be achieved without the heuristic. In the other words, is there any evidence to prove that these conceptual primes cannot satisfactorily extracted from corpora.

Diagram 1 shows the frequency/rank correlation of the CILIN conceptual primes based on their occurrences in Sinica Corpus. If conceptual primes were to be reliably extracted from corpora, they must fall (almost) exclusively in mid to high frequency rank. However, diagram 1 follows Zipf's law. In the other words, these conceptual primes areas widely distributed as other lexemes. Any corpus-based frequency threshold will unfortunately exclude the lower frequency conceptual primes.

One possible explanation for the Zipf's Law like distribution of the semantic primes is that a complete conceptual system needs to express all concepts regardless of their frequencies. The less frequently used semantic primes are those involving surprises or rarity, both are fundamental concepts. In the other words, a complete set of semantic primes necessarily contain less frequent words and their distribution should reflect the distribution should reflect the distribution of all meaning expressible by the language (i.e. the lexicon).

In fact, only 3,501 of the CILIN covering terms occur in Sinica Corpus, meaning that 421 terms are missing. These missing terms cannot be attributed solely to the lexical difference between Mainland China and Taiwan. Two authoritative dictionaries that consulted corpus extensively also do not enter all the CILIN (primes. The 57,624 entry Xiandaihanyu Cidian (XHCD hereafter) lacks 241 of them while the 39,025 entry Segmentation Standard Lexical List (Liu et al. 1994, GB hereafter) misses 546 of them. There does not seem to be a correlation between the degree of human intervention with the completeness of conceptual primes though. XHCD is

compiled by linguists who consulted corpora, while GB is extracted from a corpus and augmented with thought-up lexical items.

In diagram 2, an addition test is conducted on the distribution of these conceptual primes. The diagram shows the number of conceptual primes every 1,000 words in Sinica Corpus ordered according to frequency rank. As suspected, a high proportion of the most frequent words are conceptual primes (382 of the first 1,000), while the proportion descends dramatically. The diagram shows the slope smoothes at around rank 1,500 and levels well before rank 10,000.

Two important pieces of information can be inferred from diagram 2. First, it offers an intuitive support of the reliability of the CILIN primes. Since conceptual primes are the most economic (and often necessary) way to express ideas, they are more likely to be frequently used. Thus, we expect a valid set conceptual primes to be dominated by high frequency words. The CILIN distribution confirms such prediction.

Second, the steep descend and quick leveling suggests that it will be impractical to discover conceptual primes with pure stochastic approach. Since these conceptually primary terms are sparsely distributed in mid to lower frequency range, it would be quite impossible to achieve any reasonable recall and precision at the same time. In other words, for the moment at least, conceptual primes must be acquired independent of a corpus.

2.4 Portability: Bootstrapping with Existing Lexicons

It is impossible for a corpus, with finite total words, to cover all possible topics, genre etc. Hence it is most likely that some significant lexemes are not represented in a corpus. In other words, how can a standard lexicon be portable among all domains given the fact the corpus it based on does not contain texts from all possible domain?

This problem could be aggravated if a corpus is relatively small and geographically restricted.

The case is even worse for a Chinese lexicon because of the fact that there exist substantial lexical differences between Mainland and Taiwan Mandarin. Thus it would be futile to construct a corpus that could represent both dialects. However, it is also well-known that mutual lexical borrowings are easy and frequent contacts. Thus any purely Taiwan or Mainland corpus faces the dilemma of under-representing a critical segment of lexemes.

To solve this dilemma, we propose to bootstrap with lexicons. We consult the entries of five lexicons, including two each from PRC and Taiwan, as well as one from the U.S. The two Mainland lexicons: List of Frequently Used Modern Mandarin Words for information Processing (Appeared in Liu et al. 1994, referred to as **GB** hereafter), and Xiandai Hanyu Cidian (Chinese Academy of Social Sciences 1996, referred to as **XH** hereafter). The two Taiwan lexicons are: the Chinese Knowledge Information Processing Electronic Lexicon of Academia Sinica (last updated 1996, referred to as **CKIP** hereafter), and the on-line version Revised Revision of Mandarin Chinese Dictionary by the Council on Mandarin Chinese of the Ministry of Education (1997 version, referred to as **RMCD** hereafter). Lastly, the **ABC** Chinese-English Dictionary (DeFrancis 1997, referred to as ABC hereafter) not only represents the perspective of a language learner but also offers a perspective not dictated solely by linguistics experience in one single area.

(2) Number of Lexical Entries in the Five Dictionaries

Dictionaries (year of compilation)		**Number of Entries**
CKIP	1996	78,323
RMCD	1997	156,710
XH	1996	56,162
GB	1993	39,459
ABC	1997	70,325

Our claim is that comparing entries from compiled lexicons allows us to tap existing knowledge and labor-intensive resources. The decision to include a lexical entry in a lexicon reflects the collective knowledge of (at least a good number of) native speakers and is at least as valuable as un-processed raw corpus data.

2.4.1 Towards a Formal Definition of Lexicon Similarity

In this section, we will try to set a principled way of comparison of lexicon as well as to interpret the important of repeated occurrence of an entry in different lexicons. As shown by (2), the sizes of the five lexicons vary greatly, from just under 40 thousand entries to over 156 thousand entries. Since these lexicons are describing the same language, they should in principle have very similar entries. Thus the two questions that one must ask ate 1) roughly speaking, are the smaller lexicons subsets of the larger lexicons? 2) are the lexicons compiled in the same geographic area more similar to each other? To answer the two questions, we start by finding out the coverage rate of each dictionary pairs. The coverage of dictionary A over dictionary B is defined as

(3) Coverage of Dictionary A over Dictionary *B*

$Cov_{A/B} \overset{def}{=}$ Number of entries in the intersection of A and B/Total number of entries in B

Based on the above definition, the coverage rate among the five dictionaries are calculated as below:

(4) Coverage Among the Five Dictionaries

B \ A	CKIP	RMCD	XH	GB	ABC
CKIP	100%	68.89%	45.85%	35.94%	50.34%
RMCD	34.42%	100%	29.42%	19.72%	30.76%

XH	63.94%	82.10%	100%	50.92%	75.36%
GB	71.33%	78.30%	72.48%	100%	79.44%
ABC	56.07%	68.58%	60.18%	44.58%	100%

Take note that the above definition of coverage is dependent on the size of the lexicon. That is, mathematics speaking, a similar lexicon cannot have a good coverage of a bigger lexicon since it cannot cover of a larger lexicon over a smaller one is not especially high. For instance, although **RMCD** is almost four times as big as **GB**, it only covers 78.30% of the later. This and the wide range coverage numbers suggests that we need a better criterion for dictionary similarity. We cannot ignore the fact that number of entries is a very important feature of any lexicon. However, to make sure that extreme difference in sizes do not skew the similarity between lexicons, we propose that mutual coverage as a good measure of lexicon similarity.

(5) Mutual Coverage of Two Lexicons A and B

$$\mathbf{Mcov_{A,B}} \overset{def}{=} \mathbf{Cov_{A/B} + Cov_{B/A}/2}$$

Based on the above definition, mutual coverage among the five lexicons are given below from the highest mutual coverage rate to the lowest.

(6) Mutual Coverage among Five Lexicons (descending order)

$\mathbf{MCov_{ABC,XH}}$	67.77%
$\mathbf{MCov_{ABC,GB}}$	62.07%
$\mathbf{MCov_{XH,GB}}$	61.70%
$\mathbf{MCov_{RMCD,XH}}$	55.76%
$\mathbf{MCov_{CKIP,XH}}$	54.90%
$\mathbf{MCov_{CKIP,GB}}$	53.64%
$\mathbf{MCov_{CKIP,ABC}}$	53.21%

$\mathbf{MCov}_{\mathbf{RMCD,CKIP}}$ 61.65%

$\mathbf{MCov}_{\mathbf{RMCD,ABC}}$ 49.67%

$\mathbf{MCov}_{\mathbf{RMCD,GB}}$ 49.01%

The above result confirmed our suspicion that **ABC**, **XH**, and **GB** are more similar to one another. This is because these follow the PRC usages predominantly, including ABC, although it is compiled in the States. However, **RMCD** and **CKIP** do not show the same degree of similarity. As a matter of fact, all the other three lexicons are more similar to **CKIP** than **RMCD** according to this measure. In other words, it is more than simply geo-political influence that determines the similarity of the lexicons. The criteria of lexical selection as well as the topic areas covered will play a crucial role too. **RMCD** has a selection criterion that is quite different from the other lexicons, that is it tries to be exhaustive without being sensitive whether an entry is commonly used by the speaking community. This may contribute to the reason why it appears to be the most different from the other four lexicons in our calculation. Another way to check the similarities of these five lexicons is to find out how many entries are shared by them. We found that all together there are 206,802 different word types (i.e. entries) recorded, and among them only 21,655 entries are entered in all five lexicons.

(7) Number of shared entries

a. shared by all five lexicons 21,655

b. shared by (at least) four lexicons 35,924

c. shared by (at least) three lexicons 54,111

d. shared by (at least) two lexicons 82,332

We believe the above data points to a definition of a standard core lexicon that is used most by most Chinese in most contexts. As we see, any two lexicons are only 50% to 60% similar. We further see that the number of entries that all lexicon

compilers agree upon is only 21,655. This is only a small fraction of all number of entries in each lexicon.

2.4.2 Why Lexicons Differ: the emergence of a core lexicon

The above study of different lexicons as well as earlier computational lexicography studies based on corpus suggest that lexicographers as well as corpora are biased. That the core lexicon entries tend to be covered by different corpora and different lexicographers. But there will be a lot of disagreement among corpora as well as lexicographers when more peripheral entries are being chosen. Thus, we can see a core lexicon emerging when we compare different authoritative lexicons as well as consult reliable large corpus. In the next section, we will propose a principled way to construct standard lexicons based both on dictionary and corpus knowledge so that the bias of each methodology can be canceled and valuable information from each approach can be utilized.

3 Principle and Methodology Towards a Standard Reference Lexicon

To meet the criteria of reusability, expressive adequacy, and cross-domain portability, we combine a three step algorithm for constructing a standard reference lexicon for Chinese NLP. First, lexical entries are automatically extracted from a balanced tagged corpus if their frequencies are higher than a stochastically determined threshold. The corpus-based generation allows automatic updating and adaptation to specific domains. Second, the automatically generated lexicon is augmented with a small set of conceptual primes to ensure expressive adequacy. Last, it is further augmented with entries obtained form intersection of 5 lexicons form different

sources to ensure cross-domain portability.

First, we define three levels of standard lexicons. The **Core Lexicon** is the most stable part of the language. It will be used regardless of geographic area, topic, media, style, genre, etc. In other words, it is the core of the segmentation standard that will be portable through different uses and through a reasonable duration of time. Second, the **General Lexicon** is a superset of the core lexicon. The extension over the core lexicon allows it to give better comprehensive coverage of text in general domains (such as newspapers or general textbooks). Last, the **Reference Lexicon** is an open set that is also the superset of the general lexicon. We want to included all lexical entries that are arrested words currently being used in the language (and are also segmentation units) to be listed in the reference lexicon. Ideally, the reference lexicon will have attribute attached so that special sub-lexicons can be automatically extracted for the special uses. But such annotation and expansion of the reference lexicon will involve voluntary cooperation of users from all different backgrounds. Right now, we envision the reference lexicon as an open set maintained virtually by R.O.C. Computational Linguistics Society. Any new lexical items nit covered by the current version of the reference lexicon will be reported on-lined. A team of experts will double-check that the reported new entries meet the required criterion of being a segmentation unit, and admit the entry to the reference lexicon.

On the other, the core and general lexicons will be maintained and updated periodically, perhaps every 3 to 5 years. The update will be based on corpus data as well as revisions on the dictionaries consulted. The update will allow the two lexicons to keep with linguistics changes, which is most evident in the area of lexicon.

3.1 Extraction of the Standard Lexicon: a hybrid approach

Our current algorithm for extracting the three levels of standard lexicons are:

(8) Core Lexicon

Entries must be listed in all five lexicons (**ABC**, **CKIP**, **GB**, **RMCD**, and **XH**), as well as occur for at least 10 times in the Sinica Corpus.

(9) General Lexicon

Entries must be listed in at least three of the five lexicons (**ABC**, **CKIP**, **GB**, **RMCD**, and **XH**), as well as occur for at least 4 times in the Sinica Corpus.

(10) Reference Lexicon: Entries must either

a. be listed in at least three of the five lexicons (**ABC**, **CKIP**, **GB**, **RMCD**, and **XH**); or

b. be listed in at least one of the five lexicons (**ABC**, **CKIP**, **GB**, **RMCD**, and **XH**) and occur at least once in the Sinica Corpus; or

c. be listed as one of the semantic primes in *Tongyici Cilin*.

Please note that the heuristic for the reference lexicon above attempts to extract the largest list possible of legitimate entries without human intervention. The three disjunction conditions are three different ways to make sure that an entries is indeed a lexical entry and segmentation unit in the language and not just a careless mistake of a lexicographer or an accidental error in a corpus. As mentioned above, it will then require continuing human intervention in the future to maintain the growth of the reference lexicon. The number of entries thus collected are listed in (11).

(11) Number of Entries of

a. **Core Lexicon: 13,049**[1]

b. **General Lexicon: 26,443**

c. **Reference Lexicon: 81,787**

4 Verification and Expendability

To verify that our standard reference does not meet the requirements set out by the three criteria, we will do both internal and external tests. Tests are performed with an automatic segmentation procedure to determine coverage of the lexicon of all words appearing in their language. Internal tests will be performed in texts extracted from Sinica Corpus, which are marked with topic, genre, style, media etc. Our aim will be to ensure that consistently high coverage is achieved across all possible variations. External tests will be performed with texts not included in Sinica Corpus, especially texts from Mainland China as well as texts extracted from WWW.

4.1 Verification of the Versatility of the Core Lexicon

We have mentioned above that the most important attributed of the core lexicon is its versatility, i.e. that it will be least sensitive to the change of texts and will still offer the same coverage. To test this requirement, we use all the texts in Sinica Corpus to as internal test set. As described in Chen et al. (1996), the over 500 texts in the Sinica Corpus are given textual mark-up in five different dimensions: Spoken/Written, Topic, Media, Genre, and Style. In each dimension, there are further divisions. For instance, Topic attributed included: Philosophy, Psychology, Chemistry, Society Culture, International Relationship etc. And Media attributed included Newspapers, Academic Journals, Audio-Visual etc. Thus we will be able to check the coverage of the core lexicon with regard to the dimensions of variations. The baseline lexicon

[1] The number of Core Lexicon is comparable to the theory of "詞涯七千" (Cheng, 1998).

we use to compare in this case is the 13,049 most frequent words in the Sinica corpus. In other words that are known to have the highest coverage of the collective texts. Thus the fact that the core lexicons has more stable coverage than this set of words will be one of the strongest possible evidence to show the versatility of the core lexicon. First, we adapt the definition of coverage given in (3) define the lexical coverage of a text.

(12) Lexical Coverage of a Text by a Lexicon L

$\text{LexCov}_L \overset{def}{=}$ Number of L's entries that appear in the text/Total number of word types in the text

Sine the baseline set contains the most frequent words of the corpus, it is mathematically impossible for the core lexicon to have higher coverage. So what we need to show crucially is that core lexicon will have a more stable coverage regardless of the nature of texts, given that its coverage is not too much lower than the most frequent word list. The statistical method we choose is the standard deviation of the coverage among all texts.

(13) Core Lexicon vs. Most Frequent Words

		Lexical Coverage	Standard Deviation
a) Spoken	Core	62.728%	6.05422%
	HiFre	76.7088%	6.69072%
b) Written	Core	57.434%	6.63843%
	HiFre	69.1228%	8.40772%
c) Topic	Core	53.1445%	2.47149%
	HiFre	64.5383%	2.71369%
d) Media	Core	58.1081%	3.39812%

	HiFre	69.6637%	3.93821%
e) Genre	Core	58.2538%	3.26635%
	HiFre	69.2185%	3.68371%
f) Style	Core	56.8369%	0.958196%
	HiFre	68.0522%	0.796978%

Take note that the above average is calculated based on the parameters within each dimension. For instance, the average and standard deviation under Topic is calculated based on the average coverage of the 56 topic divisions. The coverage of each topic division is in term calculated based on the coverage of all the texts assigned to that topic division. Thus what the test shows us is the performance of the core lexicon when confronted with variations in 5 different dimensions. The result is very reassuring in that although the lexical coverage of the core lexicon is slightly lower than the most frequent word list, as expected; its standard deviation is almost always lower than that of the most frequent words. And in the four dimensions where the core lexicon has a lower standard deviation, the difference is statically significant. The only case where the most frequent words have a lower standard deviation is in the Style dimension. However, in this case both standard deviation are very low and the difference even lower (only about 0.16%). This actually suggest that lexical coverage does not differ when the style (e.g. descriptive vs. expository etc.) changes.

In addition to internal tests on texts in the Sinica Corpus, we also did external tests with texts extracted from WWW. Since the Sinica Corpus is based in Taiwan, we tried to extract texts from the PRC. One caution with the external test is that the texts are automatically segmented, and were not manually checked like the Sinica Corpus. Thus the segmentation result may not only be 90%-95% correct. The test size is about 100,000 words.

(14) Lexical Coverage: external test

	Lexical Coverage	Standard Deviation
Core	59.1798%	3.68189%
HiFre	64.7171%	4.34954%

As expected, the standard deviation of lexical coverage by the core lexical is still significantly lower than that of the most frequent words from the Sinica Corpus. What is also reassuring is that the lexical coverage remain reliable at around 60% for the external texts. Since these are more frequent words, the textual coverage (i.e. coverage of tokens) is actually around 80%.

To sum up, both the internal and external tests attested to the versatility and stability of the proposed core lexicon. We expect this result to be applicable to future uses. The core lexicon should prove to be stable regardless of all sorts of textual variations.

4.2 Verification of the Applicability of the General Lexicon

As mentioned above, the general lexicon is constructed such that it will have comprehensive coverage of general texts not in a special domain. Thus, its goal is similar to that of the GB lexicon. Although, there are only 26,443 entries in our general lexicon, only 2/3 of the size of the GB lexicon (39,459 entries). However, our test will show that the disadvantage in size does not prevent the general lexicon form out-performing out-performing the GB lexicon.

(15) Textual Coverage of a Text by a Lexicon L

$\text{TextCov}_L \overset{def}{=}$ Number of tokens in the texts that are also L'2 entries /Total token number in the text

According to the above definition we can calculated the average textual coverage of all Sinica Corpus texts by our general lexicon to be 86.7619%; while the average textual coverage of the mush larger **GB** lexicon is only 83.3796%. The standard deviation of the coverage by the general lexicon is also almost 1% lower than that of the **GB** lexicon (3.9655% vs. 4.82408%). The lexicon coverage test also show similar results. In sum, we have attested that the general lexicon serves its purpose and our hybrid approach constructs a lexicon that out-performs one that is mainly corpus-based.

5. Conclusion

In this paper, we have proposed an approach to construct standard reference lexicons for NLP. This approach crucially depends on both corpora and lexical knowledge represented in human-compiled lexicons. In the process, we have also proposed formal principles to measure similarities between lexicons, as well as measures of coverage of a text by a lexicon. We use these formal measures to obtain data in support of our approach. We have also proposed a three level structure of standard lexicon, where the **Core Lexicon** will be the most versatile and most portable; the **General Lexicon** is less portable will be efficient and give comprehensive coverage for general applications; last, the reference lexicon is the open set reference that will contain as many words in the language as possible and will ideally allow users to extract their own special domain lexicons from; as well as to contribute their special domain entries to[2]. It is our hope that this first step towards a formal study of lexical selection principles as well as measurements for lexical coverage will point to a fertile

[2] The Lexicons are available under the following website:
http://rocling.iis.sinica.edu.tw

ground in computational lexicography, in addition to fulfilling its original goal of offering reliable data support for Chinese segmentation standard.

Acknowledgement: We would like to thank the helpful comments of professor C. C. Cheng and ROCLING reviewers, as well as CKIP colleques' help in preparing the data. Responsibilities for any remaining error is of course ours alone.

Reference

Armstrong-Warwick, S. 1995. Automated Lexical Resources in Europe: A Survey. In D.E. Walker, A. Zampolli, and N. Calzolari Eds. Automating the Lexicon. 397-403. Oxford: Oxford U. Press.

Chen, K.-j., C.-R. Huang, L.-P. Chang, and H.-L Hsu. 1996. SINICA CORPUS: Design Methodology for Balanced Corpora. In B.-S. Park and J.-B. Kim Eds. Language, Information, and Computation. Selected Papers from the 11th PACLIC. Seoul: Kynung Hee U.

Cheng, C. C. 1998. 從歷代經史子集研究人對語言詞彙的認知. 發表於人文計算研討會, Taipei, Academia Sinica..

Chinese Academy of Social Sciences. 1996. Xiandaihanyu Cidian [A Dictionary of Contemporary Chinese (Revised Edition)]. Beijing: Shangwu.

Chinese Knowledge Information Processing Group. 1996. ShouWen JieZi – A Study of Chinese Word Boundaries and Segmentation Standard for Information Processing [In Chinese]. CKIP Technical Report 96-01. Taipei: Academia Sinica.

---, 1995. The Grammatical Categories of Mandarin Chinese.[in Chinese] CKIP Technical Report 95-03. Taipei: Academia Sinica.

Huang, Chu-Ren, Keh-Jiann Chen, Feng-yi Chen, Wen-Jen Wei, and Lili Chang.

1997. The Design Criteria and Content of the Segmentation Standard for Chinese Information Processing [in Chinese]. Yuyan Wenzi Yingyong. 1997.1.92-100.

---, Keh-Jian Chen, Lili Chang and Feng-yi Chen. 1997. Segmentation Standard for Chinese Natural Language Processing. Computational Linguistics & Chinese Language Processing. 2.2.46-62.

---, Zhao-ming Gao, Claude C.C Shen, and Keh-jiann Chen. 1998. Towards a Sharable and Reusable Lexical List: The construction of a standard reference lexicon for Chinese NLP. Presented at the 1998 Pacific Neighborhood Consortion (PNC) Annual Meeting. To appear in the Proceedings. Taipei: Academia Sinica.

Lin, X.G., and C.J. Miao. 1997. Guifan+Cibiao yu Jinyen+Tongji. Yuyan Wenzi Yingyong. 1997.1.7-91.

Liu, Y., Q. Tan, and X. Shen. 1994. Segmentation Standard for Modern Chinese Information Processing and Automatic Segmentation Methodology.[in Chinese] Beijing: Qinghua U. Press.

Liu, Y., N. Liang, and Q. Tan. 1991. Lexical Selection Criteria for 'A Lexicon of Frequent Modern Mandarin Words for Information Processing'. Proceedings of the Tenth Anniversary of Chinese Information Society of China. 127-141.

Mei, J., Y. Zhu, Y. Gao, and H. Yin. 1983. Tongyici Cilin. Shanghai: Shangwu Press and Shanghai Dictionaries.

Nagao, M. 1994. A Methodology for the Construction of a Terminology Dictionary. In B.T.S. Atkins and A. Zampolli Eds. Computational Approaches to the Lexicon. 379-412. Oxford U. Press.

Sinclair, J. M. 1987. Ed. Looking Up-An account of the COBUILD Project in Lexical Computing. London: Collins. Sproat, R. 1992. Morphology and Computation.

Cambridge: MIT Press.

Sun, M.S., and L. Zhang. 1997. Renjibingcun, Zhiliangheyi –tantan Zhidinh xinxi chuliyong hanyu cibiao de celue. Yuyan Wenzi Yingyong. 1997.1.79-86.

Wang, M.-C., C.-R. Huang, and K.-j. Chen. 1995. The Idenfication and Classification of Unknown Words in Chinese: A N-gram- Based Approach. In A. Ishikawa and Y. Nitta Eds. The Proceedings of the 1994 Kyoto Conference. A Festschrift for Professor Akira Ikeya. 113-123. Tokyo: The Logico-Linguistics Society of Japan.

Zhang, Y. and X. Qi. 1997. The Statistics[s] and Analysis of Words Included in Several Chinese Dictionaries.[In Chinese] In L.W. Chen and Q. Yuan Eds. Language Engineering. 82-87. Beijing: Qinghua U. Press.

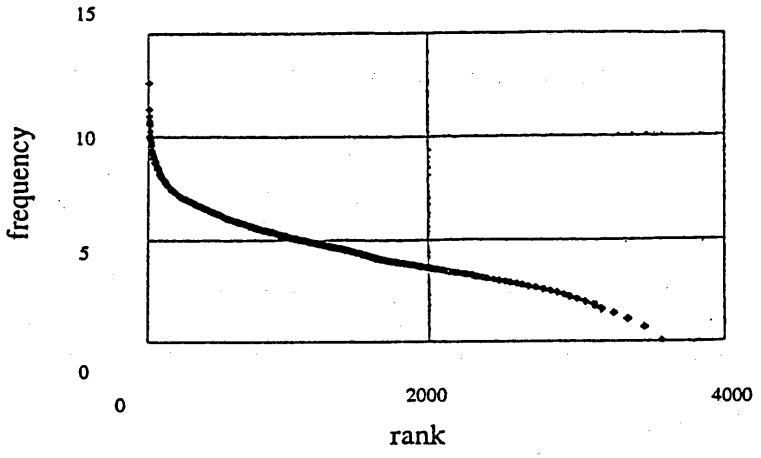


Diagram 1. CILIN entries frequency distribution in Sinica Corpus (Zipf's Law)
$Y = log\ f,\ X = rank$

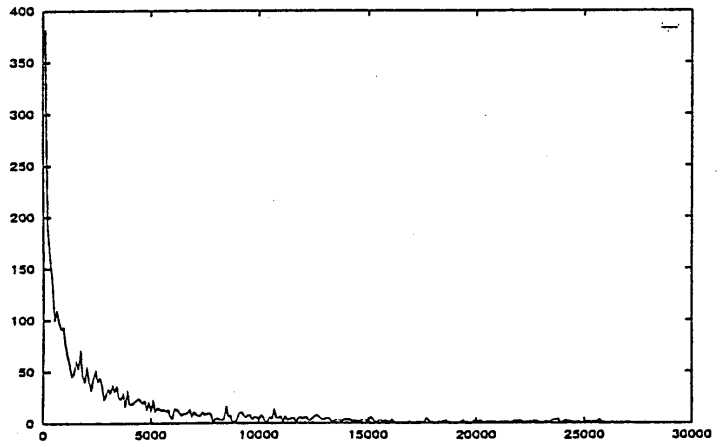


Diagram 2. CILIN entries distribution by frequency range in Sinica Corpus

(number of CILIN entries per every 1,000 rank interval)

Speaker-Independent Continuous Mandarin Speech Recognition Under Telephone Environments

Jia-lin Shen[1], Ying-chieh Tu[2], Po-yu Liang[2], Lin-shan Lee[1, 2]
1. Institute of Information Science, Academia Sinica
2. Department of Electrical Engineering, National Taiwan University
Taipei, Taiwan, R.O.C.
Tel. 886-2-27883799 ext. 2414, Fax. 886-2-27824814
Email : jlshen@iis.sinica.edu.tw

Abstract

This paper presents a study on speaker-independent continuous Mandarin speech recognition over the telephone. A comparison of several cepstral bias removal techniques such as cepstral mean subtraction (CMS), signal bias removal (SBR) and stochastic matching (SM) for telephone channel compensation was first investigated. Then some modifications and combinations of these techniques were developed for further improvement of the environmental robustness under telephone environments. To better estimate the contextual acoustics and co-articulation in spontaneous telephone speech, the between-syllable context-dependent phone-like units (such as triphones, biphones and demiphones) were used to train the speech models. In addition, the discriminative capabilities of the speech models were further enhanced using the minimum classification error (MCE) algorithms. Experimental results showed that the achieved recognition rates for Mandarin syllables were as high as 59.53%, which indicated a 27.81% of error rate reduction.

1. Introduction

During the past few years, interest has increased in developing spoken dialogue systems over the telephone [1]. Apparently, the recognition performance under telephone environments becomes crucial for a successful spoken dialogue system [2-3]. However, many problems arise from high-quality microphone to telephone networks such that the telephony based speech recognition is still very challenging. First, the speaker independence is highly desired in telephone environments. Secondly, the environmental variabilities become much more serious due to the channel distortions and the fairly high ambient background noise levels. Thirdly, the spontaneous speech over the telephone is very often ill-structured and co-articulated [4-5]. In this paper, some methods for overcoming these problems were developed and investigated.

As we know, the channel noise is usually convoluted with the speech signal in time domain, which becomes an additive term in the logarithmic spectral domain or cepstral domain. Therefore the channel noise can be compensated by subtracting a bias term from the noisy speech signal in cepstral domain (called cepstral bias removal). A comparative study of some widely used cepstral bias removal techniques such as cepstral mean subtraction (CMS)[6], signal bias removal (SBR)[7] and stochastic matching (SM)[8] were first investigated. Then some modifications and combinations were applied based on these techniques for further improvement of the environmental robustness under telephone environments. In order to better estimate the contextual acoustics and co-articulation in spontaneous telephone speech, the between-syllable context-dependent phone-like units (such as triphones, biphones and demiphones) are modeled. Moreover, the minimum classification error (MCE) algorithms are further used to enhance the discriminating ability of the speech models [9].

The baseline system is based on the context-dependent phone-like units (PLU)

considering the within-syllable parts only and without any compensation, in which the average recognition rates for Mandarin syllables were 43.94%. The recognition accuracies can be immediately increased to 49.24% using the cepstral bias removal techniques for channel noise compensation and further improved to 58.56% when the between-syllable context-dependent phone models are used. Furthermore, the achieved recognition rates were improved to as high as 59.53% using the minimum classification error algorithms as the post processing, which indicated a 27.81% of error rate reduction as compared to the baseline system.

This paper is organized into 5 sections. Section 2 describes the baseline recognition system and the speech database used in the experiments. The cepstral bias removal techniques are described in section 3. In section 4, the experiments based on different types of between-syllable context-dependent phone models are performed and discussed. Section 5 finally gives the concluding remarks.

2. Baseline Recognition System

2.1 Speech Database

The speech database was produced by 59 male and 54 female speakers over the telephone provided by Telecommunication Laboratories, Taiwan, Republic of China. Each speaker produced 120 Mandarin sentences such that a total of 13,560 Mandarin sentences (5.87 hrs) are included in the speech database. The signal-to-noise ratios (SNR) of this database are distributed from 10 to 40 dB, in which 9.09%, 56.36% and 34.55% of this database locate in 10~20 dB, 20~30 dB and 30~40 dB, respectively. In the following experiments, 51 male and 49 female speakers were used to train the gender-dependent, speaker-independent models and the rest 8 male and 5 female speakers were used as the testing speakers.

2.2 Front-end Processing

The telephone speech, which has a band of 150 Hz ~ 3.8 kHz, was sampled at an 8k Hz rate. After end-point detection is performed, 32 ms hamming window is applied every 10 ms with a pre-emphasis factor of 0.95. 14-order mel-frequency cepstral coefficients (MFCC) were derived from the power spectrum filtered by a set of 30 triangular band-pass filters. In addition, the first order derivatives of the 14 mel-frequency cepstral coefficients as well as the first and second order derivatives of the log short-time energy were also calculated to result in a feature vector of 30 dimensions for each frame [10].

2.3 Acoustic Modeling

The basic speech units used for recognition in this study are phone-like units (PLU) [11-12], in which a total of 34 context-independent (CI) PLU's are included. In fact, the most widely used units in the Mandarin speech recognition are the 22 Initial's and 40 Final's, where Initial means the initial consonant and Final means the vowel part but including possible media and nasal ending [10]. This is because of the mono-syllabic structure of the Mandarin Chinese, in which each Mandarin syllable can be decomposed into an Initial/Final format. One can note that each Initial is represented by one phoneme while each Final contains one to several phonemes. Accordingly, the numbers for the context-independent (CI) Initial/Final and PLU are 34 and 62, respectively. Also, when the right context dependency is considered, i.e., the speech units are regarded as different ones with respect to the beginning phonemes of the following units, the numbers for the right context dependent (RCD) Initial/Final and PLU can be expanded into 149 and 145, respectively. However, when the inter-syllable transitions are considered, the numbers for the RCD Initial/Final and PLU are immediately increased to 1269 and 480, respectively. Furthermore, if both the right and the left context dependencies are included, the numbers for Initial/Final and PLU will be further increased to 13,336 and 4605,

respectively. One can find that the amount of Initial/Final units is nearly 3 times of that of phone-like units considering both the left and the right contextual effects. Because it is highly necessary to model the contextual acoustics and co-articulation in spontaneous telephone speech, we choose the PLU as the basic speech unit. The 3-state left-to-right continuous hidden Markov model (CHMM) [13] was trained for each PLU and the number of mixtures per state is dynamically determined by the amount of available training data with a maximum of 8 mixture components.

The block diagram of the training phase is shown in Fig. 1. The context-independent (CI) PLU based models are first obtained using the forward-backward algorithm, in which the initial model parameters were derived from uniform segmentation. Then the CI-PLU models were used as the initial seed models to derive the within-syllable CD-PLU models using the forward-backward algorithm. Furthermore, the between-syllable CD-PLU models can be trained using the within-syllable CD-PLU models as the initial models. Finally, the minimum classification error (MCE) algorithms are used for further enhancement of the discriminative capability of the between-syllable CD-PLU models.

2.4 Performance Baseline

This recognition process is based on the Viterbi search algorithm for obtaining the optimal Mandarin syllable sequence. Also, the recognition rates are evaluated as one minus substitution rates, insertion rates as well as deletion rates. In the baseline experiments, the within-syllable right-context-dependent (RCD) PLU's were used as the speech units. The average recognition rates for male and female testing speakers were 45.30% and 42.57% respectively as shown in the Table 1.

	male	female	average
Recognition rates(%)	45.30	42.57	43.94

Table 1 : The baseline experimental results using 145 within-syllable right-context-dependent phone-like units.

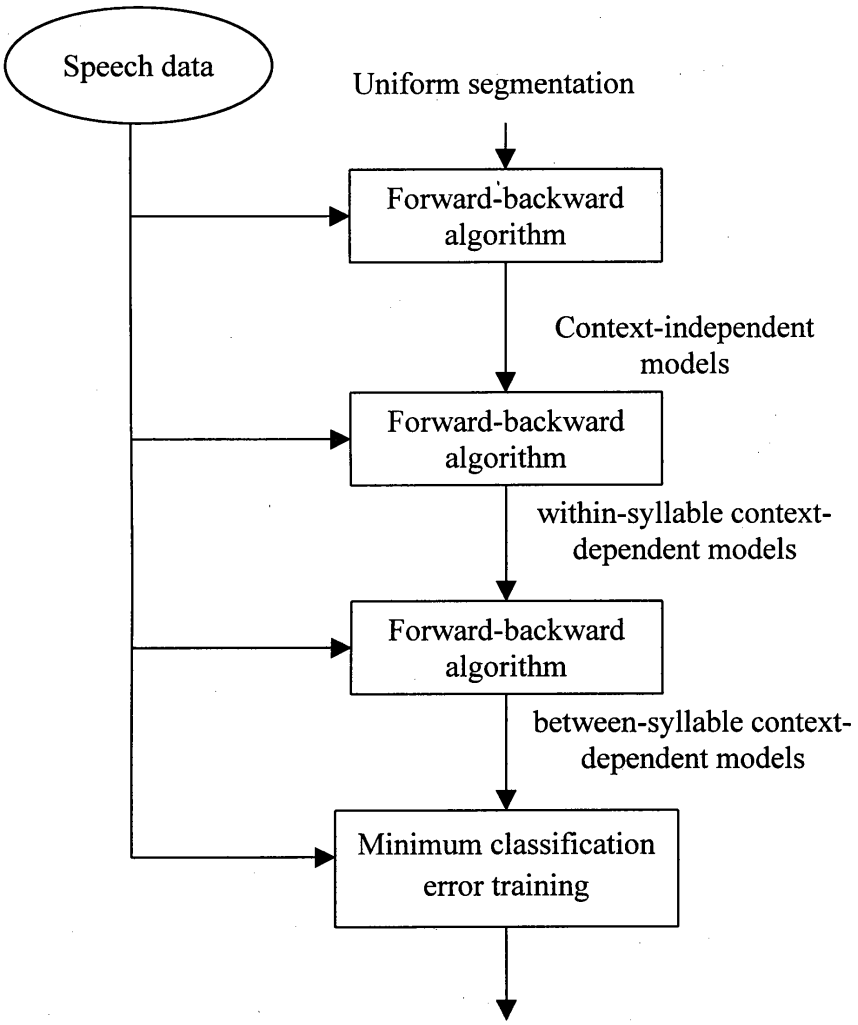


Figure 1 : The block diagram of the training procedure.

3. Cepstral Bias Removal

As mentioned previously, the channel noise is convoluted with the clean speech signal in time domain and becomes additive in logarithmic spectral domain or cepstral domain. Therefore, the corrupted speech signal y can be represented by the bias transformation $y = x + h$, where y, x and h denote the cepstral representations for noisy speech, clean speech and channel noise, respectively. The cepstral bias removal techniques are thus developed to estimate the cepstral bias h and then subtract the bias from the noisy speech cepstral vectors. Three kinds of widely used cepstral bias removal techniques are discussed and improved in the following, including cepstral mean subtraction (CMS), signal bias removal (SBR) and stochastic matching (SM).

3.1 Cepstral Mean Subtraction (CMS)

In CMS [6], we make the assumptions that the cepstral mean of speech signal over a long time equals to zero such that the cepstral bias of channel noise can be estimated by long-time average of the noisy speech cepstral vectors.

$$h = \frac{1}{T}\sum_{t=1}^{T} y_t , \qquad (1)$$

where y_t means the noisy feature vector at frame t with a total of T frames. A few methods are investigated here for the estimation of the cepstral bias h in CMS, depending on the amount T of the speech data.

1. Global bias : A single bias vector is estimated with all of the available training speech data and shared by all of the training speakers.
2. Speaker-dependent bias : The bias vectors are estimated for each speaker separately such that a total of 100 bias vectors are obtained for all the 100 training speakers, respectively.
3. Sentence-dependent bias : Each sentence can obtain its individual bias vector for the compensation of the channel noise.

4. Sequential sentence-dependent bias : It is very often that the estimation of the cepstral bias is coarse on a sentence-by-sentence basis due to the insufficient length for an individual sentence. Therefore, the cepstral bias is sequentially obtained by the interpolation of the current estimate with the previous estimates.

The experimental results are shown in Table 2. One can find that the performance was even degraded using the global bias in CMS (43.03% vs. 43.94%). This is probably because the channel effects in telephone environments are almost constant for a given call but vary with calls such that a single bias can not represent the channel effect very well and even smears the speech signal characteristics. However, when the speaker-dependent cepstral biases are used, the average recognition rates can be improved from 43.94% to 48.86%, which indicates a 8.78% of error rate reduction. Also, the sentence-dependent bias estimation provides an average recognition rate of as high as 46.01%. It is apparent that the compensation due to the speaker-dependent bias outperforms that using the sentence-dependent bias. However, the sentence-dependent bias estimation is much more practical and feasible in real-world applications. Therefore, the sequential sentence-dependent bias estimation approach is developed to incrementally update the cepstral bias. It can be noted that comparable recognition rates with that using the speaker-dependent cepstral bias were achieved based on the sequential sentence-dependent bias estimation (48.74% vs. 48.86%).

	male	female	average
Global	44.53	41.52	43.03
Speaker-dependent	50.86	46.86	48.86
Sentence-dependent	48.78	43.23	46.01
Sequential sentence-dependent	51.01	46.46	48.74

Table 2. The experimental results using different cepstral bias estimation methods in cepstral mean subtraction (CMS).

3.2 Signal Bias Removal (SBR)

In SBR [7], a codebook Ω is first trained using all the available training data and the cepstral bias is obtained by maximizing the likelihood function $p(Y|\ h,\ \Omega)$, where Y means a set of noisy speech vectors $Y = \{y_1, y_2, \ldots, y_T\}$.

$$v_t = \arg\max_j p(y_t \mid h, \Omega_j), \tag{2}$$

$$h = \frac{1}{T}\sum_{t=1}^{T}(y_t - v_t) \tag{3}$$

where v_t designates the encoded codeword for the observation vector y_t at frame t. Apparently, CMS is a special case of SBR with the codebook size set to 1. In this study, three kinds of codebooks are developed, including *ad hoc* codebook, hierarchy codebook and phone-dependent codebook. In the *ad hoc* codebook, the codebook size is fixed and the codewords are trained using all the training speech based on the LBG algorithm, while in the hierarchy codebook, the codebook size is gradually increased such that the cpestral bias can be hierarchically updated using the codebook from smaller size to larger size. Instead of the data-driven codebook by vector quantization methods, the phone-dependent codebook is used, i.e., the training data corresponding to same context-independent PLU is clustered such that a total of 34 codewords cab be obtained.

On the other hand, in the encoding process, the soft decision is used for the estimation of the cepstral bias such that eq. (3) is expressed as below.

$$h = \frac{1}{T}\sum_{t=1}^{T}[\sum_{k=1}^{m} w_t^k (y_t - v_t^{\ k}) / \sum_{k=1}^{m} w_t^k] \tag{4}$$

where $v_t^{\ k}$ means the k-th nearest codeword for the observation vector y_t and $w_t^k = 1/\| y_t - v_t^{\ k} \|^2$ is the corresponding weighting factor.

Table 3 shows the experimental results using different types of codebook in SBR. It can

be found that competitive recognition accuracies can be obtained using the *ad hoc* codebook with different sizes (46.79%, 46.48% and 46.26% for codebook size of 16, 32 and 64, respectively). In addition, when the hierarchy codebook is used where the codebook size is gradually increased from 16, 32 to 64, the recognition rates can be further improved to 47.35%. As shown in the last row of Table 3, the phone-dependent codebook can further provide slight improvement in recognition rates up to 47.50%. In Table 4, the encoding processes based on soft decision and hard decision are compared, in which the recognition rates can be further improved by 0.3%~0.5% using the soft decision for different types of codebook.

codebook type	codebook size	male	female	average
ad hoc	16	48.68	44.89	46.79
	32	48.73	44.22	46.48
	64	48.41	44.11	46.26
hierarchy	16,32,64	48.80	45.90	47.35
phone-dependent	34	49.33	45.67	47.50

Table 3. The experimental results using different types of codebook in signal bias removal (SBR).

codebook type	decision type	male	female	average
ad hoc(64)	hard	48.41	44.11	46.26
	soft	49.21	44.31	46.76
hierarchy	hard	48.80	45.90	47.35
	soft	49.41	45.96	47.69
phone-dependent	hard	49.33	45.67	47.50
	soft	49.81	46.04	47.93

Table 4. The comparative experimental results using hard decision and soft decision in encoding process in signal bias removal (SBR).

3.3 Stochastic Matching (SM)

In SM [8], the bias transformation function ($y = x + h$) is used to map the input corrupted speech onto the acoustic space of speech models such that the recognition process can be performed in matched conditions. The cepstral bias h can then be estimated in a maximum likelihood manner.

$$\begin{aligned} S^{(n+1)} &= \arg\max_{S} p(Y, S^{(n)} \mid h^{(n)}, \Lambda_X) \\ h^{(n+1)} &= \arg\max_{h} p(Y, S^{(n+1)} \mid h^{(n)}, \Lambda_X) p(S^{(n+1)}) \end{aligned} \tag{5}$$

where $S^{(n)}$ denotes the state sequence at the n-th iteration while Λ_X means the speech models. Suppose Λ_X is modeled by Gaussian distributions, the cepstral bias can be estimated in the following.

$$h = \frac{\sum_{t=1}^{T} \sum_{n} \sum_{m} \gamma_t(n,m) \Sigma_{n,m}^{-1} (y_t - \mu_{n,m})}{\sum_{t=1}^{T} \sum_{n} \sum_{m} \gamma_t(n,m) \Sigma_{n,m}^{-1}} \tag{6}$$

where $(\mu_{n,m}, \Sigma_{n,m})$ denotes the mean vector and covariance matrix of the speech models at state n and mixture m while $\gamma_t(n,m)$ means the corresponding posterior probability observing the feature vector y_t at frame t. In comparison with the formulations of the cepstral bias estimation in eqs. (3) and (6) basded on SBR and SM separately, we found that similar forms can be obtained, i.e., the weighting average of the difference between the noisy feature vectors and the corresponding centroids in the acoustic space of training data. However, the corresponding centroid for each observation vector comes from the speech models by Viterbi decoding in SM while in SBR it is obtained by the vector quantization process of a training codebook. In addition, because the cepstral bias is iteratively updated in the recognition process in the SM method, better initial estimate of the bias can provide better improvement of the performance. In other words, the SM method can be applied as the post processing after the CMS or SBR compensation is used. The block diagrams of the three kinds of cepstral bias

removal techniques discussed in this section are shown in Fig. 2.

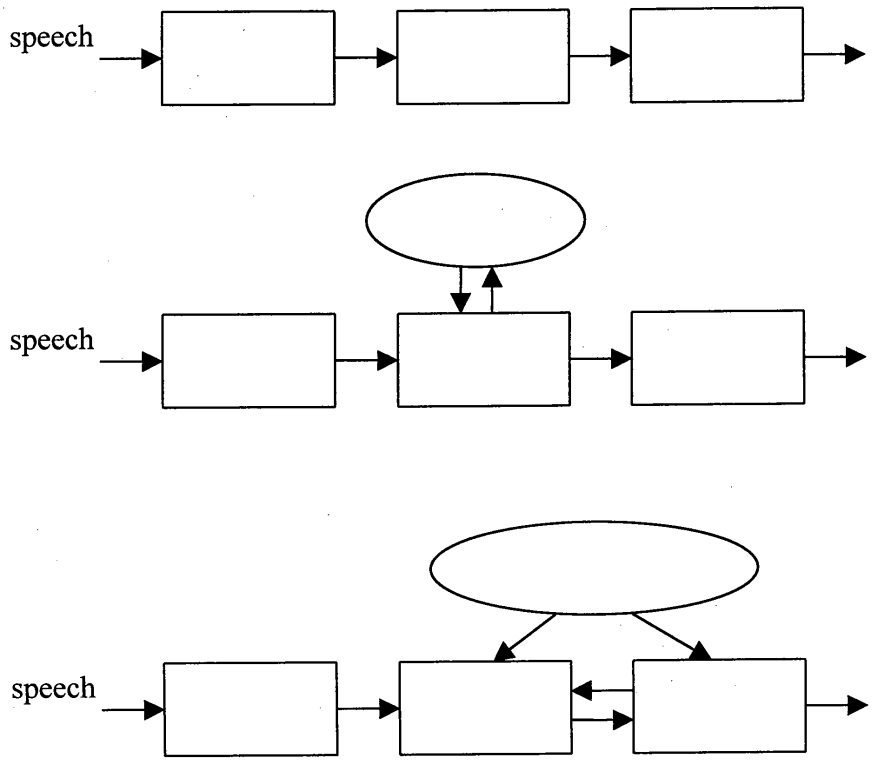


Table 5 shows the experimental results based on SM approach. Note that although the recognition rates can be increased from 43.94% to 45.56%, the improvements are indeed the least as compared to the CMS and SBR methods. This is probably due to the mis-classified labeling of the observation vectors in the model matching process. That is, the corresponding distribution $(\mu_{n,m}, \Sigma_{n,m})$ in eq. (6) for the feature vector y_t is probably incorrect. Therefore, better speech models can provide more correct labelling results and thus better estimation of the cepstral bias can be obtained. As shown in the last two rows of Table 5, when the SM is

used as the post-processing after the CMS or SBR is performed, the performance can be further improved. The recognition accuracy using the combination of SBR and SM outperforms that using SBR only (47.93% vs. 47.48%) and so does CMS (49.24% vs. 48.86%).

	male	female	average	origin
SM	46.35	44.77	45.56	--
SBR+SM	49.62	46.23	47.93	47.48
CMS+SM	51.46	47.02	49.24	48.86

Table 5. The experimental results using different initial process in stochastic matching (SM).

4. Between-syllable Context-dependent Phone Models

4.1 Between-syllable Context-dependent Phone-like Units

In order to deal with the inter-syllable context variations for further improvement of continuous Mandarin speech recognition, the between-syllable triphone models are used. In other words, each speech model represents a phone with specific left and right contexts [14-15]. As mentioned previously, the number of triphones is 4605 for the 34-phone set, which is more than 30 times of that for the 145 within-syllable RCD phones used in the baseline system. Apparently, the trainability will become poor due to the insufficient amount of training data. In this study, we adopt two ways to increase the trainability using the triphone models.

1. Back-off : When the occurrence of a triphone unit in the training database is less than a pre-defined threshold, this triphone is replaced by its corresponding context-independent phone unit or context-dependent biphone unit considering left or right context dependency only.

2. Sharing : The triphone units are tied together by the linguistic constraints.

• Biphone : Unlike the triphone units that depend on both the right and the left context, the biphone units only depend on single context. Therefore, the right context-dependent (RCD) and left context-dependent (LCD) biphone units are used instead.

• Demiphone : Each demiphone unit can be divided into two sections where the right part is dependent on the right context while the left part depends on the left context, separately. In this way, the needed number of mixture components will not be increased if the number of state per phone model is unchanged [16].

The structures of the between-syllable context-dependent phone based hidden Markov models are shown in Fig. 3, including triphone, biphone and demiphone units. To further improve the discriminative capability of the speech models, the minimum classification error (MCE) algorithm can be used as the post-processing in the training procedure [9]. During the MCE training, the model parameters are iteratively adjusted in a maximum discriminability manner such that the recognition errors can be minimized for the training speech database.

model	male	female	average
Intra-LCD phone	48.21	38.65	43.43
Intra-RCD phone	51.53	46.75	49.19
Triphone	58.92	54.68	56.80
Inter-RCD phone	60.52	56.59	58.56
Inter-demiphone	59.20	54.88	57.04
Intra-RCD Initial/Final	52.92	48.55	50.74
Inter-RCD Initial/Final	59.41	51.51	55.46

Table 6. The experimental results based on different types of context-dependent speech units (intra- denotes within-syllable while inter- denotes between-syllable).

4.2 Experiments

In this subsection, we investigate the recognition performance based on different types of context-dependent phone-like speech units. Here the cepstral mean subtraction (CMS) technique based on speaker-dependent cepstral bias estimation discussed previously is used as the front-end robust processing. Also, an extra silence model is added for the improvement of the speech end-point detection. As shown in first two rows of Table 6, the recognition results using within-syllable left context-dependent (LCD) and right context-dependent (RCD) are compared. It can be found that the right contextual effects are more influential on the recognition accuracy than that of left contexts (49.19% vs. 43.43%). Also, slight improvement can be obtained with the addition of the silence model as compared to the result shown in the second row of Table 2 (48.86% vs. 49.19%). Then, when the triphone based models are used, the recognition rates can be immediately improved to 56.80%, in which the error rates are reduced by 14.98% with the expense of more than 30 times of mixture components as shown in Fig. 4. It is noted that there exist around 2600 unseen triphones out of 4605. Here the back-off method is applied using between- syllable RCD PLU's to predict the unseen triphones. When the biphone and demiphone units are further used to tie the states of the triphone based models, the needed mixture components can be reduced from 55,272 to 7,701 and 10,480 respectively as also shown in Fig. 4. The recognition accuracies are also improved from 56.80% to 58.56% and 57.04% respectively as listed in Table 6. In other words, the trainability as well as the sensitivity can be increased by sharing the parameters of the triphone models. As a comparison, the within-syllable and between-syllable RCD Initial/Final based models are trained and the results are also shown in Table 6. One can find that although the recognition rates using Initial/Final units outperform that using PLU's considering within-syllable right context variations only (50.74% vs. 49.19%), the error rates and needed mixture components are greatly increased when the between-syllable context dependency is included

as also shown in Table 6 and Fig. 4. It is indicated that the error rates are reduced by 6.96% using less than one half of mixture components compared with between-syllable RCD PLU and Initial/Final based models. Finally, when the minimum classification error (MCE) training algorithm is applied to the most successful between-syllable RCD PLU based models, the recognition rates can be further improved from 58.56% to 59.53% as shown in Table 7. In comparison with the baseline system listed in Table 1, the recognition rates are increased from 43.94% to 59.53%, which indicates a 27.81% of error rate reduction.

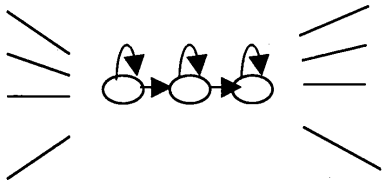
Inter-RCD phone	male	female	average
ML	59.41	51.51	58.56
MCE	61.78	57.28	59.53

Table 7. The comparative results using ML and MCE training based on between-syllable RCD phone models.

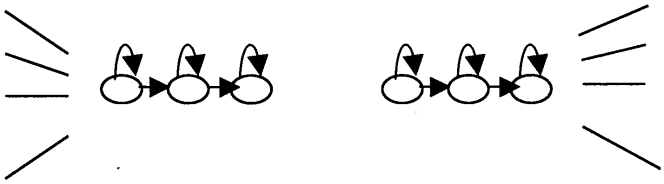
3. Conclusion

This paper presents a study on speaker-independent continuous Mandarin speech recognition under telephone environments. The widely used cepstral bias removal techniques (CMS, SBR and SM) were first compared and improved. Then the between-syllable context-dependent phone models (triphones, biphones and demiphones) were trained. The minimum classification error (MCE) training algorithm was further applied. Experimental results showed that the achieved recognition rates can be improved from 43.94% to 59.53% as compared to the baseline system using within-syllable RCD phone models.

(a)



(b)



(c)

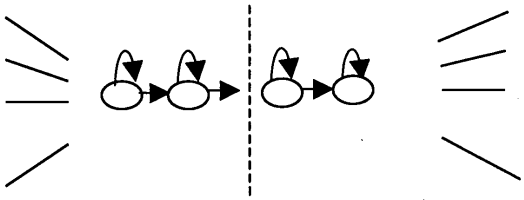


Figure 3. The structures of the between-syllable context-dependent phone based hidden Markov models (HMM) : (a). triphone, (b). biphone and (c). demiphone.

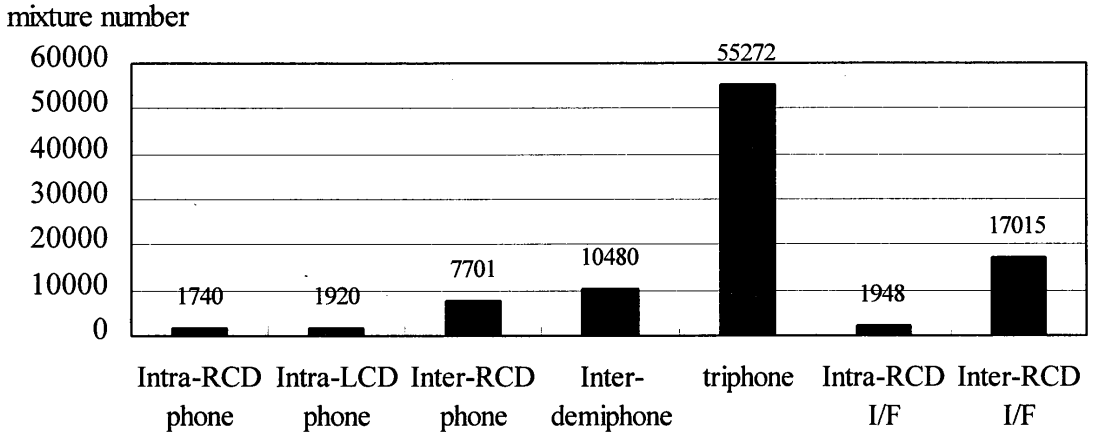


Figure 4. The total number of mixture components for the acoustic models based on different types of speech units.

References

1.R. Cole, *et.al.*, "The challenge of spoken language systems : research directions for the nineties", *IEEE Trans. On Speech and Audio Processing*, Vol. 3, No. 1, Jan. 1995, pp. 1-21.

2.J. Takahashi, N. Sugarmura, T. Hirokawa, S. Sagayama & S. Furui, "Interactive voice technology development for telecommunication applications", *Speech Communication*, 17:pp. 287-301, 1995.

3.D. Johnson, "Telephony based speech technology – from laboratory visions to customer applications", *Journal of Speech Technology*, Vol. 2, No. 2, Dec. 1997, pp. 89-100.

4.C. Mokbel, D. Jouvet & J. Monne, "Deconvolution of telephone line effects for speech recognition", *Speech Communication*, Vol. 19, pp. 185-196.

5.P.J. Moreno and R.M. Stern, "Source of degradation of speech recognition in the telephone network", in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, 1994, pp. 109-112.

6.S. Furui, "Cepstral analysis technique for automatic speaker verification", *IEEE Trans. Acoust. Speech, Signal Processing*, ASSP-29, Apr. 1981, pp. 254-272.

7.M.G. Rahim & B.H. Juang, "Signal bias removal by maximum likelihood estimation for robust telephone speech recognition", *IEEE Trans. on Speech and Audio Processing*, Vol. 4, No. 1, pp. 19-30, Jan. 1996.

8.A. Sankar & C.H. Lee, "A maximum likelihood approach to stochastic matching for robust speech recognition", *IEEE Trans. on Speech and Audio Processing*, Vol. 4, No. 3, pp. 190-202, May, 1996.

9.B.H. Juang, W. Chou, C. H. Lee, "Minimum Classification Error Rate Methods for Speech Recognition", *IEEE Trans. On Speech and Audio Processing*, Vol. 5, No. 3, pp. 257-265, May 1997.

10. L.S. Lee, "Voice dictation of Mandarin Chinese", *IEEE Signal Processing Magazine*, Vol.

14, No. 4, pp. 63-101, July 1997.

11. R.Y. Lyu, H.M. Wang & L.S. Lee, "A comparison of different units applied to isolated/continuous large vocabulary Mandarin speech recognition", in *Proc. Int. Conf. Computer Processing of Oriental Language*, May 1994, pp. 211-214.
12. C.H. Lee & B.H. Juang, "A survey on automatic speech recognition with sn illustrative example on continuous speech recognition of Mandarin", *Computational Linguistics and Chinese Language Processing*, Vol. 1, No. 1, Aug. 1996, pp. 1-36.
13. L.R. Labiner, "A tutorial on hidden Markov models and selected applications in speech recognition", *Proc. IEEE,* 77(2) : 257-286, Feb. 1989.
14. K.F. Lee, "The SPHINX speech recognition system", in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, 1989, pp. 445-448.
15. J. J. Odell, "The use of context in large vocabulary speech recognition", *Ph.D. dissertation*, Queen's college, UK, Mar. 1995.
16. J.B. Marino, A. Nogueiras, A. Bonafonte, "The dimiphones : an efficient subword units for continuous speech recognition", *Int. Conf. Eurospeech*, pp. 1215-1218, 1997.

A Large-Vocabulary Taiwanese (Min-nan) Speech Recognition System Based on Inter-syllabic Initial-Final Modeling and Lexicon-Tree Search

Ren-yuan Lyu[1], Yuang-jin Chiang[2], Ren-zhou Fang[2], Wen-ping Hsieh[2]
[1] Chang Gung University [2] National Tsing Hua University
Email: rylyu@mail.cgu.edu.tw, rylyu@ms1.hinet.net
Tel: 886-3-3283016#5677

Abstract

In this paper some preliminary work about Taiwanese (Min-nan) speech recognition research has been done and described. Also, we report some pioneer experimental results on an initial study about a large-vocabulary (with 20 thousand words) Taiwanese multi-syllabic word recognition system. For the speaker dependent case, 9.4% word error rate is achieved. A real-time prototype system implemented on a Pentium-II personal computer running MS-Windows95/NT is also shown to validate the approaches proposed in this paper.

1. Introduction

Taiwanese, one of the major Chinese dialects, is the mother tongue of more than 75% of the population in Taiwan. It belongs to a larger Chinese dialectical family called Min-nan (or Southern-Min, Southern-Hokkian), which is also used by many overseas Chinese living in Singapore, Malaysia, Philippine, and other areas of Southern-East Asia. It was estimated that this language has more than 49 millions speakers and is ranked in the 21th place in the world, according to the 1996 Ethnology. In the past few decades, scientists in Taiwan do speech recognition research on Mandarin speech. Some achievements have been achieved in recent years.[Lyu95] Since Taiwanese is another major language spoken in this land, and Taiwan is basically a multilingual society, we decided to develop a similar large-vocabulary speech recognition system for Taiwanese speech.

In this paper, some preliminary work has been done, including the study of Taiwanese phonetics, setting up a Taiwanese lexicon and a set of phonetic alphabet to symbolize

Taiwanese speech, selecting several sets of phonetically balanced words to be used in speech data collection, and recording a Taiwanese speech database.

The basic technology adopted here is the continuous Hidden Markov Model (CHMM) because of its success in speech recognition in the past decades. We adopt CHMM to model the Taiwanese Initial/Final phonetic units, considering both the inside- and inter-syllabic coarticulation. A promising result, with the error rate being 9.4%, for the speaker dependent case was obtained.

Additionally, a real-time prototype system in a Pentium-II personal computer running MS-Windows95/NT was implemented for further study and to validate the approaches we proposed here.

This paper is organized as follows: First, the Taiwanese Phonetics was summarized in section 2. The scripts, lexicon, and database were described in section 3. The front-end signal processing was then described in section 4. Section 5 was about the selection of the speech units. Section 6 was about the search in the lexicon tree. In section 7, we reported the results of the experiments and some discussions. A prototype system was shown and several concluding remarks were finally given in section 8.

2. Taiwanese Phonetics

Taiwanese, like Mandarin as a member of Sino-Tibetan language family, is a tonal, monosyllabic language. Traditionally, a Taiwanese syllable is decomposed into three parts, namely an Initial, a Final and a tone. Take the syllable "dan4" (to wait) as an example, where /d/ is an Initial, /an/ is a Final, and /4/ represents a high-falling tone. There are 18 Initials (including one null Initial), 47 Finals, and 7 lexical tones in Taiwanese. [Wang54] [Cheng97] An Initial is equivalent to a consonant, but a Final can be further decomposed into 1 to 3 vowels plus possible consonants. In particular, for each Final, there is a corresponding "entering-tone" Final, which is ended with an unreleased /p/, /t/, /k/ or /h/. All the phonemes are listed in <table.1>, each of which has a corresponding Chinese character with that phoneme as part of its pronunciation. The phonemes are represented by 3 alternative symbolic systems, including the International Phonetic Alphabet (IPA), the Mandarin Phonetic Alphabet (MPA), and a set of specially designed phonetic alphabet called Daiim, where Daiim is especially convenient to encode Taiwanese speech, and is adopted in the following parts of this paper.[Chiang94] All the Initials and Finals are thus listed in <table.2> and

<table.3> with their corresponding Chinese characters and Daiim representations. (Note that: Some Taiwanese syllables used in daily conversation have no widely accepted Chinese characters, and we use "○" to represent such syllables).

Furthermore, Taiwanese is also a tonal language with more complex tonal structures than that of Mandarin. It has 7 lexical tones, two of which are carried in syllables ending with final /p, t, k, h/ (called entering-tone) and the other five are carried in those not ending with final /p, t, k, h/ (called non-entering tone). An example of these 7 tones with one corresponding Chinese character for each tone is shown in <table.4>. [Chinag97] Some acoustic characteristics, including the waveform, the contour of fundamental frequency, the description of relative frequency level, and the traditional phonological order [Zhou97][Cheng97] are also shown in <table.4>. The tone sandhi issue is even more complex and beyond the discussion of this paper.

Since the task we are considering here is the recognition of multi-syllabic words, which have relatively few homonyms even when the tones are disregarded. In this initial study, we decided not to deal with the issues of tones and then reduce the 1683 phonologically allowed tonal syllables to 714 base syllables. That is, each word in the lexicon is represented as a concatenation of base syllables. The word recognition task becomes the recognition of base syllable strings.

3. Training Script, Lexicon, Testing Script and Database

To initiate the study of the large-vocabulary speech recognition of a new language, like Taiwanese we are studying here, one of the most important preliminary jobs is to construct a pronunciation lexicon. For this, we have set up a Taiwanese pronunciation lexicon of about 20 thousand words, each of them has a corresponding string of phonetic symbols encoded in Daiim phonetic alphabet. [Chiang94] In this lexicon, there are 19152 ordinarily used Taiwanese words, composed of 48318 syllables, i.e., each word contains 2.52 syllables in average.

Another important preliminary task is to select a training script which contains as few words but as much phonetic variety as possible. To achieve this, a word selective procedure is set to choose appropriate words as follows:

1) Determine the phonetic unit to be used in the recognition system;

2) Each new word selected contains the maximal number of possible new phonetic units;

3) Include all distinct speech units which appear in the lexicon.

As a result, a minimal set of 472 words containing all the 1029 distinct Right-Context-Dependent (RCD) phonemes found in the lexicon were selected. In addition, several extended sets of words, which contain as many distinct RCD phonemes as possible, were also selected to enhance the phonetic variety. Furthermore, a set of single-syllabic words, containing all 2874 phonologically possible syllables, was picked out, too. The statistics of all the sets of words used in the training session is listed in <table.5>.

For evaluation of the recognition system, we select several sets of words with different features:

1) R1000: 1000 randomly selected words, each of which contains 2.55 syllables;
2) H500: 500 highest frequently used words, each of which contains 2.12 syllables;
3) N407: 407 place names, each of which contains 2.08 syllables;
4) P396: 396 phonetically rich words, each of which contains 3.24 syllables.

The statistics of the evaluation set is listed in <table.6>.

The speech database used for training and evaluation were recorded by two adult speakers, including one male and one female, over a period of one month. A close-talk head-set microphone plugging in a SoundBlaster card in a Pentium-II personal computer was used. The speech waveform was sampled at 16 KHz. The statistics of the speech database is also listed in <table.5> and <table.6>.

4. Front-end Signal Processing

The speech waveform was multiplied by a 16-ms Hamming window first. A set of 12-dimensional mel-cepstral coefficients and 1-dimentional log energy was extracted to form a 13-dimentional feature vector for each frame which shifts forward every 8 ms. A time window of 5 frames of feature vectors were used to compute the corresponding 13-dimentional delta coefficients. These 2 sequences of feature vectors and delta feature vectors were treated as statistically independent and modeled by separate Gaussian mixture densities in CHMM.

5. Selection of Speech Units

In this paper, we adopted Initial-Final's, considering the context dependency both inside

a syllable and inter syllables, as the basic speech units to be modeled as CHMM. It is believed that the coarticulation effect inside a syllable is more severe than that between 2 syllables for the monosyllabic language, such as Mandarin or Taiwanese. So, it is natural for researchers to consider the inside-syllable coarticulation in the previous literatures. [Lyu95] In such a case, only Initials can be right context dependent and all Finals are right context independent. There are thus 147 RCD Initial models and 77 CI Final models.(Note that there seem to be 47*2=94 Finals by observing <table.3>. But since some of them do not exist in the lexicon, they are not chosen in our training script and not used as the basic speech units.) However, when the speed of utterance increases the coarticulation across 2 syllables becomes severe. In addition, for the vowel-vowel concatenation between 2 neighboring syllables, the coarticulation effect may be very severe even when the speed of utterance is slow. To alleviate such a difficulty, the inter-syllabic modeling was considered. However, the number of general RCD Finals is so large that we chose not to use it directly. Instead, we added the inter-syllabic RCD bounded phones explicitly to model the coarticulation effect. For examples, the bi-syllabic word "pue-e" (皮鞋), will be looked upon as the concatenation of /p+u/, /ue/, /e+e/, and /e/, where the unit /e+e/ is what we called the inter-syllabic RCD bounded phone. By this approach, 105 additional units were obtained. As we will see in the following experiments, such an explicit consideration about the inter-syllabic coarticulation does decrease the word error rate at a little cost of additional computation.

6. Lexicon Tree Search

The 20K-word lexicon is organized in terms of the chosen speech units as a tree data structure to be used as the search space. There are about 58K nodes in the lexicon tree, with each node containing one chosen speech unit. Compared with a plain linear lexicon, which contains about 124K nodes, the tree lexicon saves more than a half storage space. In addition, the searching speed is much faster in the tree lexicon. A roughly estimate of speed improvement is more than 10 times! A sub-tree is shown in <fig.1>. A widely used Viterbi beam search is then used to find N best paths and then the N candidates of the recognized words. [Lee89]

7. Experiments and Discussions

The experimental results for the testing corpus are listed in <table 7>. The word error rates we achieved in this initial study in average for 2 speakers are 11.4% for inside-syllable modeling and 9.4% for inter-syllable modeling. The speed for each case is approximately the same.

From <table.7>, it is observed that the word error rate is lower when the average length of each word is longer. Also one can observe that the average length of words in the 4 testing sets is very close to the average length of words in the whole 20K-lexicon, as shown in <table.6>, where 2.49 and 2.52 syllables per word for the 4 testing sets and the whole lexicon respectively. It is thus safe to claim the recognition rate for the testing sets can represent well the recognition rate for the whole 20K-lexicon.

It is so surprising to observe that there is almost no increase in computation when we added 105 additional inter-syllabic RCD units! The reason is because the width of the beam of the Viterbi search was set to be constant, and thus there was almost the same number of states active in each forward calculation.

8. A Prototype System and Concluding Remarks

To validate the approaches proposed in this paper, a prototype system was implemented on a Pentium-II personal computer running MS-Windows95/NT. The block diagram of the system is shown as in <fig 2>, and the graphic user interface (GUI) is shown as in <fig.3>.

Compared with the speech recognition systems for the major languages in the world, such as English or Mandarin, the Taiwanese speech recognition research is still in the baby stage. However, since Taiwan is famous with her computer industry, and Taiwanese is so popular in Taiwan, we hope there are more and more researchers in Taiwan devote themselves in the study of this language. I hope in some day my old grandmother can talk to the computer in Taiwanese, which is the only one language for her to communicate.

9. Reference

[Lyu95] Ren-Yuan Lyu, et al. "Golden Mandarin (III)-User-Adaptive Prosodic-Segment-Based Mandarin Dictation Machine for Chinese Language with Very Large Vocabulary", ICASSP-95, pp57-60

[Wang57] 王育德,“台灣語常用語彙”, 永和語學社, 1957.

[Chinag94] 江永進,"台音式輸入法 version4.1”, 臺灣新竹清華大學統計所, 1994

[Chiang97] 許世楷等, 江永進執筆,“口語調自然調形”, 台灣世界 12 期, 台中市,1997

[Zhou97] 周長揖 康啓明 “台灣閩南語教程”, 1997

[Cheng97] Robert L. Cheng, "Taiwanese and Mandarin Structures and Their Developmental Trends in Taiwan--I: Taiwanese Phonology and Morphology", 1997

[Lee89] C.H. Lee, etc, " A frame-synchronous network search algorithm for connected Word recognition", IEEE Trans. ASSP, pp. 1649-1658, Nov. 1989

10. Tables and Figures

<table.1> A List of Phonemes in Taiwanese

Consonant				Consonant				Vowel			
IPA	Chinese Character	MPA	Daiim	IPA	Chinese Character	MPA	Daiim	IPA	Chinese Character	MPA	Daiim
p	保	ㄅ	b	ts	資	ㄗ	z	a	阿	ㄚ	a
p'	坡	ㄆ	p	ts'	此	ㄘ	c	i	伊	ㄧ	i
m	冒	ㄇ	m	s	思	ㄙ	s	u	有	ㄨ	u
b	帽		v	z	如		r	ɛ	鞋	ㄝ	e
t	刀	ㄉ	d	x	好	ㄏ	h	ɔ	烏	ㄛ	o
t'	討	ㄊ	t	ø	英			ə	蚵	ㄜ	or
n	怒	ㄋ	n					ã	餡		ann
l	路	ㄌ	l					ĩ	嬰		inn
k	糕	ㄍ	g					ũ	樣		unn
k'	科	ㄎ	k					ɛ̃	嬰		enn
ŋ	雅	ㄥ	ng					ɔ̃	惡		onn
g	鵝		q								

IPA: the International Phonetic Alphabet

MPA: the Mandarin Phonetic Alphabet widely used in Taiwan

Daiim: A specially designed Taiwanese Phonetic Alphabet used throughout this paper

<table.2> 18 Initials of Taiwanese syllables.

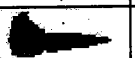
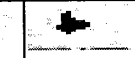
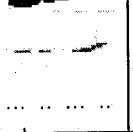
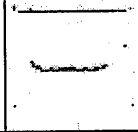
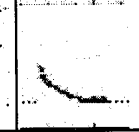
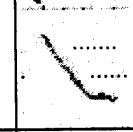
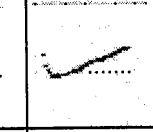
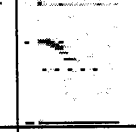
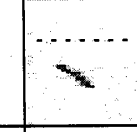
	Chinese Character	Daiim		Chinese Character	Daiim
1.	保	b	10.	科	k
2.	坡	p	11.	雅	nq
3.	冒	m	12.	鵝	q
4.	帽	v	13.	資	z
5.	刀	d	14.	此	c
6.	討	t	15.	思	s
7.	怒	n	16.	如	r
8.	路	l	17.	好	h
9.	糕	g	18.	英	(null initial)

<table.3> 47 finals and their counterparts for entering-tone in Taiwanese

	Chinese Character	Dai-im	Chinese Character	Dai-im (entering-tone)		Chinese Character	Dai-im	Chinese Character	Dai-im (entering-tone)
1.	阿	a	鴨	ah	25.	鴦	iunn	○	iunnh
2.	會	e	窄	eh	26.	妙	iaunn	○	iaunnh
3.	伊	i	裂	ih	27.	碗	uann	○	uannh
4.	烏	o	○	oh	28.	妹	uenn	○	uennh
5.	蚵	or	學	orh	29.	黃	uinn	○	uinnh
6.	有	u	○	uh	30.	橫	uainn	○	uainnh
7.	愛	ai	○	aih	31.	姆	m	○	mh
8.	後	au	○	auh	32.	秧	ng	○	ngh
9.	野	ia	頁	iah	33.	暗	am	盒	ap
10	腰	ior	葯	iorh	34.	安	an	扎	at
11	優	iu	○	iuh	35.	紅	ang	沃	ak
12	邀	iau	○	iauh	36.	蔘	om	○	op
13	娃	ua	活	uah	37.	汪	ong	惡	ok
14	話	ue	喂	ueh	38.	音	im	立	ip
15	威	ui	挖	uih	39.	因	in	一	it
16	歪	uai	○	uaih	40.	英	ing	益	ik
17	餡	ann	○	annh	41.	鹽	iam	葉	iap
18	嬰	enn	脈	ennh	42.	煙	en	拽	et
19	院	inn	物	innh	43.	央	iang	○	iak
20	惡	onn	膜	onnh	44.	勇	iong	育	iok
21	哼	ainn	○	ainnh	45.	溫	un	熨	ut
22	貌	aunn	○	aunnh	46.	彎	uan	越	uat
23	影	iann	○	iannh	47.	○	uang	○	uak
24	薑	ionn	○	ionnh					

Note: "○" represents the syllable which has no widely accepted Chinese character

<table.4> The 7 lexical tones of Taiwanese

漢字	東	洞	棟	黨	同	獨	督
Waveform							
Fundamental Frequency							
Relative Frequency	High level	Mid Level	Low falling	High falling	Rising	High Stop	Low stop
Traditional Tone order	1	7	3	2	5	8	4

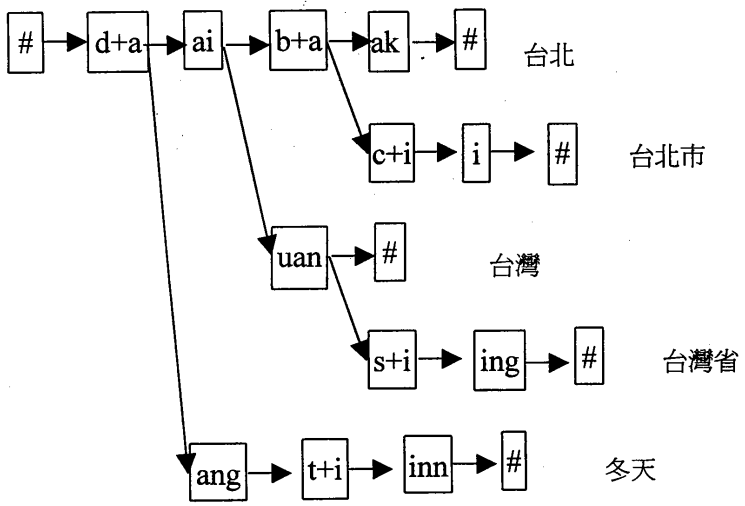
<table.5> Statistics of the Lexicon and the Training Word Sets

		Number of words	Number of distinct RCD phonemes	Speech Length in seconds	
				Male	Female
Training Word Sets	Single_syllble	2,874	213	1417	1486
	Min_word	472	1,029	459	445
	Ext_word	1,045	1,029	965	981
	The whole	4391	1029	2841	2912
Lexicon		19,152	1,029	N/A	N/A

<table.6> Statistics of the Lexicon and the Testing Word Sets

		Number of words	Number of syllables per word	Speech Length in seconds	
				Male	Female
Testing Word Sets	R1000	1000	2.55	826	656
	H500	500	2.12	361	397
	N407	407	2.08	304	311
	P396	396	3.24	385	256
	The whole	2303	2.49	1876	1620
Lexicon		19,152	2.52	N/A	N/A

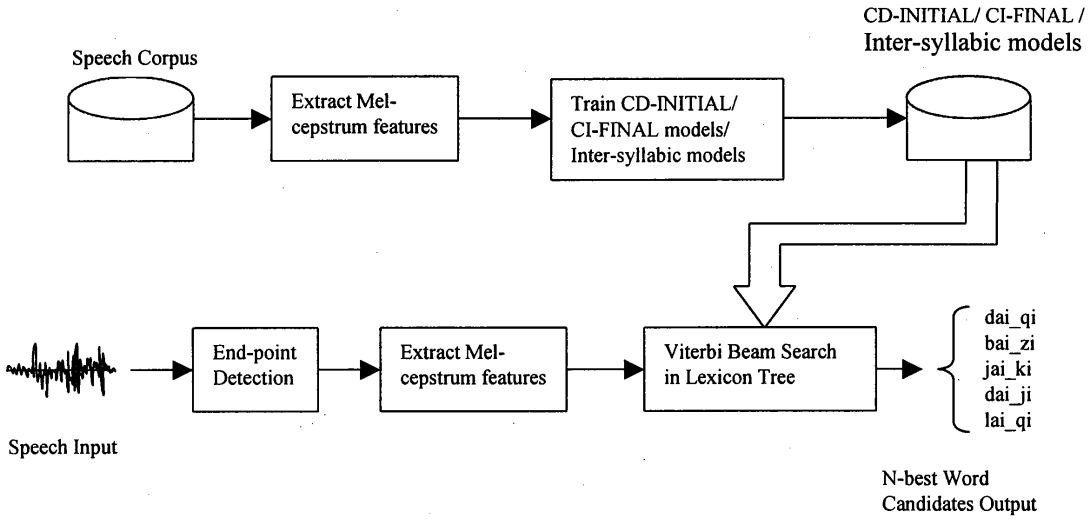
<fig.1> A sub-tree of the lexicon tree



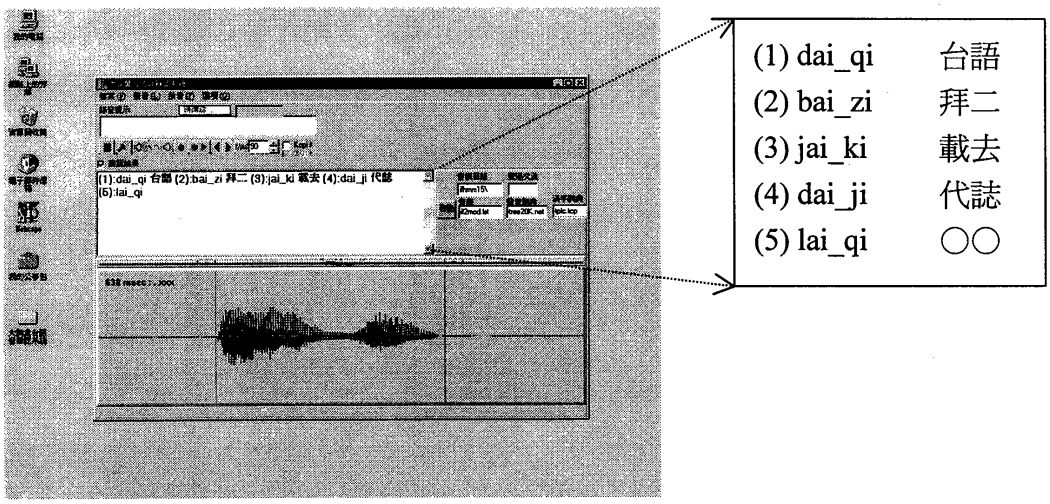
<table. 7> Experimental results for the large-vocabulary Taiwanese word Recognition by using Inside-syllable modeling v.s. Inter-syllable modeling

	Inside-syllable modeling		+ Inter-syllable modeling	
	Error Rate%	CPU time	Error Rate%	CPU time
R1000	9.5	3.19 x realtimes	7.2	3.20 x realtimes
H500	16.8		13.8	
N407	13.8		12.1	
P396	6.9		6.4	
Average	11.4	3.19 x realtimes	9.4	3.20 x realtimes

<fig 2> A prototype system running on MS-WindowNT



<fig.3> The GUI of the prototype system implemented in MS-Windows95/NT



Using Keyword Spotting and Utterance Verification to a Prank Call Rejection System

Chun-Jen Lee, Eng-Fong Huang, and Jung-Kuei Chen
李俊仁　　黃英峰　　陳榮貴

Applied Research Laboratory, Telecommunication Laboratories,
Chunghwa Telecom Co., Ltd., Taiwan, ROC
中華電信研究所 應用科技研究室

Email: {cjlee, engfong, jkchen}@ms.chttl.com.tw

Abstract

In this paper, a single keyword spotting and verification prototype system aiming at rejecting prank calls is reported. The system issues an announcement in Mandarin which instructs International Operator Direct Connection (IODC) customers to speak a keyword in Mandarin. If the system recognizes the keyword, then it switches the line to a telephone operator. If not, the call is assumed to be a prank call and the line is cut off. The underlying algorithm of this current system consists of a keyword spotter, to extract a single keyword, and a rejector, to verify whether a valid keyword or not, in each spontaneous speech utterance. The experimental results demonstrate that 97.1% of prank calls were rejected while only 2.4% of customer calls were rejected. The field-trial system was developed and has been in operation at the Chunghwa Telecom International Business Group since March 1998.

Keyword: ***keyword spotting*** ***utterance verification***

1. Introduction

Chunghwa Telecom *IODC*, a home country direct service, enables Taiwanese travelers to place a call to a Taiwanese telephone operator directly from overseas. It occupies a large portion of traffic of the overseas incoming calls to Taiwan due to its ease of use. However, over 80% of the incoming calls from some countries are prank calls that seriously compromise the quality of services. To address this problem, a prank call rejection system has been developed which is capable of automatic detecting and rejecting prank calls without connecting to a telephone operator.

During recent years, keyword spotting (Rohlicek 1989, Rose 1990, Wilpon 1990) and utterance verification (Rahim 1995, Kawahara 1997) technologies have become popular methods for domain specific speech understanding tasks. The former is capable of detecting and recognizing keywords embedded in the utterance. The latter is to reject utterances that do not contain valid keywords and utterances that have low confidence scores. An important task in keyword spotting and utterance verification is the selection of an appropriate operating point or critical threshold to provide a desirable combination of *Type I error* (false rejection) and *Type II error* (false alarm).

Present *IODC* operators report that legitimate *IODC* users usually understand Mandarin, while prank callers often do not understand nor speak Mandarin. Hence to detect a prank caller, one may instead determine whether this caller understands and speaks Mandarin. This is realized in our prank call rejection system using both keyword spotting and utterance verification technologies. Upon receiving an incoming *IODC* call, the system issues an announcement in Mandarin asking the customer to say a keyword in Mandarin. If the system recognizes the keyword in the caller's response, it then switches the line to a human telephone operator. If not, the call is determined to be a prank call and the line is cut off automatically without being transferred to the operator.

In this paper, we report a recently developed prototype system for an application of prank call rejection using keyword spotting and utterance verification. The system issues an announcement in Mandarin which instructs *IODC* customers to pronounce a keyword in Mandarin. If the system recognizes the keyword in the response, it then switches the line to a

telephone operator. If not, the call is assumed to be a prank call and the line is cut off. This system was developed based on the fact that the Taiwanese or Chinese customers will understand the announcement but prank callers probably will not. In this paper, we mainly concern with a speech recognition technology used in the system.

The remainder of the paper is organized as follows. In Section 2 of this paper, we briefly describe system concepts. Phases of the development are discussed in Section 3. In Section 4, we describe a speech recognition technology used in the system. Experiment results are reported in Section 5. Finally, some conclusions are given in Section 6.

2. Concept of the System

Prank calls to Chunghwa Telecom IODC are made by natives of foreign countries who do not understand Mandarin. On the other hand, almost all customers of the service are Taiwanese. Hence, we designed the prank call rejection system as shown in Figure 1. After the keyword " **Chunghwa Telecom** (中華電信)" is announced in the system prompts, the IODC customers are connected with the operator only by repeating the keyword. The following shows an example of dialogue between a prank caller and the system.

User: (Call up system)

System: This is Chunghwa Telecom IODC service system. You are now connected to an automatic response system. Please say "**Chunghwa Telecom**" after the beep-tone, and we will connect you with the telephone operator (beep).

User: … (The system waits for few seconds.).

System: Please say "**Chunghwa Telecom**" once more after the beep-tone (beep).

User: #%&9?!

System: Sorry! Please call again.

3. Phases of the Development

The system was developed in the phases described as follows. A trial was made of incoming calls from the top 1 country where the prank call rates had been 80-90%.

Phase 0: Two telephone speech databases were setup to train and evaluate the proposed system. The first speech database (SDB1), used for training, consists of 400 phrases and short paragraphs that are chosen from TDB and read by 60 male and 40 female speakers. The second speech database (SDB2), used for testing, consists of 340 spontaneous utterances for IODC service uttered by 7 male speakers. And, there are 164 utterances containing valid keywords in SDB2.

Phase 1: A HMM-based keyword spotter and rejector were developed and integrated into the proposed system. A two-pass strategy was adopted consisting of recognition followed by verification. In the first pass, keyword spotting was performed to detect the position and its likelihood score of the possible keyword. In the second pass, for each keyword segmentation, a likelihood score was also obtained for the corresponding anti-keyword model. A confidence score based on a likelihood ratio test was then performed and the utterance was either accepted or rejected.

Phase 2: A selection of an appropriate operating point to provide a desirable combination of Type I error (false rejection) and Type II error (false alarm) were performed in this phase using SDB2. The experimental results demonstrate that 97.1% of prank calls were rejected while only 2.4% of customer calls were rejected.

Phase 3: The field-trial system was developed and has been in operation at the Chunghwa Telecom International Business Group since March 1998. A trial was only made of incoming calls from the top 1 prank call country. If the system recognizes the keyword, then it switches the line to a telephone operator. If not, the call is assumed to be a prank call and the line is cut off. Also, all calls were collected and will be used to improve the system performance.

4. Keyword Spotter and Rejector

In the Chunghwa Telecom *IODC* service system, the core technology module is the keyword spotter and rejector. Following keyword recognition, an input utterance was segmented and labeled as keyword and non-keyword hypotheses. Besides, their corresponding positions and HMM likelihood scores are also detected and calculated by the keyword spotter. Then, the rejector will verify whether the utterance is a prank call or not.

4.1. Keyword Spotter

In Chunghwa Telecom *IODC* service application, the expected utterance usually contains at most one keyword embedded in non-vocabulary speech. We inferred that performance could be significantly improved by imposing this single keyword constraint. To achieve this, we proposed a keyword-filler network. In this modified keyword spotter, only four kinds of utterances containing one valid keyword are allowed:

Type A: A single keyword
Type B: A single keyword followed by a non-keyword speech
Type C: A non-keyword speech followed by a single keyword
Type D: A single keyword embedded in non-keyword speech in both sides

In order to generate HMM models from *SDB1*, a segmental k-means training algorithm (Rabiner 1986) is used to optimize the likelihood of the observation sequence and the state sequence over all model parameters. To reduce the likelihood computation, subsyllabic units (Chen 1994), syllable initials and syllable finals, were used as basic HMM building blocks. Each initial and final model has 3 and 5 states, respectively. Overall there are 440 states for all the subsyllabic HMMs. A left-to-right HMM scheme with no skipped states was chosen for all the models.

4.2. Rejector

As a generalization to keyword spotting, utterance verification (*UV*) attempts to reject or accept an utterance based on a computed confidence score. This is particularly useful in situations where utterances are spoken without valid keywords or when significant confusion exists among keywords which may result in a high substitution error probability. *UV* is carried out by testing the *null hypothesis* that a specific keyword exists in a segment of speech O versus the *alternative hypothesis* that the keyword is not present. Based on a likelihood ratio test, to accept or reject an utterance depends on whether the log likelihood ratio $LR(O \mid \Lambda)$ is higher than a specific verification threshold τ (here $\Lambda = \{ \lambda_t \}, \{ \lambda_a \}$). Sets of $\{ \lambda_t \}$ and $\{ \lambda_a \}$ are the models of the keyword and anti-keyword HMMs respectively.

Several different formulations for the alternative hypothesis have been proposed. Two formulations will be described in this section. The first choice is simply to use the general acoustic filler model λ_f which is keyword independent. The likelihood for the alternative hypothesis is defined as $\log[p(O \mid \lambda_f)]$. The second choice for the alternative hypothesis is to introduce a keyword-specific anti-keyword model. There are many strategies for constructing such models, such as constructing additional keyword-specific anti-keyword models or using the likelihood of all competing models, $\{ \lambda_a \}$. The likelihood for the alternative hypothesis is defined as $\log[p(O \mid \lambda_a)]$. In this paper, we will only discuss the latter type since it does not need to train additional models and is easily constructed. The confidence measure is evaluated by the log likelihood ratio

$$LR(O \mid \Lambda) = \log[p(O \mid \lambda_t)] - \log[p(O \mid \lambda_a)], \qquad (1)$$

where $\log[p(O \mid \lambda_t)]$ is the likelihood for the null hypothesis.

An appropriate operating point is selected to provide a desirable combination of *Type I error* (false rejection) and *Type II error* (false alarm). Here, we chose an operating point to minimize the total error which is defined as the sum of false rejection and false alarm errors. An utterance is rejected if the test of the log likelihood ratio

$$LR(O \mid \Lambda) < \tau, \tag{2}$$

where τ is the operation point. This enables rejection of utterances which contain non-vocabulary words or noise. A general form of the likelihood of the alternative hypothesis based on anti-keyword model could be further formulated as

$$\log[\frac{1}{2}\exp\{\eta \log[p(O \mid \lambda_a)]\} + \frac{1}{2}\exp\{\eta \log[p(O \mid \lambda_f)]\}]^{\frac{1}{\eta}}, \tag{3}$$

where η is a constant. Currently, experiments are conducted using the confidence measure function defined in equation (1) only.

5. Experiments

In order to evaluate the performance of our rejection scheme, experiments have been conducted with *SDB2* which consists of 340 spontaneous utterances for *IODC* service uttered by 7 male speakers. In *SDB2,* there are 164 utterances containing valid keywords and 176 utterances spoken without valid keywords. Figure 2 shows the two histograms for the keyword and non-keyword log likelihood ratio scores. Figure 3 shows overall system performance as a function of threshold. The FA (False Acceptance) is the rate of accepting prank calls and FR (False Rejection) is the rate of rejecting customer calls. We can control the rates of *Type I error* and *Type II error* with the threshold value. The operating point is designed to minimize the sum of false rejection and false alarm errors. The experimental results demonstrate that 97.1% of prank calls were rejected while only 2.4% of customer calls were rejected, as shown in Table 1. Figure 4 presents the ROC curve for the experiment. The underlying algorithm has very high probability of detection at very low false alarm rates, where the vocabulary size is only one. However, we also notice that as we increase the vocabulary size, the decrease in performance is evident in the experiments of the TL phone directory assistant task.

6. Conclusions

In order to reject prank calls for the Chunghwa Telecom *IODC* service, we developed the prank call rejection system using the technologies of keyword spotting and utterance verification. The system has been in operation at the Chunghwa Telecom International Business Group since March 1998. Over 90% of prank calls were successfully rejected in the first two weeks in the field-trial phase.

Acknowledgments

The authors would like to thank Dr. J. T. Wang, Director of CHT-TL, Dr. J. H. Liang and Dr. B. S. Jeng, Deputy Director of CHT-TL, for their fruitful support. The authors also would like to thank Dr. K.-Y. Chang and Dr. C.-S. Liu for their invaluable advice and timely encouragement. We also thank the colleagues of the CHT-TL speech recognition group for their carrying out a certain part of the work described in this paper.

References

Rohlicek, J. R., W. Russel, S. Roukos, H. Gish, "Continuous hidden Markov modeling for speaker-independent word spotting," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (Glasgow, Scotland), May 1989, pp. 627-630.

Rose, R. C., D. B. Paul, "A hidden Markov model based keyword recognition system," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (Albuquerque, New Mexico), April 1990, pp. 129-132.

Wilpon, J. G., L. R. Rabiner, C. H. Lee, E. R. Goldman, "Automatic recognition of keywords

in unconstrained speech using hidden Markov models," IEEE Trans. Acoust., Speech, Signal Processing, vol. 38, no. 11, pp. 1870-1878, November 1990.

Rahim, M. G., C. H. Lee and B. H. Juang, "Robust utterance verification for connected digits recognition," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing , 1995, pp. 285-288.

Kawahara, K., C. H. Lee and B. H. Juang, "Combining key-phrase detection and subword-based verification for flexible speech understanding," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, 1997, pp. 1159-1162..

Rabiner, L. R., J. G. Wilpon, and B. H. Juang, "A segmental k-means training procedure for connected word recognition based on whole word reference patterns," AT&T Tech. J., vol. 65, no. 3, pp. 21-31, May 1986.

Chen, J.-K., F. K. Soong, and L.-S. Lee, "Large vocabulary word recognition based on tree-trellis search," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (Adelaide, South Australia), April 1994, pp. II 137-140.

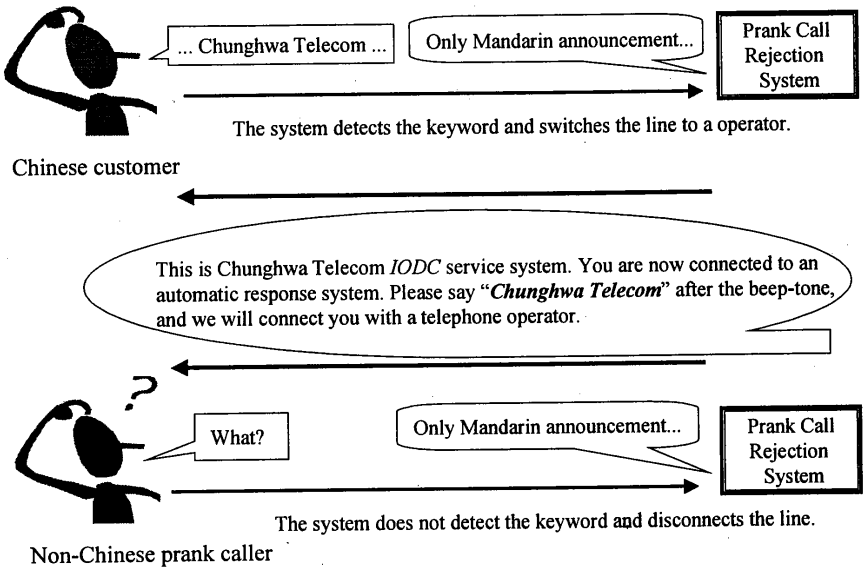


Figure 1. Dialogue between callers and a system.

	164 utterances with keywords	176 utterances without keywords
Accept	Correct acceptance	*Type II error (5 utterances)*
Reject	*Type I error (4 utterances)*	Correct rejection

Table 1.

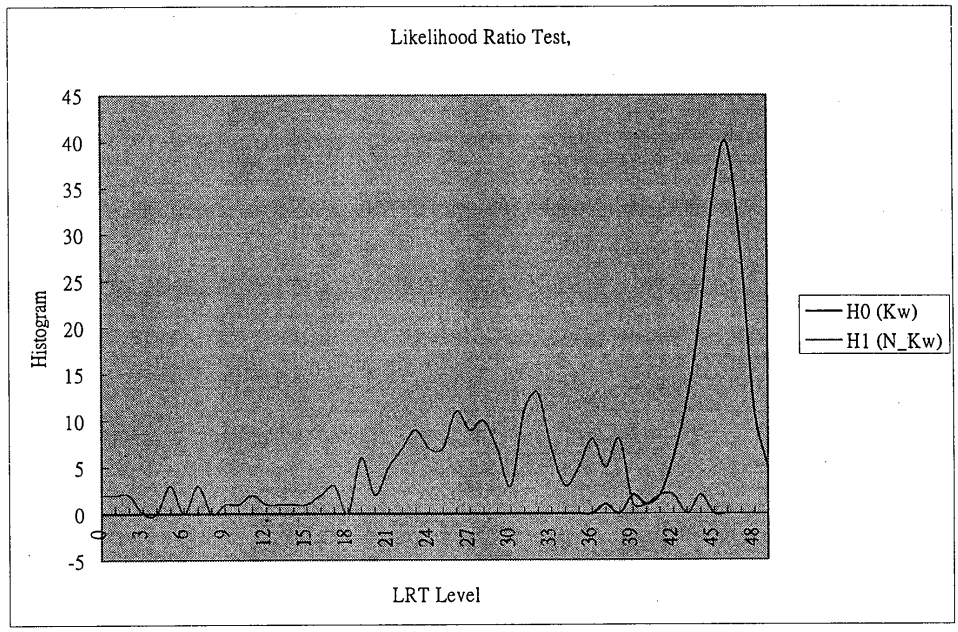


Figure 2. Histograms showing the distribution of the log likelihood ratio scores.

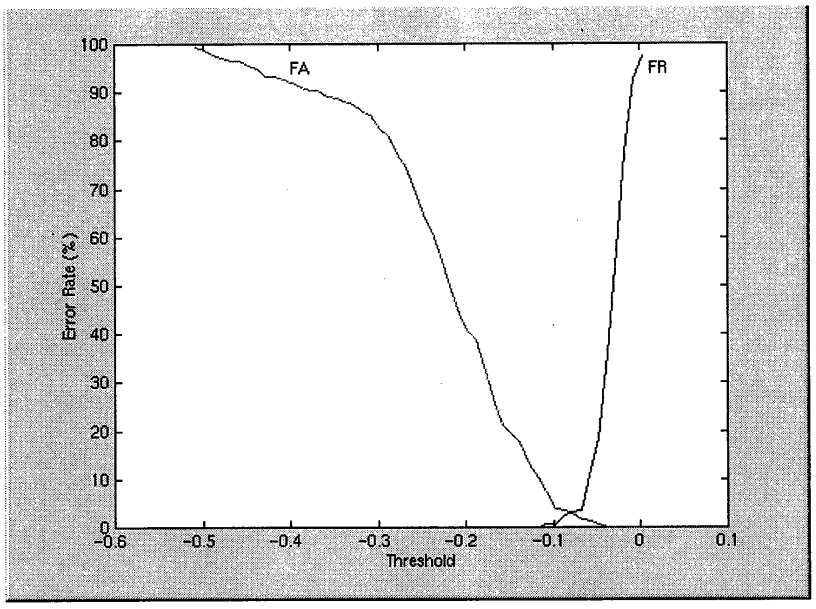


Figure 3. Error rate as a function of threshold.

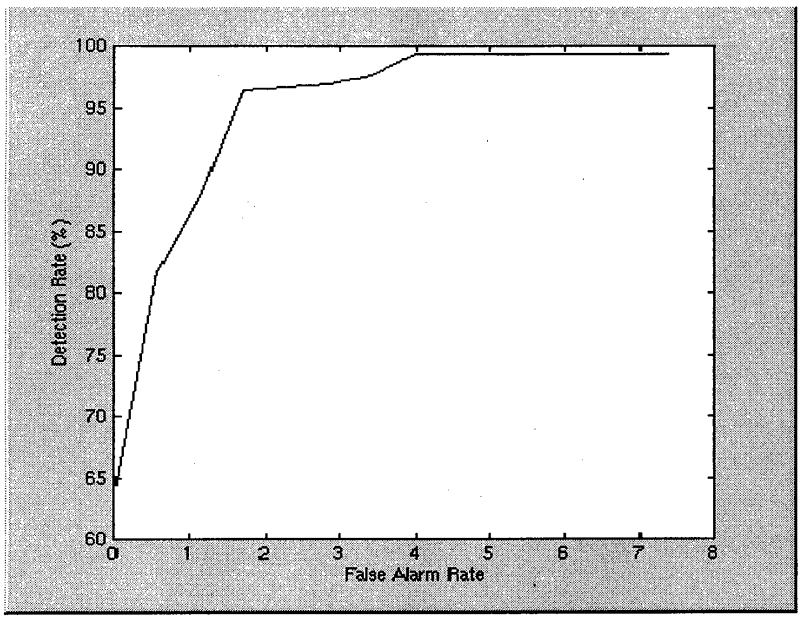


Figure 4. ROC curve.

CPAT-Tree-Based Language Models with an Application for Text Verification in Chinese

Chun-Liang Chen[1], Bo-Ren Bai[2], *Lee-Feng Chien[3] and Lin-Shan Lee[1,2,3]
[1]Dept. of Computer Science and Information Engineering, National Taiwan University
[2]Dept. of Electrical Engineering, National Taiwan University
[3]Institute of Information Science, Academia Sinica
Taipei, Taiwan, R.O.C.
E-mail: {liang, white}@speech.ee.ntu.edu.tw, {lfchien, lsl}@iis.sinica.edu.tw

Abstract

PAT tree is an efficient n-gram indexing structure. Except for text retrieval, it is believed also useful in many natural language processing applications for the construction of n-gram language models. But, an original PAT tree requires much space in memory to maintain fast speed of n-gram access and is limited to construct a large language model in practical environments. The purpose of this paper is to present an improved PAT tree structure, called CPAT tree (Compact PAT tree) for natural language modeling applications. The CPAT tree can significantly reduce the main memory requirement of original PAT trees and is found very efficient in constructing large n-gram language models. Such an advantage has been proven in OCRed-text verification and will be also introduced in this paper.

1. Introduction

PAT tree is an efficient n-gram indexing structure [Frakes 1992]. Except for text retrieval, it is also useful in many other applications, for example topic classification, spelling error detection and correction, DNA sequence search [Ricardo 1992], or even Markov n-gram language models. Markov n-gram language models were frequently used in many natural language processing applications, such as speech recognition, OCR, input methods, etc. Due to the considerations of memory space and computational complexity in practical implementation, conventional models are often an approximation, e.g., bi-gram or tri-gram models. Natural language processing systems based on such an approximated model cannot be robust enough. PAT-tree-based n-gram indexing was found very efficient to construct high-order n-gram language models in our previous work [Chien 1997]. Nevertheless, an original PAT tree requires much space to store the whole tree in the internal memory to maintain fast speed of n-gram access. It will not as efficient when the PAT tree is too large to be loaded into

memory. To run a large language model using the PAT tree indexing, it needs other improvements. The purpose of this paper is to present an improved PAT tree structure, called CPAT tree (Compact PAT tree). The CPAT tree can significantly reduce the main memory requirement of original PAT trees and is found very efficient in constructing large n-gram language models.

The CPAT tree is extended from original PAT tree. It separates a PAT tree into memory part and disk part. The memory part is basically a linear transformation of original tree structure. Pointers for tree traverse are no longer required. On the other hand, the disk part is primarily the recorded information such as string contents, frequency values etc, which are removed from main memory to release more space for allocating larger models.

The first application performed by using the CPAT trees is the experiment on OCRed-text verification in Chinese. It needs to note that the concept of "text verification" has a little difference from conventional "spelling checking". Since there is no explicit delimiters as a marker of word boundary in Chinese and some other Asian languages, the function of "text verification" in Chinese is primarily to check the validity of a text at the context level rather than word level as found in conventional English spelling checkers. The text verification problem is therefore defined to verify the validity of an arbitrary text string, including detect various input errors, e.g. speech recognition errors, typing errors, OCR errors, etc., and correct them automatically.

Primary methods for Chinese text verification (or text error checking) can be divided as dictionary lookup [Shr 1992, Liu 1997], n-gram analysis [Shr 1992, Chang 1994, Xia 1996] and parsing. The dictionary lookup approach needs to face with the word segmentation and rigid dictionary collection problems. The n-gram analysis approach, instead, relies much on the adopted bi-gram or tri-gram models. As to the parsing approach it is seldom found because it is unable to perform effective sentence parsing in Chinese texts at present. Although all of these methods can combine with morphological rules or heuristics about similarity in shapes, pronunciations, meanings or input keystrokes between similar characters for advance processing [Shr 1992, Chang 1994, Liu 1997], it is believed the text verification problem has a lot of space to improve. The CPAT-tree-based approach is proposed of this purpose. The proposed text verification process functions like spelling checking as in commercial word processors, but with high degree of differences in the used technology. Instead of detecting errors primarily at the word level as was done in conventional spelling checkers, global analysis up to the sentence level can be handled in the proposed text verification technique by means of a CPAT-tree-based large-scale language model and sophisticated text searching skills. As found in our experiments on verifying OCRed texts, the

proposed CPAT approach compared with the above methods is more competitive, if the adopted CPAT tree is trained with a sufficient corpus and more effective models continually developed.

2. CPAT Tree

PAT tree and PAT array are two well-known and frequently-used data structures for n-gram indexing in text retrieval. PAT tree is based on PATRICIA algorithm [Morrison 1968] for indexing every possible position in a continuous data stream. Each indexing point of interest is called a semi-infinite string or different suffix. PAT array [Clark 1996] is another compact representation of PAT tree. It can be considered as a sorting collection of all external nodes of PAT tree. For large text retrieval, there have been many previous researches about finding an efficient n-gram indexing data structure to take both time and space into consideration. P. Ferragina [Ferragina 1996] proposed a text indexing structure for secondary storage, which is called SB-tree, that combines the B-tree and suffix arrays. E. F. Barbosa [Barbosa 1995] proposed an optimized algorithm to improve the retrieval time of the indirect binary search in PAT array. In Sato's paper [Sato 1997], a new data structure called TS-file (Tree Structured file) and a set of algorithms were proposed to make arbitrary string retrieval especially fast. In addition, M. Shishibori [Shishibori 1997] designed a compact data structure for digital search trie and introduced a hierarchical structure in order to improve the efficiency of large registered keys retrieval. The compact concept proposed in the CPAT tree is similar to that in Shishibori's work, but the main difference is that Shishibori uses binary search tree (the terminal node stores the registered keys) while the CPAT tree proposed is a binary search trie (each node can be a terminal node or non-terminal node). Furthermore in CPAT it is added a "booster" data structure for search speedup.

2.1 PAT Tree Data Structure

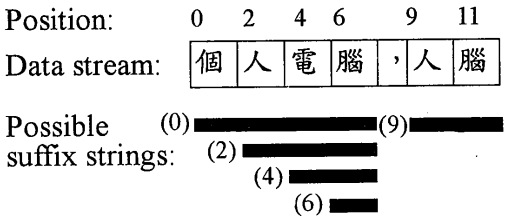
The proposed CPAT tree is extended from original PAT trees. The superior features of the PAT tree data structure mostly come from its ability to perform fast full-text indexing and searching. Using this data structure to fully index the documents, all possible words or character strings, including their frequency counts in the documents, can be updated and retrieved in a very efficient way. Besides, the data stream to be indexed can be any type of information, including part-of-speech strings, phone strings or other strings depending on applications.

For convenience of description, an example data stream with two sentence fragments "個人電腦" and "人腦" is shown in Fig.1(a), in which the "Position" above the "Data

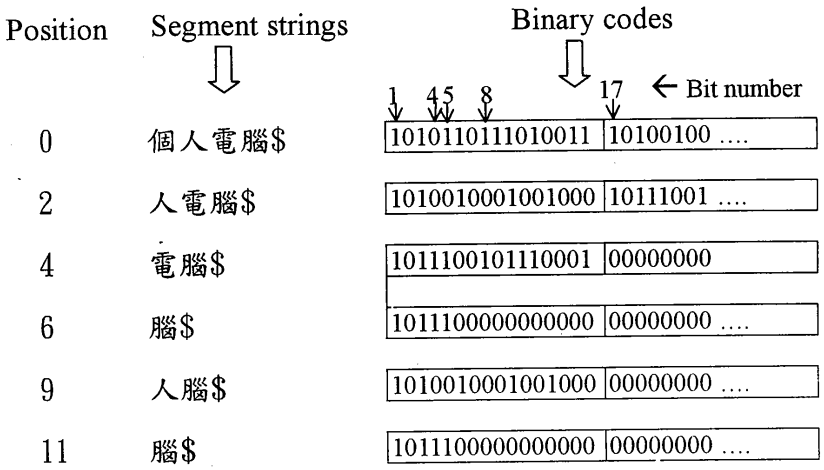
stream" means the real byte offset of the indexing points and the "Possible suffix strings" marks the 5 unique suffix strings, i.e., "個人電腦", "人電腦", "電腦", "腦", "人腦". Fig.1(b) also shows the corresponding binary bit streams of each indexed suffix string. In the processing, all of the indexed suffix strings are appended an marker "$" to identify the ending. Besides, Fig.1 (c) shows the physical representation of corresponding PAT tree, in which each node represents a unique suffix string and is associated with four-tuple of information including "comparison bit number", "frequency count", "accumulated frequency count" and "data position". The "comparison bit number" is used to indicate the bit number needs to compare and decide the left or right way to go when traversing at this node. The "frequency count" is the number of total frequency value of the indexed suffix string occurring in the data stream. The "accumulated frequency counts" stands for the sum of frequency counts of the total nodes in the sub-trees. The data position is the pointer to the data stream.

The detailed steps of the construction of PAT tree are ignored here. It is based on the process of binary search trie insertion. When a node pointed to a suffix string is inserted into a PAT tree, it will be inserted into the neighborhood of the node with a longest bit stream similarity and will be tagged with the minimal comparison bit to discriminate them. If the newly inserted suffix string has been registered in PAT tree, only the frequency counts of the representing node will increase but no extra node is needed. Just like the example in Fig.1 (c) , the node C is used as a shared indexing node for pattern "腦" both in "個人電腦" and "人腦" in the corpus. The total number of nodes in PAT is exactly equal to the total number of unique or distinct suffix strings in the corpus. The advantage of the PAT tree structure is useful for speeding up searching. For illustration, the full traverse for searching for the pattern "電腦" with the pre-constructed PAT tree is demonstrated here. At first, we encode the character string "電腦" as its binary representation (BIG5 code) "1011100101110001....". The searching process will start from the root A. At the first step, the default branch to go is left because the root is a dummy node. At this time, it will stop at the node B and the comparison bit to check is bit 4. To take a look at 4th bit of "電腦", it is 1. So, it takes the right branch to go. Now it will stop at node C and check the 8th bit as indication. It is 1 as usual and has a right branch to go also. By the right branch, it will return back to the node B and find the comparison bit changed from 8 to 4. In the PAT tree, when the examining comparison bit is lower than the previous one, it means the branch is an upper link or the destination node is an external node. At last, by the data position pointer of node B, the destination suffix string "電腦" will be extracted from the data stream, and, after making a string comparison , it can be proven that the examining string "電腦" appears in the data stream. From the associated information, it also knows that "電腦" occurs one time and the accumulated frequency count is 6.

(a)



(b)



(c)

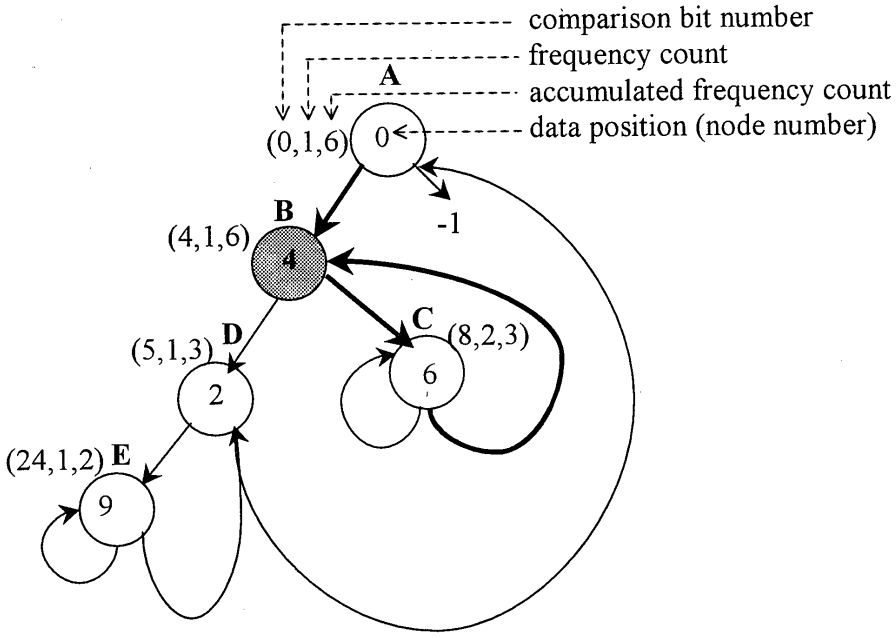


Fig.1 : PAT tree data structure

2.2 CPAT Tree Data Structure

PAT tree and PAT array are compromise in their features. The advantages of PAT tree are easy for update and fast for search. But, an original PAT tree requires much space in the internal memory to maintain fast speed of n-gram access. On the other hand, in PAT array the sorted array is usually stored in disk and indirect binary search performed for text retrieval. PAT array is flexible to index a huge data stream. But the random access of the pointers in disk slows down the searching speed. CPAT tree is developed to find out a tradeoff between PAT tree and PAT array.

Before constructing a CPAT tree, its original PAT tree must be built as a temporal media at the first stage. Then the transformation process will be performed to compress PAT tree into CPAT. The original PAT tree is separated into two parts: the memory part (RamPart) and the disk part (DiskPart) in CPAT as in Fig. 2. The RamPart is basically a linear transformation of the original tree structure. Pointers for tree traverse are no longer required. On the other hand, the DiskPart is primarily the recorded information such as string contents, frequency values etc, which are removed from main memory to release more space for allocating larger models. The whole CPAT tree looks like an iceberg. The RamPart can be viewed as the top of CPAT or iceberg. The DiskPart is that under water, which is often several times larger than the RamPart above water. This kind of indexing structures can release space in memory and afford larger or multiple indexing trees.

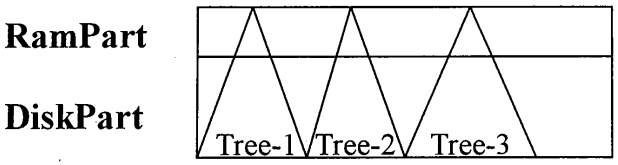


Fig. 2: Multiple CPAT trees

The RamPart should be loaded into memory before the searching starts. The RamPart consists of TreeMap and BitMap as in Fig.3. The TreeMap is a preorder mapping representation from 2-D PAT tree to one dimension of binary sequence in which '0' means internal node and '1' means external node. A node is referred to as an external node only when its comparison bit is equal to or greater than that of its parent nodes. At the same time, the BitMap is a linear array which stores the corresponding comparison bit number in sequence for each node. The stream in the BitMap will be aligned and packed with that in the TreeMap as a two-byte sequence for the sake of memory saving.

As for the DiskPart, it stores some useful information of the indexed nodes, including frequency counts, accumulated frequency counts and pointers to starting address of the indexed suffix in the data stream. Only the information of the external nodes need to save with the sequence of external nodes in TreeMap.

The left and right branches of original nodes in PAT tree have been eliminated in CPAT for saving space. The problem arising here is how to reach the left and right child for each node in TreeMap. The left child is not hard to find since in preorder sequence the left child is just the next node in TreeMap, but the right child is not so natural to examine. It should be reached based on a property of common tree structure that "the number of external nodes is exactly greater by one than that of internal nodes in a binary tree".

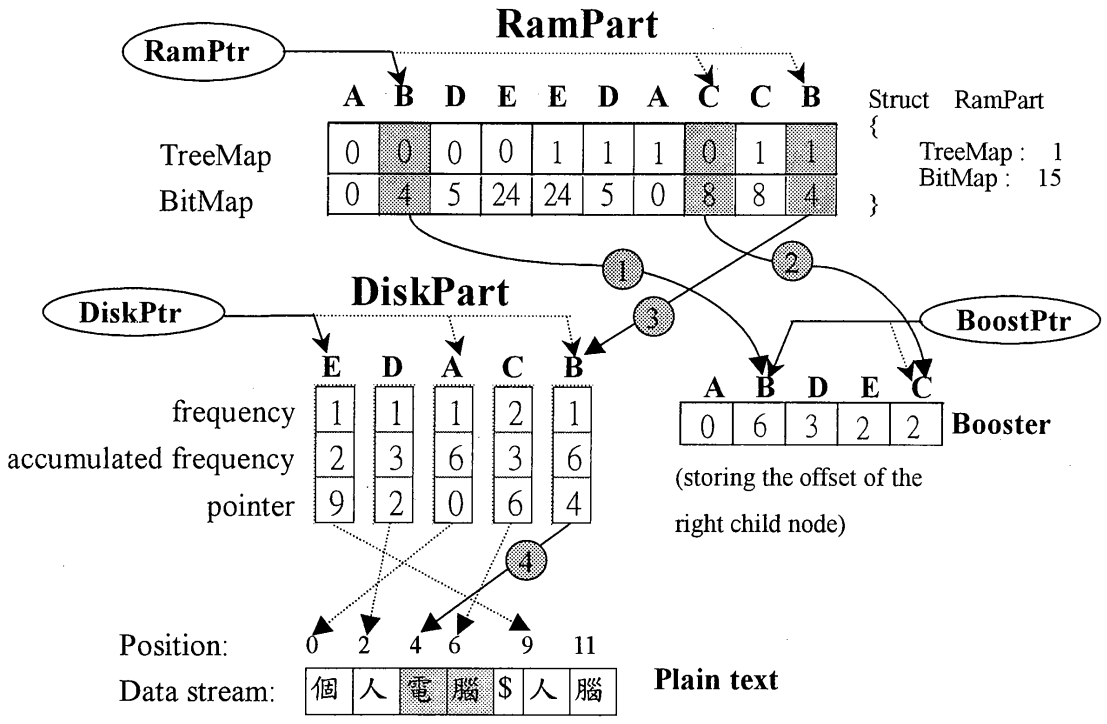


Fig. 3: Data structure content of CPAT for PAT tree of Fig. 1.

To jump from a certain node to its right child node, it should stride the overall left sub-tree . So until it jumps a sequence of bit streams in the TreeMap such that number of '0' bits is greater than that of '1' bits by one, we exactly get the right child node. Because the sequential scanning in binary bit streams will cost much CPU time, another linear array called "booster" is used here for storing the forward offset of right child node for each internal node in TreeMap. The function of "Booster" is to speedup the search.

We will use the same example for illustration with the CPAT tree. The total searching process will be divided into 4 steps with grayed and numbered circles shown in Fig.3. Besides, there are three pointers named "RamPtr","DiskPtr" and "BoostPtr" are used to indicate the current positions in the corresponding arrays during searching.

At the first step, we still ignore the dummy root node and directly go to node B (the second cell in TreeMap). So the RamPtr will point to B at this moment. According to BitMap, we check the 4th bit of binary stream of "電腦" and find it is '1'. Since node B is marked as '0' in TreeMap, it is an internal node as definition. Shift the BoostPtr to the corresponding node B and get the offset '6' for forwarding to next internal node C. At this time, the RamPtr points to node C in RamPart and pointer BoostPtr moves to next internal node C in Booster. As for the DiskPtr, it will move to node A since there are three external nodes (E, D, A) been passed from node B to node C.

Then second step, like the first step, it will check the 8th bit of binary stream of "電腦" and find it is still '1'. By the BoostPtr, the RamPtr will move forward 2 elements to reach the right child of internal node C. At the same time, it passed through 2 external nodes (C,B), so the DiskPtr also moves forward 2 elements and gets node B.

Now, at the third step, we find the element in TreeMap is marked '1', which means that the destination node has been reached. The whole record about node B will be therefore retrieved through the DiskPtr. The last step is just retrieving the string content of node B and making a comparison with the examining string "電腦".

2.3 Performance Evaluation of CPAT tree

This section shows the obtained results on both time and space tests with CPAT and PAT tree. The tests were performed under the following environment: PII-266 PC, NT workstation 4.0 and Quantum SCSI disk.

Some theoretical values of space needed by PAT tree, CPAT and PAT array are listed in Table 1 for reference. If the data stream, i.e., the Chinese text for indexing, is n bytes (n/2 characters), the theoretical space needed by PAT tree is O(9n)~O(10n), PAT array is O(5n) and CPAT is O(7n), as our observation. In this table, the RamPart size of CPAT doesn't include the "Booster" size, since it depends on if it is loaded into memory. If it is, the memory space will expand to O(2n) but n-gram access will be accelerated.

Strategy	Text Size	PAT tree	CPAT tree	PAT array
RamPart	0	9n~10n	n	0
DiskPart*	n	0	6n**	5n
Total	n	9n~10n	7n	5n

Table 1: Space usage comparison (theoretical values) in which "*" indicates that the DiskPart includes size of plain text (n), pointers (2n), counts (n) and frequencies (n), and "**" that includes booster with size n.

Then, the practical time and space needed by CPAT and PAT trees are listed in Table 2. The corpus used for testing are as follows:

Test1-O(1K) :[三字經] full text
Test2-O(10K) :[中華民國憲法] full text
Test3-O(100K) :[清靜經][道德經][金剛經][心經][六祖壇經][大學][中庸] full text
Test4-O(1M) :[紅樓夢] full text
Test5-O(10M) :[金庸小說] full text
Test6-O(20M) :[1997 中央社上半年新聞] full text
Test7-O(100M) :[1997 中央社全年新聞] full text

The obtained average ratio value (the space needed with respect to original text size as 1) in the last row in Table 2, is lower to almost 50% as compared with the theoretical value in Table 1. Although the theoretical DiskPart space is O(7n), in real test only O(3.7n) space requirement is required. This is because there exist many repeated suffix strings in the indexed data stream. The repeated suffix strings will not take space to store but only updates the frequency counts. Table 2 also shows the real time spent for constructing various sizes of PAT and CPAT trees.

CORPUS ID	SPACE (KB)				Time(sec)	
	Corpus Size	PAT	CPAT		CPAT	CPAT
			Ram	Disk	Construction	Transformation
Test1-O(1k)	3	23	4	15	0.03	0.30
Test2-O(10k)	22	122	21	79	0.27	0.13
Test3-O(100k)	111	544	100	260	1.16	0.49
Test4-O(1M)	1,791	10,839	1,885	7,069	35.02	18.13
Test5-O(10M)	12,013	64,984	11,182	42,618	272.01	413.43
Test6-O(20M)	19,541	82,779	14,587	53,604	508.87	680.93
Test7-O(100M)	107,333	439,087	107,771	397,447	2381.00	3214.35
Ratio for Test7	1	4.09	1.04	3.7	1	1.35

Table2: Time and space comparison between PAT and CPAT.

Furthermore, Table 3 shows the tests on n-gram access speed with PAT and CPAT trees. The indexed data stream is "金庸小說" (about 12MB) and the examining n-gram strings are automatically generated by keyword extraction from the data stream. The number of keywords for each n-gram is shown in Column 2. The time unit for speed measurement is second here. It is clearly to see that the time spent on CPU, hard disk and totally needed. In "Average" column, it shows the capability of how many n-grams can be accessed per second with PAT and CPAT respectively. The last column "Speed Ratio" means the ratio of CPAT over PAT in "average" column. It's obvious that disk access time always dominates the total access time. Although the achieved access speed with the CPAT tree is slower than that with the PAT tree in main memory, it was found fast enough in many natural language processing applications.

Length	# keyword	PAT	CPAT			Average		Speed Ratio
		CPU	CPU	HD	Total	CPAT	PAT	
2-gram	5047	0.170	0.400	30.405	30.805	164	29688	181.21
3-gram	2221	0.080	0.190	8.472	8.662	256	27763	108.28
4-gram	3447	0.160	0.251	7.430	7.681	449	21544	48.01
5-gram	678	0.100	0.080	1.082	1.162	583	6780	11.62
6-gram	414	0.030	0.050	0.541	0.591	701	13800	19.70
7-gram	183	0.020	0.040	0.180	0.220	832	9150	11.00
8-gram	20	0.000	0.000	0.020	0.020	1000	N/A	N/A
9-gram	7	0.000	0.000	0.010	0.010	700	N/A	N/A

Table 3 : Tests of N-gram access speed with PAT and CPAT trees.

3. OCRed-Text Verification

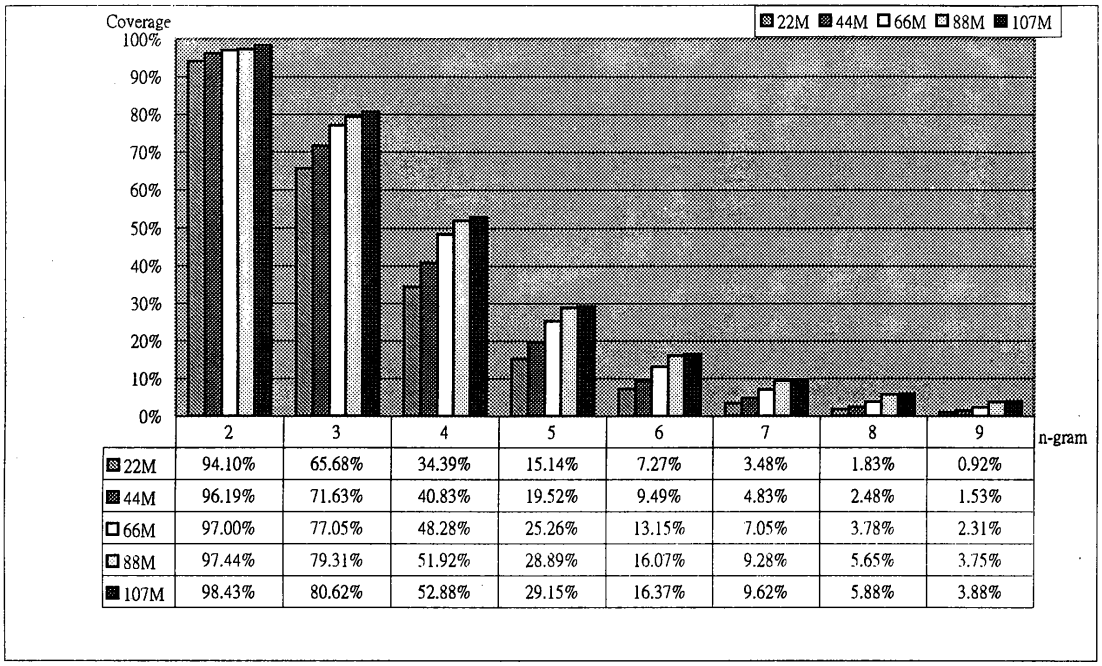
Optical Character Recognition (OCR) has been widely used for entering printed texts into computers especially for languages like Chinese, for which the complicated characters make it difficult to enter the texts through keyboards. But because such OCR processes always produce some errors, manually verifying the entered characters becomes very time-consuming. It is therefore highly desired to do such verification automatically by machine. Since it is easy to search for arbitrary character string patterns and their frequency counts in a CPAT tree, any such pattern in the entered text which has never appeared in a large text collection, or in the corresponding CPAT tree, will very possibly represent an OCR error. As a result, we can simply check the existence and frequency counts of any such character string patterns of the entered texts with the CPAT trees constructed previously to detect the errors.

This is a very attractive application of the CPAT-tree-based language models mentioned here, because errors always occur in any text entering methods, regardless of whether it is OCR, handwriting recognition, speech recognition, or even keyboards.

For Chinese and some other Asian languages this is similar to the spelling checking problem in western language, but with much higher degree of difficulties. In western language the words are well defined so simply checking the spelling of each word with a lexicon will give most of the spelling errors, but in Chinese or some other Asian languages, as mentioned before, there are no explicit word boundaries in the texts and no commonly accepted lexicon can be used in such checking processes. Therefore, instead of detecting errors primarily at the word level as was done on western languages, global analysis up to the sentence level have to be performed to handle texts without explicit word boundaries. With the approach proposed here, the full-text indexing functions given by the CPAT trees can provide the desired solution and avoid the need to use other sentence level knowledge such as grammar rules and syntactic structures. Here we'll simply use such an OCR output verification problem to test the feasibility of the proposed CPAT-tree-based language modeling techniques.

In the tests, each sentence of the OCR output is first segmented into all possible character string patterns, then each of these patterns is fed to the CPAT trees to check its existence and extract its probability (normalized frequency counts) to appear in the CPAT trees. In other words, if a sentence consists of N characters, then there are totally N*(N-1)/2 variable length patterns should be examined. In this way it is very easy and efficient to identify the character string patterns with recognition errors if it is not covered in the n-gram examining process. Actually, the power of error detection is proportional to the coverage rate of variable n-gram in the corrected testing data. The coverage rate is the percentage of total n-grams in corrected testing data that appear in the CPAT trees. The all coverage rates for n changes from 2 to 9 are shown in Fig. 4. As the corpus size increases from 22MB to 107MB, the bigram coverage rate also increases from 94.10% to 98.10% and trigram increases from 65.68% to 80.62%. Even for 5-gram, the coverage rate is approaching 30% under 107MB corpus. It is believed that the higher n-gram coverage rate, the more robust or reliable the n-gram language model is. We even can have an assumption that as the training corpus grows up to a huge size, there is no need to smoothing. If a n-gram never appears in a huge corpus, then the probability that it is an error is very large. OOV (out of Vocabulary) is another exceptive phenomenon. In addition to the statistical information extracted from the CPAT trees, some heuristic rules are found to be very helpful and integrated into the verification process as well. One example rule is that longer patterns can be considered valid if it is found to appear in the CPAT trees even with relatively lower probabilities or frequency counts, while shorter patterns need higher probabilities or frequency counts in the CPAT trees to

support its validity. This is certainly due to the fact that a longer pattern itself provides linguistic information with higher reliability.



	2	3	4	5	6	7	8	9
22M	94.10%	65.68%	34.39%	15.14%	7.27%	3.48%	1.83%	0.92%
44M	96.19%	71.63%	40.83%	19.52%	9.49%	4.83%	2.48%	1.53%
66M	97.00%	77.05%	48.28%	25.26%	13.15%	7.05%	3.78%	2.31%
88M	97.44%	79.31%	51.92%	28.89%	16.07%	9.28%	5.65%	3.75%
107M	98.43%	80.62%	52.88%	29.15%	16.37%	9.62%	5.88%	3.88%

Fig. 4: Variable n-gram coverage rates with respect to the different sizes of corpora used to construct the CPAT trees.

The testing set used here was taken from a section of printed Chinese newspaper, including 469 Chinese sentences with a total of 5,394 characters. This section of newspaper was recognized by a commercially available Chinese OCR system. The output of this OCR system is taken as the input of the OCR verification approach here. It was found manually that 131 characters among the total of 5,394 characters were incorrectly recognized. So the accuracy of the commercially available OCR system used in the test is 97.57%, and the purpose of the test here is to detect these 131 recognition errors. The results of the test are listed in Table 4 and plotted in Fig. 5 in terms of the recall rates (percentage of the 131 manually determined errors being identified correctly) and precision rates (percentage of the automatically identified errors being among the manually determined errors) with respect to different sizes of the 107MB corpora used to construct the multiple CPAT trees. In fact, the corpora used here to build the CPAT trees stem from CNA (Central News Agency) electronic news in different subject domains. It can be found from Table 4 and Fig. 5 that when the corpus size is increased from 22MB (1/5 of the whole corpus) to 107MB, the precision rate was improved significantly from 57.52% to 70.53%, while the recall rate was at the same time reduced somewhat from 67.18% to 60.30%. Such results are intuitively reasonable. A larger

corpus provides better precision performance, since more character string patterns can be observed in a larger corpus and included in the CPAT trees, thus less correct patterns will be incorrectly detected to be OCR recognition errors in the verification processes. On the other hand, when a larger corpus was included in the CPAT tree, more OCR recognition errors may be considered to be correct when the error patterns can be found as parts of some valid character string patterns in the corpus, therefore the recall rate is inevitably degraded. When measuring the system performance by the average of the precision and recall rates, the *averaged precision-recall* APR=(Precision+Recall)/2, it can be found that the APR is improved from 62.35% to 65.42% when the corpus size is increased from 22MB to 107MB. On the other hand, the same OCR testing data are also input into IBM SmartSuit'97 (a commercial word processor with Chinese text error checking functionality) to take a comparison. The results for error detection are recall 68.94%, precision 22.75% and thus APR is 45.85%.

Corpus size	22 MB	44 MB	66 MB	88 MB	107 MB
Recall(%)	67.18	64.12	61.83	61.07	60.30
Precision(%)	57.52	64.62	68.64	69.57	70.53
APR	62.35	64.37	65.24	65.32	65.42

Table 4: The error detection performance for the OCR output verification test with respect to the different sizes of corpora used to construct the CPAT trees.

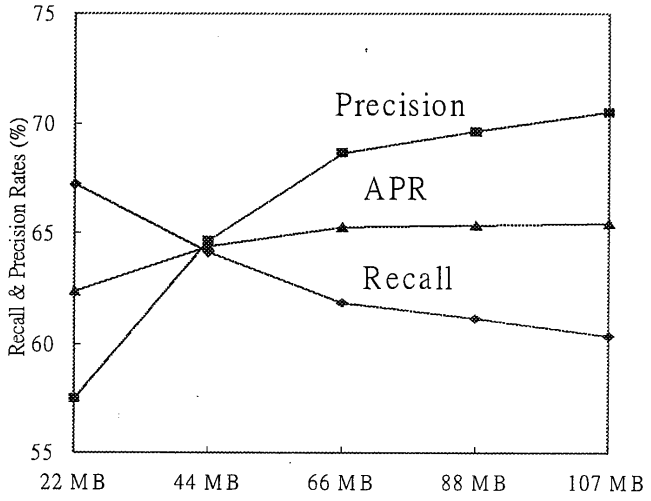


Fig. 5: The recall and precision rates for OCR output error detection with respect to the different sizes of corpora used to construct the CPAT trees.

These results are in fact significantly better then the recent results of 58% of APR obtained with a much more complicated rule based approach [Liu 1997], while in comparison here only very simple string searching techniques were used to perform the error detection. These initial results are very encouraging, and it is believed that further work will produce better performance.

4. Conclusion

The proposed CPAT data structure makes it feasible to build n-gram indexing on a large corpus and fully makes use of memory and secondary storage. It inherits both merits of PAT tree and PAT array to alleviate the memory requirement and reach a modest n-gram access speed between PAT tree and PAT array. Some initial experiments were performed to test the feasibility in OCR output verification by using a large-scale n-gram language model, which takes the CPAT tree as the core working structure. The initial result is very encouraging.

Reference

Frakes and Baeza-Yates, "Information Retrieval: Data Structures & Algorithms", Prentice-Hall, 1992.

Ricardo A. Baeza-Yates, "Text Retrieval: Theory and Practice", Information Processing 92, Vol. I, pp. 465-483,1992

L.F. Chien et al., "Internet Chinese Information Retrieval Using Unconstrained Mandarin Speech Queries Based on a Client-Server Architecture and a PAT-tree-based Language Model", Vol. 2, pp.1155-1158, ICASSP'97

Morrison, D., "PATRICIA: Practical Algorithm to Retrieve Information Coded in alphanumeric", JACM, PP.514-534,1968

Clark, D.R., and Munro, J. I. "Efficient Suffix Trees on Secondary Storage", ACM-SIAM Symposium on Discrete Algorithm ,1996

Paolo Ferragina and Roberto Grossi "Fast String Searching in Secondary Storage: Theoretical Developments and Experimental Results", ACM-SIAM Symposium on Discrete Algorithm, 1996

E. F. Barbosa, G. Navarro, R. Baeza-Yates, C. Perleberg, and N. Ziviani "Optimized binary search and text retrieval", In Algorithm- ESA'95, Third Annunal European Symposium, pp. 311-326,Greece, September 1995

T. Sato, "Fast Full Text Retrieval Using Gram Based Tree Structure", proceedings of 17-th International Conference on Computer Processing of Oriental Languages, pp.572-577, ICCPOL'97,1997

M. Shishibori, K. Morita, K. Ando and J.-I. Aoe, "The Design of a Compact Data Structure for Binary Tries", proceedings of 17-th International Conference on Computer Processing of Oriental Languages, pp.606-611, ICCPOL'97,1997

D. S. Shr et al. "A Statistical Method for Locating Typo in Chinese Sentence", Computer and Telecommunication, August pp19-26,1992

Chao-Huang Chang "A Pilot Study on Automatic Chinese Spelling Error Correction", Communication of COLIPS, Vol. 4, No 2, pp143-149, 1994

Y. Xia, X. G. Chang, S. P. Ma, X. Y. Zhu and Y. J. Jin "Co-occurrence Probability Between Chinese Characters" Communications of COLIPS, Vol. 6, No.1, pp.19-23, JUN 1996

Yuhsiang Liu, Zhili Guo, Chiching Hsu, Shauyi He and Naipo Lee, "Checking Chinese Text Errors in the Unicode Environment", 11th International Unicode Conference and Global Computing Showcase, San Jose, CA., Sep. 1997

Corpus-based Evaluation of Language Processing Systems Using Information Restoration Model

Chao-Huang Chang
E000/CCL, Building 51, Industrial Technology Research Institute
Chutung, Hsinchu 31015, TAIWAN, R.O.C.
changch@e0sun3.ccl.itri.org.tw

Abstract

In the recent years, several standard Chinese corpora, such as NUS's PH corpus and Academia Sinica's sinica corpus version 1.0, 2.0 have been released to the academia. These corpora are useful not only for training and testing corpus-based NLP systems, but also for objective evaluation of the systems. In this article, we present a noisy channel/information restoration model for automatic evaluation of NLP systems. The proposed model has been applied to two common and important problems related to Chinese NLP for the Internet: the 8-th bit restoration of BIG-5 code through non-iso8859-1 channel, and GB-BIG5 code conversion. Sinica Corpora version 1.0 and 2.0 are used in the experiment. The results show that the proposed model is useful and practical.

1. Introduction

In 1992 (Chang 1992), we proposed a concept of *bidirectional conversion,* using corpora for automatic evaluation of accuracy of syllable to character conversion systems. After that, the concept was extended to an adaptation mechanism for these systems, which has stimulated some following researches (Chen and Lee 1995). In the recent years, several standard Chinese corpora, such as NUS's PH corpus (Guo and Lui 1992) and Academia Sinica's sinica corpus version 1.0 (Huang *et al.* 1995), 2.0, and 3.0 have been released to the academia. These corpora are useful not only for training and testing corpus-based NLP systems, but also for objective evaluation of the systems. In this article, we present a noisy channel (Kernighan *et al.* 1990, Chen 1996)/information restoration mode l for automatic evaluation of NLP systems. The proposed model has been applied to two common and important problems related to Chinese NLP for the Internet: the 8-th bit restoration of BIG-5 code through non-iso8859-1 channel, and GB-BIG5 code conversion. Sinica Corpora version 1.0 and 2.0 are used in the

experiment. The results show that the proposed model is useful and practical.

Internet and World Wide Web are very popular in these days. However, computer and network are not designed for the coding of huge number of Chinese ideographic characters, since they are originated in the western world. For example, the popular ASCII code is a seven-bit standard, and a byte only has eight bits. Obviously, they can not encode the thousands of Chinese characters in a natural way. The situation is worsened due to the political separation of the Chinese Mainland and Taiwan. The Mainland and Taiwan use different styles of Chinese characters (simplified in the Mainland and traditional in Taiwan), and also invent different standards for Chinese character coding. This situation has caused several serious problems in Chinese information processing on the Internet (Guo 1996). In order to fit in different Chinese environments, usually more than one version of web pages are provided, one for English, and the others for Chinese. Chinese versions of web pages are encoded in either BIG5 (Taiwan standard) or GB (Mainland standard). Furthermore, Unicode version would become popular in the near future. In this paper, we will deal with two of Chinese processing problems on the Internet: the 8-th bit restoration of BIG-5 code through non-iso8859-1 channel, and GB-BIG5 code conversion.

BIG-5 code is one of the most popular Chinese character coding schemes used in computer network. It is a double-byte coding, the high byte ranges from (hexadecimal) A1 to FE, 8E to A0, and 81 to 8D; and the low byte from 40 to 7E, and from A1 to FE. The most and secondary commonly used Chinese characters are encoded in A440 to C67E, and C940 to F9D5, respectively; the other ranges are for special symbols and used-defined characters.

In the Chinese mainland, the most popular coding for simplified Chinese characters is GB2312-80, also called GB Code. It is also a double-byte coding, the coding ranges for high byte and low byte are the same, (hexadecimal) A1 to FE.

In most international computer networks, the electronic mails are transmitted through 7-bit channels (so called non-iso8859-1). Thus, if messages coded in BIG5 are transmitted without further encoding (using tools like *uuencode*), the receiver side will only see some *random code* messages. In the literature, little work can be found in studying this problem. S.-K. Huang of NCTU (Hsinchu) designed a shareware called Big5fix (Huang 1995), which is the only previous solution we can find for solving the problem. The input file for Big5fix is supposed to be 7-bit file. Big5fix divides the input into regions of two types: English Region and Chinese Region. The characters in the Chinese regions are reconstructed based on

collected character unigrams, bigrams, trigrams and their occurrence counts. Huang estimated the reconstruction accuracy to be 90 percent (95% for Chinese region and 80% for English region). It is well known that sharewares are provided without charge for the general public. The accuracy rates are estimated without large-scale experiments. Our proposed corpus-based evaluation method based on information restoration can be used for this purpose, if the large-scale standard corpora are available.

In addition to automatic evaluating the accuracy rate of Big5fix, we will describe an intelligent 8-th bit reconstruction system, in which statistical language models are used for resolving ambiguities. (Note that there is no similar ambiguity in a pure GB text, in which both high bits of the two bytes are set. As one of the reviewers points out that practical GB documents may be a mixture of ASCII text and GB codes. In that case, the 8-th bit reconstruction problem exists if the channel is not 8-bit clean. However, the problem would need a method to separate ASCII text from GB codes. It is actually out of the scope of this study.)

In comparison, the GB-BIG5 conversion problem, converting simplified characters to traditional characters, is well known and especially important in the days that information flows across the strait rapidly and in a great volume. In addition to book-formed dictionaries or manuals of traditional character-simplified character correspondences, many automatic conversion systems have been designed. Some of the sharewares and products are listed: HC Hanzi Converter shareware, KanjiWeb (漢字通), NJStar (南極星), AsiaSurf (亞洲通), and UnionWin (亞洲心). However, the commonly used tools in the Internet are still one-to-one code converter. Therefore, we can easily find many annoying GB-BIG5 conversion errors in the articles of some newsgroups such as alt.chinese.text.big5 or articles in the BIG5 version of HuaXiaWenZai (華夏文摘). Some typical errors are listed below: 里(裡)、几(幾)、朮(術)、准(準)、系(係)、划(劃)、采(採)、制(製). In addition automatic evaluation of the HC converter and KanjiWeb, we will also introduce a new intelligent GB-BIG5 converter. The statistical Chinese language models used in the system include inter-word character bigram (IWCB), and simulated-annealing clustered word-class bigram (Chang 1994, Chang and Chen 1993).

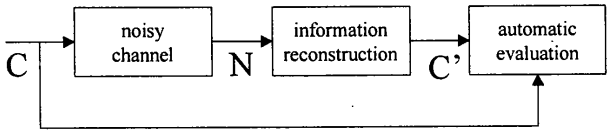


Figure 1: The proposed model

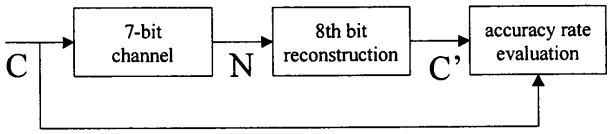


Figure 2: The proposed model for 8th bit reconstruction

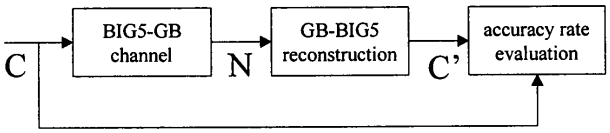


Figure 3: The proposed model for GB-BIG5 conversion

2. Information Restoration Model for Automatic Evaluation

Extending the concepts of 'bi-directional conversion', the proposed corpus-based evaluation method applies the information restoration model for automatically evaluating the performance of various natural language processing systems. As shown in Figure 1, a language processing system is considered as an information restoration process through a noisy channel. Feeding a large-scale standard corpus C into a simulated noisy channel, we can obtain a noisy version of the corpus N. Using N as the input to the language processing system (i.e., the information restoration process), we obtain the output results C'. After that, the automatic evaluation module compares the original corpus C and the output results C', and computes the performance index, accuracy, automatically.

The proposed evaluation model would have a near perfect result (obtaining real performance), if the simulation of noisy channel approaches to perfect. The perfect simulation would be one-to-one correspondence, or a process with near 100% accuracy. For example, for the syllable-to-character conversion system, the noisy channel, character-to-syllable conversion, is not a one-to-one process (there are lots of PoYinZi, homographs). However, it

is not difficult to develop a character-to-syllable converter with an accuracy higher than 98% (Chang 1992, Chen and Lee 1995). Thus, the proposed corpus-based evaluation method is readily applied to estimate the conversion accuracy of a syllable-to-character conversion system. In fact, the proposed model can be applied to various types of language processing systems. Typical examples include linguistic decoding for speech recognition, word segmentation, part-of-speech tagging, OCR post-processing, machine translation, and two problems we will study in this article: 8-th bit reconstruction for BIG5 code, and GB-to-BIG5 character code conversion.

The noisy channel simulation of the 8-th bit reconstruction process is perfect, i.e., one-to-one. The only thing the simulation needs to do is to set the 8-th bit of all bytes to zero. Thus, the proposed corpus-based evaluation method can be ideally applied to the problem. The results would be completely correct. Figure 2 illustrates the proposed model for the 8-th bit reconstruction for BIG5 code.

It is a little complex to simulate the noisy channel for the GB-BIG5 code conversion problem. Not only some traditional characters can be mapped to more than one simplified characters (e.g., 乾⇨干、乾; 覆⇨复、覆), but also more other characters can not find a suitable simplified character to map. Nevertheless, the average accuracy rate for the noisy channel simulation still approaches to 100%, based on occurrence frequency in large corpora. The proposed model is still applicable to the problem, as shown in Figure 3.

3. Preparation of Standard Corpora

In this article, we will use the Academia Sinica Balanced Corpora, versions 1.0 (1995 released, 2 million words) and 2.0 (1996 released, 3.5 million words), to verify our proposed corpus-based evaluation model. Some statistics of the two corpora are listed in Table 1.

Sinica Corpus	Size(bytes)	#files	#sentences	#words	#char.(inclu. symbols)	#char. (Hanzi only)
version 1.0	44,525,299	67	284,455	1,342,861	3,347,981	2,953,065
version 2.0	84,256,391	253	411,470	1,946,958	4,834,933	4,143,021

Table 1: Academia Sinica Balanced Corpora, versions 1.0 and 2.0

The word segmentation and sentence segmentation are used as originally provided by

Academia Sinica. The word segmentation follows the proposed standard by ROCLING, which is an earlier version of the Segmentation Standard for Chinese Natural Language Processing (Draft). The part-of-speech tag set is a 46-tag subset simplified from the CKIP tag set (Huang *et al.* 1995). However, the word segmentations and part-of-speech tags are not used in our experiments. The following steps are used for restoring the text with sentence segmentation:

1. Use *grep* (a Unix tool) to filter out the article classification headers, i.e., lines with leading %%; those sentence separator lines (lines filled with '*') are also removed.
2. Use a small program called *extract-word* to extract the words in a sentence; part-of-speech information has been removed. Output examples are something like "我 起來了 ，"; "太陽 也 起來 了 。"
3. Concatenate words in a sentence into a character string, e.g., "我起來了，"; and concatenate all files into a single huge file.
4. Replace all user-defined special characters and non-BIG5 code with a special symbol '□'.

After pre-processing, the corpus becomes a single file, one sentence per line, and all characters are double-byte BIG5 code. The statistics shown in Table 2 are calculated based on pre-processed version of the corpora.

4. The 8-th Bit Reconstruction

4.1 System Design

The 8-th bit reconstruction problem has been described in Sections 1 and 2. We will not repeat the statement here. To simulate the noisy channel, we simply set zero the 8-th bit of each byte in the input. It can be done in a few lines of program. We will use Big5fix as a baseline system, and develop an intelligent 8-th bit reconstruction system. The system resolves the ambiguity problem using statistical Chinese language models. The basic architecture follows our previous approach called 'confusing set substitution and language model evaluation' (Chang 1994, 1996, Chang and Chen 1993, 1996). As shown in Figure 4, the characters in the input are substituted by corresponding confusing character sets, sentence by sentence. In this way, numbers of sentence string candidates for an input sentence are

generated. Then the string candidates are evaluated through a corpus-based statistical language model. The candidate with the highest score (probability) is chosen to be the output of the system. Here, the step of 'confusing set substitution' can be considered as an inverse simulation of 'noisy channel'.

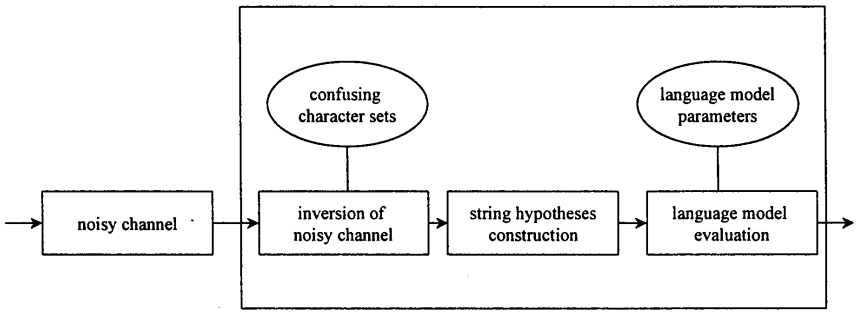


Figure 4. The'confusing set substitution and language model evaluation' approach

For the reconstruction problem, the 'confusing set' is very easy to set up. Since BIG5 is a double-byte coding, we have at most two hypotheses for each character: the 8-th bits of all high-bytes are set to 1, and the 8-th bits of the low-bytes can be either 0 or 1 (dependent on code regions). For example, the inverse simulation confusing set for 2440 (hex) contains two characters a440「一」and a4c0「分」; but the confusing set for 2421 (hex) only contains a character a4a1「丑」(a421 is out of coding region). In the system, we set up confusing sets for each of the 13,060 Chinese characters (including 7 so-called Eten characters). Among them, 10,391 confusing sets contain two characters, while the other 2,669 contain only one character. The statistical language model used in the system is an inter-word character bigram (IWCB) model (Chang 1993). The model is slightly modified from the word-lattice-based character bigram model in Lee *et al.* (1993). Basically, it approximates the effect of word bigram by applying character bigram to the boundary characters of adjacent words. For details of the IWCB model, please refer to Lee *et al.* (1993) and Chang (1993).

4.2 Experimental Results

Table 2 compares the corpus-based evaluation results (number of errors, error rate %) of Big5fix and our intelligent 8-th bit reconstruction system (called CCL-fix).

Sinica Corpus	Samples	#char.	Big5fix		CCL-fix	
Version 1.0	incl. symbols	3,347,981	125,915	3.76	57,862	1.72
	Hanzi	2,953,065	100,006	3.38	53,729	1.81
Version 2.0	incl. symbols	4,834,933	173,544	3.58	71,549	1.48
	Hanzi	4,143,021	111,809	2.69	70,758	1.70

Table 2: Corpus-based evaluation results, Big5fix vs. CCL-fix

As we can see in Table 2, the Hanzi reconstruction rates of Big5fix for Sinica Corpora versions 1.0 and 2.0 are 96.62% and 97.31%, respectively. They are higher than 95% estimated by Huang by 1.62%, 2.31%. The reconstruction rates of CCL-fix are 98.19% and 98.30%, respectively. It shows that the IWCB language model is indeed superior to the counts of character unigram and bigram.

Table 4 lists the reconstruction error analysis for Sinica corpus 1.0 by the two systems. The table shows only the top 20 types of errors with highest frequency. Each entry shows the original character, the reconstructed character, and its occurrence count. For example, the most frequent error made by Big5fix is wrongly reconstructing '分' as '一', with 3,007 occurrences.

Rank	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Big5	分一	化了	林者	外全	全外	區記	色在	股松	來沒	省某	西多	價語	代用	反力	石加	吳找	十天	船爽	油迎	村困
fix	3007	1540	1481	893	819	797	792	771	734	723	722	715	712	709	676	672	664	611	611	601
CCL	一分	了化	分一	又太	沒來	外全	天十	每並	多西	林者	十天	代用	象僅	某省	叫件	沙事	士方	女月	命所	吧扭
fix	2298	1388	1375	1327	1325	1209	1194	887	638	577	530	491	484	465	458	396	386	376	359	343

Table 3: Reconstruction error analysis for Sinica corpus 1.0, Big5fix vs. CCL-fix.

5. GB-Big5 Conversion

5.1 System Design

Three different simulations of the noisy channel for the GB-BIG5 conversion problem are

used in our experiments: (1) HC Hanzi Converter, version 1.2u, developed by Fung F. Lee and Ricky Yeung, (2) HC, revised version: the conversion table is slightly enhanced; and (3) MultiCode of KanziWEB. These three systems all use the table-lookup conversion approach. Thus, the one-to-many mapping problem is not dealt with, and lots of errors can be found when converting GB code back to BIG5.

Table 4 compares the corpus-based evaluation results (number of errors, error rate %) of the three systems: HC1.2u, HC revised, and KanjiWEB .

Sinica Corpus	Samples	# char.	HC1.2u		HC revised		KanjiWEB	
Version 1.0	incl. symbols	3,347,981	271,986	8.12%	46,162	1.37%	29,531	0.87%
	Hanzi	2,953,065	43,155	1.46%	43,070	1.45%	29,076	0.98%
Version 2.0	incl. symbols	4,834,933	403,954	8.35%	68,047	1.40%	43,705	0.90%
	Hanzi	4,143,021	60,113	1.45%	60,031	1.45%	40,561	0.98%

Table 4: Corpus-based evaluation results for HC1.2u, HC revised, and KanjiWEB

To deal with the one-to-many mapping problem in GB-BIG5 conversion, we have developed an intelligent language model conversion method, taking context into account. In the literature, Yang and Fu (1992) presented an intelligent conversion system between Mainland Chinese text files and Taiwan Chinese text files. Their basic approach is (1) build tables by classification; (2) compute scores by levels. However, they resolve ambiguities by asking, instead of using statistical language models. We still take the 'confusing set substitution and language model evaluation' approach. The Chinese language models we used are (1) IWCB model, (2) SA-class bigram model (Chang 1994, 1996, Chang and Chen 1993, 1996) . In the experiments, we use two versions of the SA-class bigram model, with 200 and 300 word-classes, respectively. They will be denoted as SA-200 and SA-300 models.

To simulate the inverse noisy channel, we must set up confusing sets, that is, collection of variants and equivalent characters. In other words, it is a simulation of one-to-many mapping from GB to BIG5. We have found three sources of variants and equivalent characters: (1) the YiTiZi file in HC version 1.2u, (2) Annotation table of simplified characters in the mainland by Zang (1996), (3) Appendix 10 of Hsiao et al. (1993)'s project report. Combining the three sources, we have arranged four versions of confusing sets (A, B, C, and D), which are used and compared in he experiments. Some statistics of the four versions of confusing sets are

shown in Table 5. The column label 'n-way' shows the number of characters for which there are *n* characters in their confusing sets.

Confusing Set	Source	1-way	2-way	3-way	4-way	5-way
A	(1)	12644	364	48	4	0
B	(1)(2)	12397	597	57	9	0
C	(3)	12301	670	68	16	5
B	(1)(2)(3)	12144	777	117	15	7

Table 5: Statistics of the four versions of confusing sets

5.2 Experimental Results

Table 6 compares the corpus-based evaluation results (number of errors, error rate %) of the three language models and four versions of confusing sets for GB-BIG5 conversion. (The input is provided by the HC Revised.)

Sinica Corpus	Number of char.	IWCB				SA-200				SA-300			
		A	B	C	D	A	B	C	D	A	B	C	D
Version 1.0	2,953,065	12,742	10,144	12,997	12,684	15,574	13,977	16,867	16,811	13,614	10,849	13,500	13,225
		0.43%	0.34%	0.43%	0.42%	0.52%	0.47%	0.57%	0.56%	0.44%	0.36%	0.45%	0.44%
Version 2.0	4,143,021	17,752	14,139	18,774	18,465	21,127	18,593	23,299	23,297	18,729	15,439	19,790	19,554
		0.42%	0.34%	0.45%	0.44%	0.50%	0.44%	0.56%	0.56%	0.45%	0.37%	0.47%	0.47%

Table 6: Comparing four versions of confusing sets with three language models

We can see that the IWCB model achieved the best performance for the problem. The SA-300 model has comparative performance, while the SA-200 model is relatively weak. However, we must notice that the three intelligent conversion methods are all superior to KanjiWEB's one-to-one mapping method. The error rates are more than doubled in one-to-one mapping system. Among the four versions of confusing sets, version B performs better than the others. Version C and version D have a larger set of confusing characters than version B, but their performance can not reflect that. The reason might be larger sets make more unnecessary confusions. In contrast, Version A has clearly insufficient numbers of confusing characters.

Table 7 lists the conversion error analysis for Sinica corpus 2.0 by the four systems

(HC1.2u, KanziWEB, IWCB, and SA300 with confusing set version B. The notation is similar to that in the above section. □ or blanks denote that no corresponding character, a1bc(hex) or a140(hex).

Rank	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
HC 1.2u	裡里	並并	術朮	幾几	準准	係系	遊游	劃划	製制	採采	證証	願愿	臺台	範□	隻只	築□	妳□	豐丰	復複	衝沖
	6207	5974	4574	3434	2052	1985	1866	1800	1513	1464	1430	1321	1071	937	860	850	825	797	758	713
Kanzi WEB	裡里	聽听	係系	遊游	製制	採采	臺台	妳奶	複復	衝沖	週周	牠它	症癥	蘇甦	幹干	儘盡	鬧鬧	碰踫	欸	佈布
	6207	2922	1985	1866	1513	1464	1071	825	781	713	668	667	620	603	564	538	455	446	440	439
IWCB /B	臺台	妳你	台臺	牠它	欸□	瞭了	佈布	昇升	裡里	週周	污汙	裏裡	周週	註注	夸誇	秘祕	佔占	儘盡	唸念	繫系
	885	825	761	603	440	383	367	325	319	270	248	220	203	196	194	183	181	178	175	155
SA-300B	裡里	臺台	妳你	牠它	欸□	秘祕	瞭了	佈布	佔占	註注	污汙	周週	台臺	念唸	週周	昇升	裏裡	升昇	夸誇	証證
	1544	994	825	634	440	355	353	310	263	239	237	234	223	221	212	206	202	196	194	154

Table 7: conversion error analysis for Sinica corpus 2.0 by the four systems

6. Concluding Remarks

In this article, we have presented a corpus-based information restoration model for automatic evaluation of NLP systems, and applied the proposed model to two common and important problems related to Chinese NLP for the Internet: the 8-th bit restoration of BIG-5 code through non-iso8859-1 channel, and GB-BIG5 code conversion. Sinica Corpora version 1.0 and 2.0 are used in the experiment. The results show that the proposed model is useful and practical.

Acknowledgements

This paper is a partial result of the project no. 3P11200 conducted by the ITRI under sponsorship of the Minister of Economic Affairs, R.O.C. Previous versions of the paper *in Chinese* appeared in JSCL-97 (Beijing) and Communications of COLIPS (Singapore). One of the reviewers suggested that the title of this paper be changed to "Noisy Channel Models for Corrupted Chinese Text Restoration and GB-to-Big5 Conversion". For consistency, we have kept the original title. Thanks are due to the reviewers for the constructive and helpful comments.

References

Chang, C.-H., Bidirectional Conversion between Mandarin Syllables and Chinese Characters. In *Proceedings of ICCPCOL-92,* Florida, USA, 1992 , pp. 174-181.

Chang, C.-H., Corpus-based Adaptation for Chinese Homophone Disambiguation. *Proceedings of Workshop on Very Large Corpora,* 1993, pp. 94-101.

Chang, C.-H. and C.-D. Chen, Automatic Clustering of Chinese Characters and Words. In Proceedings of ROCLING VI, Taiwan, 1993, pp.57-78.

Chang, C.-H., Word Class Discovery for Contextual Post-processing of Chinese Handwriting Recognition. In Proceedings of COLING-94, Japan, 1994, pp. 1221-1225.

Chang, C.-H., Simulated Annealing Clustering of Chinese Words for Contextual Text Recognition, Pattern Recognition Letters, 17, 1996, pp.57-66.

Chang, C.-H. and C.-D. Chen, Application Issues of SA-class Bigram Language Models, Computer Processing of Oriental Languages, 10(1), 1996, pp.1-15.

Chen, H.-H. and Y.-S. Lee, An Adaptive Learning Algorithm for Task Adaptation in Chinese Homophone Disambiguation, Computer Processing of Chinese and Oriental Languages, 9(1), 1995, pp. 49-58.

Chen, S.-D., An OCR Post-Processing Method Based on Noisy Channel, Ph.D. Dissertation, National Tsing Hua University, Hsinchu, Taiwan, 1996.

Guo, J., On World Wide Web and its Internationalization. In the COLIPS Internet Seminar Souvenir Magazine, Singapore, 1996.

Guo, J. and H.-C. Lui, PH: a Chinese Corpus for Pinyin-Hanzi Transcription, TR93-112-0, Institute of Systems Science, National University of Singapore, 1992.

Hsiao J.-P. et al., Research Project Report on Common Chinese Information Terms Mapping and Computer Character Code Mapping across the Strait, 1993. (in Chinese)

Huang C.-R. et al. Introduction to Academia Sinica Balance Corpus, In Proceedings of ROCLING VIII, 1995, pp. 81-99. (in Chinese)

Huang, S.-K., big5fix-0.10, 1995. ftp://ftp.nctu.edu.tw/Chinese/ifcss/software/unix/c-utils/big5fix-0.10.tar.gz

Kernighan, M.D., K.W. Church, and W.A. Gale, A Spelling Correction Program Based on a Noisy Channel Model. In Proceedings of COLING-90, 1990, pp. 205-210.

Lee L.-S. et al., Golden Mandarin (II) - an Improved Single-Chip Real-time Mandarin Dictation Machine for Chinese Language with Very Large Vocabulary. In Proceedings of ICASSP-93, II, 1993, pp. 503-506.

Yang, D. and L. Fu, An Intelligent Conversion System between Mainland Chinese Text Files and Taiwan Chinese Text Files, Journal of Chinese Information Processing, 6(2), 1992, pp.26-34. (in Chinese)

Zang, Y.-H., *How to Break the Barrier between Traditional and Simplified Characters*, China Times Culture, 1996. (in Chinese)

應用隱藏式馬可夫模型於口述對話系統之研究

顏國郎 吳宗憲 林建良

國立成功大學資訊工程研究所

Email:{yangl, chwu, lincl}@server2.iie.ncku.edu.tw

Fax : (06)2747076

摘 要

本文目的在建立一應用於航空訂票及查詢系統中之對話系統。在論文中，首先是蒐集查詢及訂票對話的語料，並分析對話中的句子，推導及歸納輸入語句的語意。在我們的系統中，我們建立一包含語意的隱藏式馬可夫模型 ，此一模型將語音辨識後所產生的候選句作一評分後，刪除一些不可能的句子，然後再由剩下的找出語意分數最高的句子，作為語意的類別並輸出適當之句子，再經由對話管理員針對每一意圖作出適當之回應，以完成顧客查詢及訂位之目的。

一、簡 介

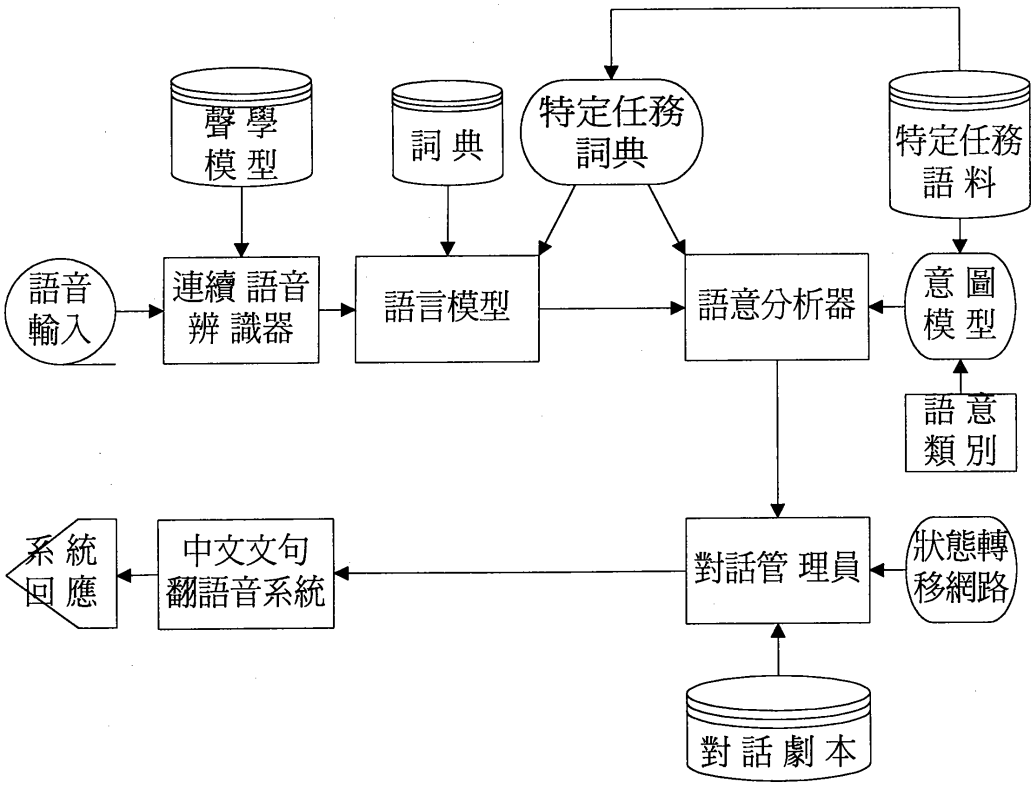
在過去幾年中，語音辨識技術已漸趨成熟，因此，口述語言對話系統也漸漸走向應用階段。例如資料庫查詢、客戶服務系統、自動總機系統、教學系統及機器控制等，均已有雛形系統相繼建立。在未來下一代電腦之基本功能上，電腦界已將對話口述語言理解列為一重要目標。然而對於無經驗之使用者與電腦間利用自然口述語言之對談，則仍存在一些問題。例如：大量詞彙之即時處理、對於不同口音使用者、自然對話處理之強健度以及人機對話中語意溝通之能力等，仍是大家正努力研究解決方法之主題。

由於完全理解人類的語言仍是相當困難，故目前的研究大多局限在其中的一個領域中。近年來已有很多項系統發表，如 MIT 之市區導覽系統、氣象資料即時查詢、車輛銷售導覽系統等[1][2]，其他如 Bennacef 及 Seide 等亦均有相關系統發表[3][4]，台大發展出銀行電話查詢系統[5]，中華電信研究所提出電話查詢服務系統[6]，而工

研院則發展出氣象查詢系統[7]，在成大則已發展出餐廳及電話查詢等系統[8][9][10]。近年來由於航空工業蓬勃發展，搭乘飛機已漸成一必須之交通方式，但由於搭機人數與日漸增，且班機時刻常常更動，因此，提供一良好之自動化航空資訊服務系統是大家所深切企盼的，所以，本論文乃針對航空資訊查詢及訂位之主題作一研究，以建立一對話模型。下面首先介紹的是系統架構，此部分著重於對話模型之研究，接著是對話語料之分析，我們說明對話語料如何分析取得結果，接下來是語意模型之建立，在此，我們採用隱藏式馬可夫模組(HMM)的觀念來建立語意模型，作為意圖的判斷依據。在對話模型，說明我們在對話中的對策與回應的方式。然後是一些實驗的結果。最後我們做一個結論與討論。

二、系統架構

在這章節中，我們介紹此一航空查詢及訂位系統之系統架構，其方塊圖如圖一所示。此系統可分為連續語音辨識器、對話模型和文句翻語音系統三大部分：



(圖 一)系統架構圖

在論文中，我們針對理解部分建立一對話模型，而對於連續語音辨識及文句翻語音兩系統，將採用本實驗室自行開發之音中仙系統作測試。而對話模型共包含下列三個模組：

(一)、語言模型

在語言模型中，我們將建立一特定任務詞典，此詞典乃根據語料庫之資料而建立，在此語言模型中，首先接受語音辨識器輸出之 syllable lattice，而後根據特定任務詞典將其轉成 word lattice 並送至任務相關語意分析器作意圖之判定。

(二)、任務相關語意分析器

在語意分析中，先根據某一特定任務蒐集之關鍵詞建構詞段類別 (Word Segment Class) 而後針對可能出現詞段類別之組合建立一隱藏式馬可夫模型 (Hidden Markov Model, HMM)，每一個 HMM 代表一個詞段類別序列或意圖 (intention)，對於前述語言模型之輸出，選擇一最佳之意圖表示，而後逐一填入語意欄位，並判斷語意是否完整，否則，則透過對話管理員繼續詢問。

(三)、對話管理員

由於語音理解必須考慮語句前後之語意，因此，在對話管理員中，我們將對話記錄利用一狀態轉移網路表示，利用此一網路選擇所需之對話行為 (dialogue act)，並預測使用者可能接續之回應。

三、對話語料之分析

蒐集的對話語料可分為兩種：一種是語音格式，即錄製在錄音帶上的聲音，可用在辨識系統的訓練跟測試方面；一種是文字格式，便是將語音資料轉換為文字，用來分析對話句型、句子的語意、句子的文法等等，這部份的工作是屬於比較瑣碎的項目，所轉換出來的文字格式對話資料，分別以使用者與系統所說的話，分別一句句標示出來。我們分析所取到的語料，將每個訂票對話分成三個部分，如圖二所示：

(一)、問候語：主要是顧客與訂票人員間彼此間開始的問候語句。

(二)、資訊交換：其中分為

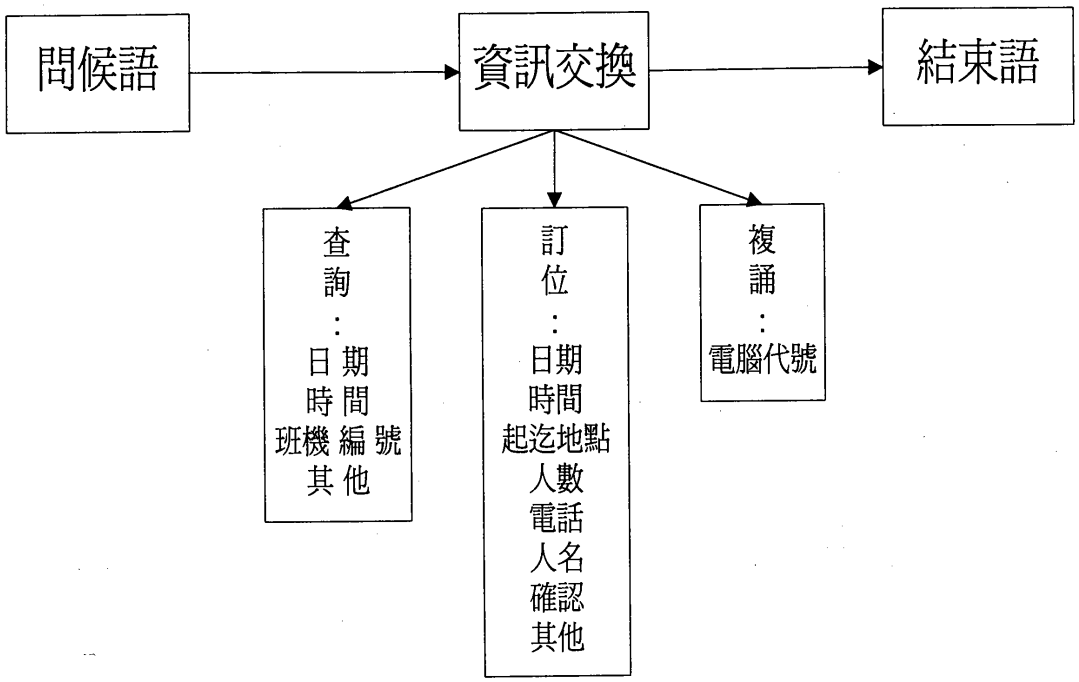
1、 查詢：即使用者查詢班機日期、時間及編號。

2、 訂位：為使用者針對系統所提的問題所做出的回答，或使用者本身提出訂位

資訊如班機之日期、時間、地點、人數、上下午、人名、確認及電話。

3、 複誦：如電腦代號

(三)、結束語：訂位結束或查詢完成之對話語，如謝謝、再見。



(圖二) 訂票對話三大部分圖

針對上述之對話，我們利用對話語料庫建立一特定任務詞典，收錄其使用之詞彙及出現機率。另外在本系統中，有一些關鍵詞在語意上均頗相似，為使電腦方便處理我們將其作分類，因此，可蒐集到之特定任務詞根據其語意分成 21 種詞段類別(Word Segment Class)

(1) 疑問詞：嗎，呢..

(2) 寒喧語 : 你好、喂..

(3) 詢問詞 : 請問、有沒有，幾點有..

(4) 目標詞：班機、班次、機票..

(5) From：從、由、自..

(6) To：到、飛、回、往..

(7) 肯定詞：是、對、好、可以..

(8) 否定詞：除了、不是、不要..

(9) 時間前約略語：大約、靠近、最早..

(10) 時間後約略語：左右、以前、以後..

(11) 動作詞：我要(訂)、我想（訂)、訂（位、票）..

(12) 時間：早上、x點x分..

(13) 班次：第x班、x班次..

(14) 地點：台北、高雄、那邊..

(15) 日期：x月x號、星期(禮拜)x、今天..

(16) 人數：位、張..

(17) Filler：哦、嗯、的、那個、怎麼..

(18) 複誦：代號、英文字、數字..

(19) 數字：1-59

(20) 人名 ： 李登輝、連戰..

(21) Goodbye：: 再見、謝謝、拜..

根據上述分類方式，每一輸入語句將可由 word lattice 轉成 class lattice ， 再送入語意分析器選擇最佳之語意序列。

四、語意模型之建立

在語意模型方面，我們採用隱藏式馬可夫模組(HMM)的觀念來建立語意模型。建立的步驟分為蒐集關鍵字、關鍵字分類與訂定語意分析模組三個階段，以下分別簡述之:

1. 蒐集關鍵詞 ： 這個階段主要的工作在於把所有蒐集到的語料，經過適當的斷詞處理，把它分為不重複且重要的詞，我們就稱此詞為關鍵詞。也就是把一個句子看成有順序的關鍵詞的組合。在電腦與人工的配合之下，我們採用半自動化的處理，對所有收集的語料共蒐集到200多個關鍵詞。
2. 關鍵字分類 ： 這個階段主要對關鍵詞的詞段作分類，也就是 Word Segment Class。分類主要的用意是在作分析語意的走勢，對其作語意評估的依據。我們利用詞的前後相關來作為分類依據，也就是說相同語意的關鍵詞會有相同的前後相關詞。
3. 訂定語意分析模組 ： 利用上面所分出來的關鍵詞類別，訂立語意分析模組是

這個階段的工作。我們所訂立的模組是採用隱藏式馬可夫模組，每一個隱藏式馬可夫模組代表一意圖(intention)，目前我們將其分成四種意圖，分別為 1. 問候語、2. 查詢、3. 訂位及 4. 結束語，而其他不屬於此四種意圖者則歸類至其他意圖，但不訓練 HMM。關鍵詞的類別是代表隱藏式馬可夫模型的狀態(state)，而在每個關鍵詞類別的關鍵詞出現的機率，代表隱藏式馬可夫模型的觀測機率(observation probability)，每一個狀態轉移即代表詞段類別之轉移方式。由這樣的觀念與動機，我們可以更精細的定義意圖 HMM 的基本要素如下:

離散觀測的意圖 HMM 中之成分定義如下：

(1) N：模組的狀態個數，而每個狀態代表一個詞段類別，所有狀態為{1, 2, . . . , N}，對於第 ℓ 個輸入關鍵詞片語，其狀態表示為 S_ℓ

(2) M：代表每個狀態不同觀測符號的個數，每個觀測符號對應每個任務中的關鍵詞片語，我們可以把觀測符號表示為

$$V = \{V_1, V_2, \ldots, V_M\}$$

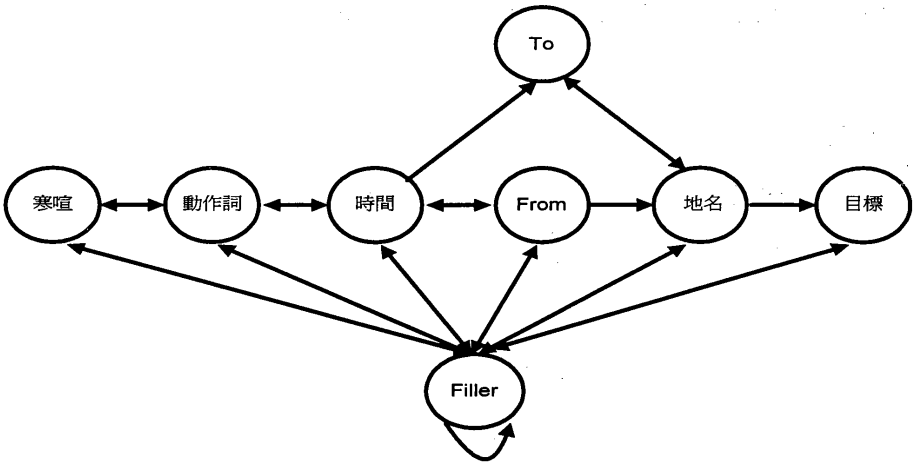
(3) 狀態轉移機率 $a_{ij} = P(S_{l+1} = j \mid S_l = i), 1 \le i, j \le N$

代表狀態 i 轉移到狀態 j 的轉移機率

(4) 觀測機率 $b_j(k) = P[O_l = V_k \mid S_l = j], 1 \le k \le M$ 定義為狀態 j 的觀測機率

(5) 初始狀態為 $\pi_i = P[S_1 = i], 1 \le i \le N$

我們所使用的 HMM 模組是採用比較寬鬆的轉移模組，即任何一個狀態均可以轉移到任何一個狀態，這個轉移模組可由訓練語料調整其轉移的方式而得到一個考量到 N-gram 的 HMM 轉移模組，圖三所示為一個訂位意圖所簡化之 HMM：



(圖三)訂位意圖之簡化 HMM

在語意辨識方面，對於一輸入語音 U，其對應片語序列之分數可以用下列的方程式決定：

$$S(PS_k \mid U) = \max_{1 \le h \le H} [\log P_h(WS_k \mid U)] + \alpha \sum_l \log P(ph_l^k \mid ph_{l-1}^k)$$

其中 PS_k 為第 k 個片語序列，H 為意圖 HMM 之個數。ph_l^k 為第 k 個片語序列中的第 l 個片語。$P(ph_l^k \mid ph_{l-1}^k)$ 為片語的 bigram 機率。$P_h(WS_k \mid U)$ 表示對於一輸入語音 U，其所對應之第 k 個詞段類別序列 WS_k，經過第 h 個意圖 HMM 所得到之機率，其可用下式表示：

$$P_h(WS_k \mid U) = \max_{1 \le i \le N} \delta_L^{k,h}(i)$$

$$\delta_l^{k,h}(i) = \max_{S_1 S_2 \ldots S_{l-1}} P[S_1 S_2 \ldots S_l = i, o_1 o_2 \ldots o_l \mid \lambda_h]$$

其中 N 為狀態數，$\delta_l^{k,h}(i)$ 表示對於一觀測序列 $O = [o_1 o_2 \ldots o_L]$，於第 h 個 HMM 中，利用 Viterbi 演算法，執行至第 l 個片語時所求得之最佳機率。

在 Viterbi 的程序中，觀測機率可用下列來計算：

$$b_j(O_l) = P(O_l \mid S_l = j) * PhS(ph_l^k)$$

其中 $PhS(ph_l^k)$ 表示在第 k 個片語序列 K 中第 l 個的片語正規化的語音分數和片語相似度。

舉例子說明:若輸入對話為:你好，請幫我訂中午十二點台北的飛機。經過上面模組，會做出最適當的語意分析，並輸出最大可能的機率。其最佳路徑之詞段類別序列如下：

寒喧語→動作詞→時間→地點→Filler→目標

五、對話模型

在對話模型方面，我們將使用者的語意分成下列幾種：

一、 問候：顧客對訂票人員的問候語句。如你好、是遠東嗎..等等，若知道使用者純粹是問候語，那我們便回應問候的語句及所需之服務。

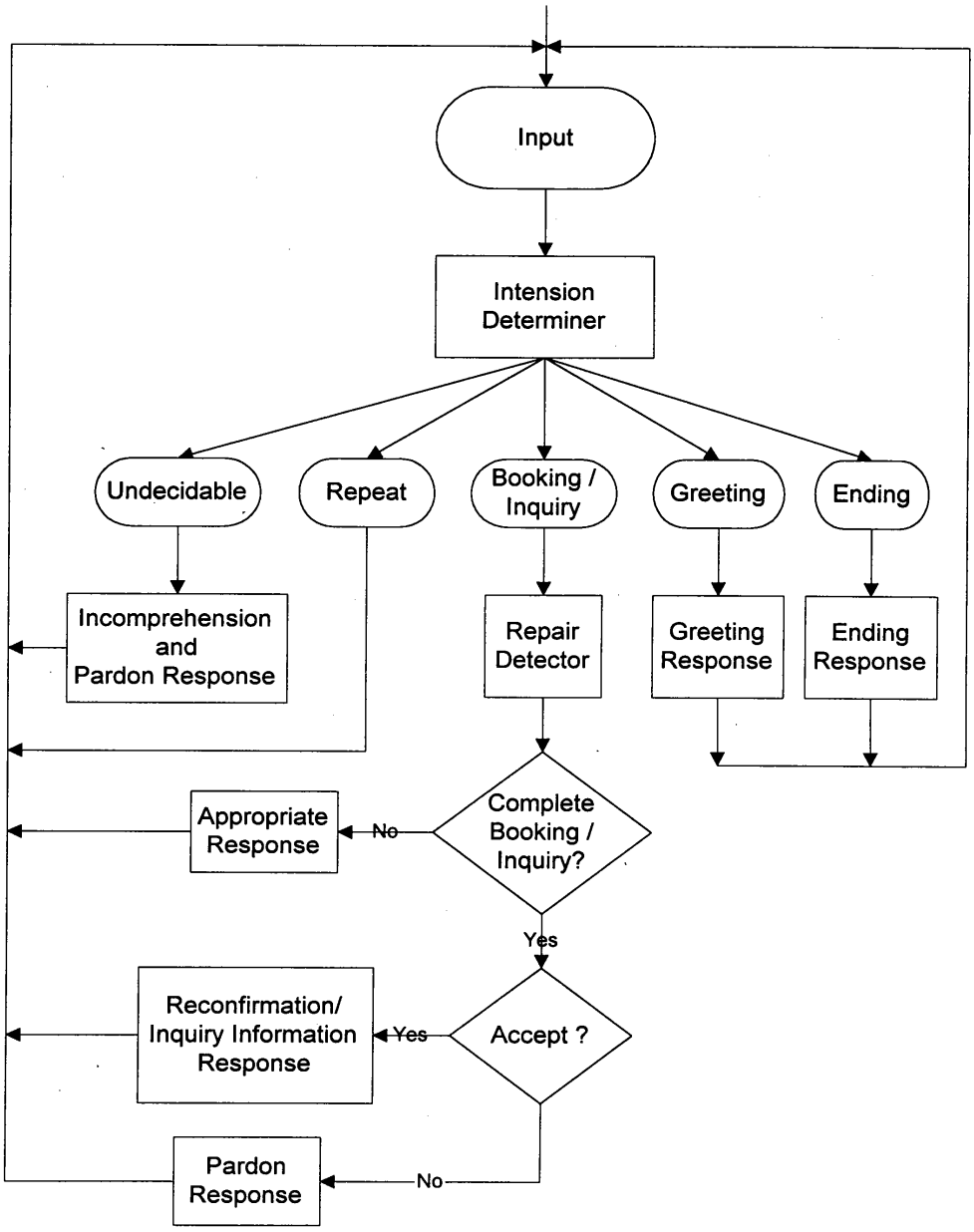
二、 訂票與查詢：要完成一個訂票或查詢過程中、都必須先有日期、起迄點、及時間這三個 Semantic Slot，所以將這兩種動作合在一起處理。若我們知道這是個訂票或查詢的動作，首先我們須從使用者的語句中擷取資料，因為我們不知使用者何時有講到那一個資料或一次講了那幾項，所以當知道使用者是一個訂位或查詢的過程時，我們必須分別對日期、起迄點、時間、人數去偵測並擷取出我們所須要的 Semantic Slots。我們不能因現在已有(時間)這個 Slot 的資料便不去偵測使用者的語句中有沒有這個資訊，因為使用者可能要更改已說過的資料，所以我們加了一個 Repair Detector 來偵測使用者是否有要更改 Slot 的動作。

三、 重複：使用者最常重複系統的話是電腦代號，系統應經由使用者重複的同時檢驗使用者是否有得到正確資料

四、 結束語：通常在對話的尾端不外乎是「拜(拜)」、「好(謝謝)你」等句子，代表使用者可能即將結束對話，此時針對 Slot 的情況給予適當的回應。

五、 其他：一些無法處理的句子，系統告知無法了解並給予上次所提示的句子。

系統的對話模型架構圖如圖四所示:



(圖四)對話模型架構圖

在對話策略上我們採取以下的幾種方式：

1. 混合式交談：系統通常導引使用者回答飛機訂位所須要的資料，相對的，使用者也可以主動的去詢問所須要的資訊，例如：較接近下午三的飛機有那幾班。
2. 重複確認：為了確認系統所取到的資訊是正確的，每當從使用者的回答取得有關訂位所須的重要資料(Semantic Slot)，便複誦給使用者得知，且在對話將完成時複誦所有的資料給使用者做最後的確認。

3. 修復：若使用者查覺系統得到不正確的資料時，便可馬上糾正，以便與系統能做正確的溝通。

4. 覆蓋：假如使用者更正系統所回答的 Semantic Slot 那麼舊的 Semantic Slot 必須被修改，因此系統的對話須架構在新的 Slot 上做出適當的回應。

在系統回應的語句上，我們分成下面幾種類型：

一、問候：在對話的開始的回應句，如"你好"，"復興航空你好"。

二、詢問：當有 Semantic Slot 的資料未取得時，詢問使用者所須的相關資料。

三、選擇：當使用者詢問系統時，系統給予相關的資料且要求使用者輸入更多的資訊，如：

使用者：請問有沒有下午三點左右的班機

系　統：較接近的有，二點四十分與三點三十分，請問你要那一班

四、重複：假如系統根據所辨識出來的句子，無法做出適當的回應，則系統告知使用者無法理解的資訊且要求使用者再輸入一次。

五、失敗：若系統超過三次無法做出適當的回應，則轉由人工處理。

在互動式的系統中，系統必須不斷地與使用者溝通，直到完整地處理完使用者的需求為止。而在訂位意圖之對話中，什麼時候才能算是對話的終結呢？我們定義，當使用者訂票的內容包含了下列七項語意類別時，我們就可以說這位使用者完成了一次的訂票流程：

[日期]、[時間]、[起點]、[終點]、[姓名]
[電話號碼]、[人數]

在對話過程中，系統要決定到目前為止，語意類別還缺了那幾項，且根據它的優先須序來決定回應語句，例如：要詢問[時間]時，必須先成[日期]與[起點、終點]這幾個語意類別。如此這個流程一直到產生完整的類別組合為止。

六、實驗結果

本航空訂票查詢系統使用「音中仙」的 API(Application Program Interface)為連續語音辨識為辨識器，並結合中文文句翻語音系統（TTS）之 API 所發展而成。語音輸入後，辨識器輸出音節(syllable)的候選音，透過自然語言的處理，配出 n 個候選句子，再把這些句子輸入 HMM 語意模型中，把一些不合理的句子刪除，再求出其語意最高分數的句子，其所代表的語意便是我們所要。

在測試系統語音辨識部份時，我們先針對每個詞段類別的關鍵詞任意挑選唸 20 遍，再統計每個類別的正確率，結果如表一所示。

表一、單一關鍵詞的辨識率

類別	疑問詞	問候語	詢問詞	目標詞	From	To	肯定詞
正確率	85	90	95	95	80	75	85
類別	否定詞	時間前約略語	時間後約略語	動作詞	時間	班次	地點
正確率	85	90	90	90	85	80	100
類別	日期	人數	Filler	複誦	數字	人名	結束語
正確率	75	85	/////	/////	85	//////	95

在測試意圖 HMM 的效能方面，我們以人工蒐集錄音之航空查詢及訂位之對話語句共 82 個對話，其中包含一千多個對話語句，而後我們以人工把蒐集的語料，做人工的標示後，分別訓練四個意圖 HMM。而後我們從訓練語料中任意挑選 284 句作測試，測試結果如表二所示。而後，我們再由一些不包含在訓練對話語句的對話中，取出 172 句作人工標示後測試，其結果如表三所示。

表二、意圖 HMM 的效能測試表(close test)

	Booking Intention (110 句)	Query Intention (54 句)	Greeting Intention (33 句)	Ending Intention (42 句)	Other Intention (45 句)
Booking 個數	103	4	1	0	3
Query 個數	6	50	0	0	5
Greeting 個數	0	0	30	0	0
Ending 個數	1	0	2	42	0
Other 個數	0	0	0	0	37

表三、意圖 HMM 的效能測試表(open test)

	Booking Intention (32 句)	Query Intention (25 句)	Greeting Intention (23 句)	Ending Intention (57 句)	Other Intention (25 句)
Booking 個數	29	4	3	5	3
Query 個數	3	21	0	0	2
Greeting 個數	0	0	19	0	0
Ending 個數	0	0	1	52	0
Other 個數	0	0	0	0	20

表格的橫列，代表所用的測試語料的句數，縱列代表系統判斷屬於該種意圖的個數。其中 other intention 代表著不屬於前面四種意圖的句子，也就是被前面四種意圖所拒絕的句子，我們把他歸納為此種意圖。在所有辨識結果中，close test 平均約有 92%的辨識率，open test 約有 82%的辨識率，其中以結束意圖最為正確，原因在於結束用語自成一格，不太會與其他意圖的關鍵詞混淆所致。而其他意圖辨識最差，原因在於其他意圖中包含複誦部分，而複誦句子均是一些數字、英文字母等，且多夾雜很多非關鍵詞，導致辨識效果比較差。

在測試系統對於使用者的語句處理能力方面，我們從收集到的語料庫中找出 20 個

完整的對話做為測試，我們將整個對話根據語意的類別分成五個部份即問候、訂票與查詢(包括日期、時間、起迄點)、電話號碼、複誦及結束語分別做測試，且每個部份再分為 A、B 部兩部分，A 表輸入一次系統便可作出正確的回應，B 表輸入三次以內可作出正確的回應，結果如表四。

表四、回應能力統計表

		A		B	
		正確語句	正確率	正確語句	正確率
問候句	20	16	80%	18	90%
訂票/查詢	20	15	75%	17	85%
電話號碼	20	9	45%	10	50%
複誦	20	13	65%	14	70%
結束句	20	16	80%	18	90%

七、結論與討論

語意是一句話最重要的關鍵所在，所以判斷語意對於理解是最重要的部分。本論文在語意模型方面，提出隱藏式馬可夫模組(HMM)的觀念來建立與判別意圖的依據，不但結合了詞與詞之間的連接架構(N-Gram)，也考量到了合法句子的結構。

實驗的結果顯示，在測試的第一部份，針對每個類別的關鍵詞的正確率，其中關鍵詞長度過短或過長都會影響辨識結果，再者有關數字的關鍵詞辨識率較不高；其他的關鍵詞辨識率還不錯，皆可達到八、九成的效果。第二部分針對句子意圖的分析，意圖的正確與否，關係著對話系統的正確率，本論文所建構的意圖 HMM，在辨識率有 82%，對於對話系統處理有顯著的影響。第三部份針對整句的輸入做分析，我們分成問候句、訂票/查詢、電話號碼、複誦、結束句五個部分分別做測試，其中問候句、結束句的正確率較高；電話號碼與複誦這兩部份因有數字及英文字的辨識，若其中一個數字或英文字辨錯便算錯所以辨識率較低。整體上面，HMM 所構成的意圖模組，確實可以運用於意圖的辨識。

在未來的發展，人名的部份仍是一個需要研究與處理的部分。觀察語料的人名部份顧客與航空訂票人員的互動時，若是很普遍的表示方式，訂票人員很快能了解顧客所要表達的字，但通常這樣的來回並不是非常的順利，若是要使電腦能辨認與了解一般的姓名還要再研究。目前我們對姓名這部份尚未加以處理，這將是我們以後的研究重點。

致謝:

本研究部分由工研院電通所支持才得以完成，特此感謝。

重要文獻(References)

[1] Victor W. Zue, "Toward System that Understand Spoken Language," ARPA Strategic Computing Initiative, 1994.

[2] Helen M., Senis B, and Victor Zue, et al. "WHEELS: A Conversational System in the Automobile Classification Domain," ICSLP '96 Vol. 1.

[3] S. Bennacef and L. Lamel et al., "Dialog in the RAILTEL Telephone-Based System," ICSLP'96 Vol. 1.

[4] Frank Seide and Andreas Kellner, "Toward an Automated Directory Information System," EuroSpeech'97 Vol.3. pp.1327-1330

[5] 李琳山, "國語對話技術之初步研究", 交談系統暨語境分析研討會, 中央研究院 1997

[6] Chun-Jen Lee, Eng-Fong Huang, and Jung-Kuei Chen, "A Multi-Keyword Spotter for the Application of the TL Phone Directory Assistant Service," Proceedings of 1997 Workshop on Distributed System Technologies & Applications, pp. 197-202

[7] Tung-Hui Chiang, Chung-Ming Peng, Yi-Chung Lin, Huei Ming Wang and Shih-Chieh Chien, "The Desigh of A Mandarin Chinese Spoken Dialogue System," in Proceedings of COTEC'98 , 台北 1998, pp. E2-5.1~E2-5.7

[8] 張哲賓,王駿發, “以音中仙為基礎之火車時刻表查詢系統”, 國立成功大學資訊工程研究所, 研究報告, 1996.

[9] 劉定儀, “中文電話自動回應系統之研究”, 國立成功大學,電機工程研究所,碩士論文, 1993.

[10] Hsien-Chang Wang, Jhing-Fa Wang, and Yi-Nan Liu, “A Conversational Agent for Food ordering Dialog Based on Venus Dictate,” Proceedings of ROCLING X International Conference 1997, pp. 325-334

電話查詢口語對話系統中語音辨識不確定性之處理

Dealing With The Uncertainty Of Speech Recognition For Spoken Telephone-Number Inquiry System

+*王駿發　*王獻章　*劉倚男

{wangjf, wangsj, liuyn }@server2.iie.ncku.edu.tw

*國立成功大學資訊工程學系

+國立成功大學電機工程學系

摘要

在口語對話系統中，語音辨識的正確性是很重要的環節。一般語音辨識系統產生不確定的因素有聲調、代換、少字、多字的問題或者建立不恰當的 Bigram 語言模型產生的錯誤。在本論文中，我們提出了一個語音辨識後的精鍊處理系統來解決上述的問題。我們在語音辨識之後，運用原有電話查詢領域的知識庫來做後處理，使得使用者的輸入在語音辨識處理之後，能夠進一步地加以精鍊成正確率更高的文句。

1. 簡介

最近這些年來，學者們對口語對話系統進行了廣泛的研究[1]，其目的就是要完成一台可以適當的與使用者對話進而提供服務的機器。目前有許多的應用系統，像是觀光導覽系統[2]，鐵路資訊查詢/定票系統[4]，汽車買賣資訊系統[5]，或者是餐廳的點菜[3,6]等等，都一一被開發出來並且都展示其服務人類的功能。

而對話系統中佔核心地位的語音辨識系統的正確率目前仍然不能達到百分之百，語音辨識的錯誤通常會導致對話系統無法得到正確的結果。因此我們對語音辨識後的結果做一個分析，然後針對語音辨識可能的結果或錯誤如聲調(Tone)、代換(Substitution) 、或者多字(Insertion)、少字(Deletion)、Bigram 語言模型的錯誤問題加以處理，並將它應用在辦公室語音轉接系統[7]上。實驗結果可將句子的關鍵詞的正確率由原來的 72.5%提升到 82.5%，句子的正確率由原來的 65%提升到 78%，而對話的完成率由 78.10%提升到 88.58%。

我們也歸納出一個快速的演算法可以縮減關鍵詞比對的運算量，此演算法將代換、

多字和少字的處理程序合併運算，使得原先比較的運算量，在一字錯誤(1-error)的情形下，時間的複雜度由 $O(n^2)$縮減為 $O(n)$。

本論文的章節架構如下：第二節介紹我們的系統架構；在第三節，我們描述系統知識庫的建立；第四節則介紹我們精鍊模型的建立與處理；第五節說明如何建立快速的演算法以考慮一字錯誤的比對情形；實驗的結果則在第六節描述；最後，第七節則是結論與討論。

2. 系統架構

我們針對語音辨識端的辨識模組所產生的候選音節(syllable lattice)和構句之後的文字串(word sequence)再加以精練處理(refinement)以得到更高的辨識效果。精鍊模組的處理流程如圖 1。

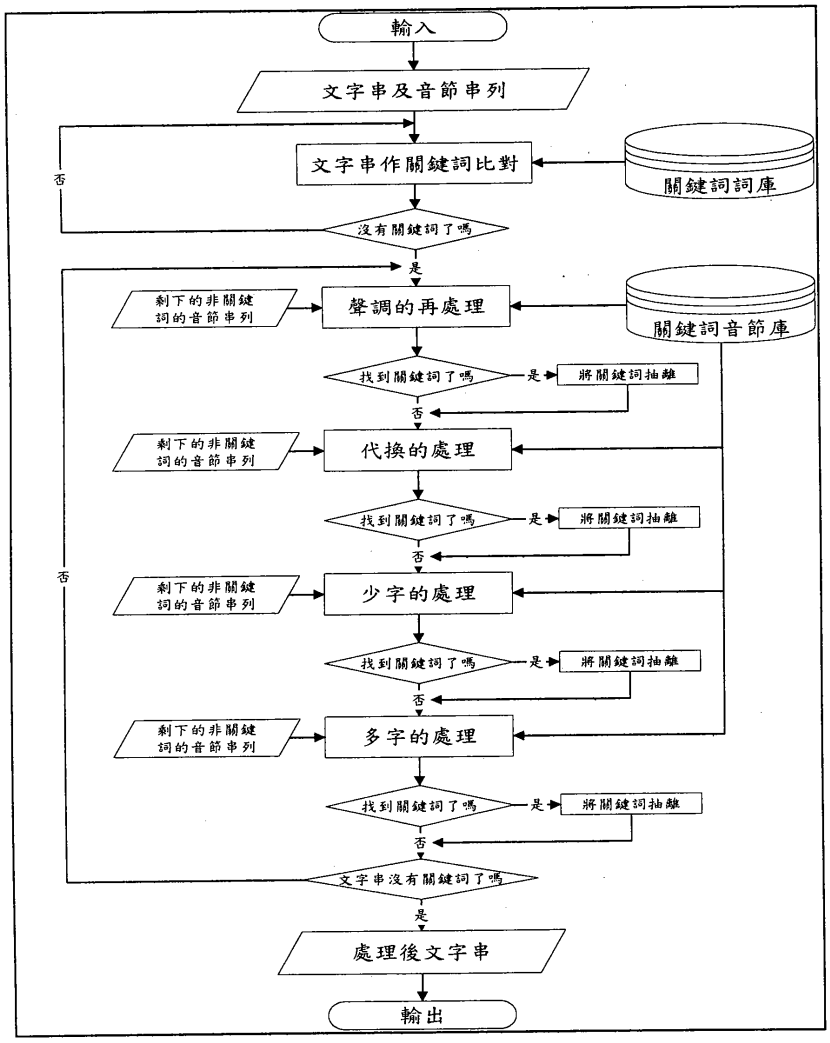


圖 1. 精鍊處理的流程圖

首先，文字串會與關鍵詞庫作比對，將關鍵詞抽取之後，剩下的非關鍵字開始作後處理的動作，我們使用非關鍵詞的候選音節來處理。第一步是作聲調的再處理，我們使用事先轉換好的關鍵詞音節庫作比對，忽略聲調的資訊可以修正大部分的錯誤；接著我們作代換字的處理，除了處理一般音節被代換嘗試修正外，我們並且以候選音節相似度的分數來做考量；再來就是多字少字的處理，這個步驟的混淆度比較高，所以本論文只處理一個字的多字少字的問題。在比對之前我們會先選擇適當的關鍵詞類別來做比對，選擇的依據是以詞類間相連關係來做判定。在處理完之後，我們便將處理後的文字串輸出到對話管理模組跟使用者進行對話。

3. 知識庫的建立

關鍵詞可以作為系統可認知的知識庫，在這裡我們使用查詢系統所定義的關鍵詞來當作精鍊模組的知識庫，我們所訂定的詞庫類別有兩類，每一個類別是以相同的類型作分類，第一類是重要關鍵詞(primary keyword)，是系統認知人員資料的重要資訊，包括姓、全名、性別稱謂、單位、工作/研究領域，第二類是次要關鍵詞(secondary keyword) [9]，定義語者的意圖或語末詞，包括前置詞、後置詞，如下表所示：

詞庫類別	類別名稱	意義	類別資料範例
Primary Keywords	姓(surname)	所有的姓，以百家姓裡的姓為主。	趙，錢，孫，李
	全名(full-name)	人名的全名。	王獻章，劉倚男
	性別(sexual)	男性與女性的性別稱謂。	先生，小姐
	職稱(title)	單位中可能出現的職稱。	教授，所長
	單位(department)	各個單位的名稱。	電算中心，資訊所
	工作/研究領域 (working/research area)	工作及研究領域的名稱，並且將各個名稱可能的簡稱也一併列入。	語音辨識，資訊安全
Secondary Keywords	前置詞 (pre-keyword)	接在重要關鍵詞之前的關鍵詞。	麻煩幫我轉，請幫我接
	後置詞 (post-keyword)	接在重要關鍵詞之後的關鍵詞。	在不在，可以嗎

表 1. 關鍵詞詞庫類別與範例

4. 精鍊模組的建立與處理

4.1 辨識結果錯誤的分析

我們對口述語言對話系統中的語音辨識模組所產生的錯誤作了整理並且將它歸類。這些錯誤有的是與者在輸入語音時發音不良所導致的；有的則是辨識模組的核心程式強健性不足所產生的。這些錯誤分別舉例如下：

1.聲調(Tone)的錯誤：

由於語者說話的口音或者習慣而會有聲調上的辨識錯誤。下面是辨識後聲調錯誤產生錯誤結果的例子如下：

- 使用者：麻煩【找】吳宗憲老師
- 辨識後：麻煩【招】吳宗憲老師

2.代換(Substitution)的錯誤：

語者將某個音唸錯或發音不標準，造成辨識的錯誤，這樣的錯誤會導致配詞結果成為較奇怪的文字串，常常是不合文法或語法或是變成一些贅語，這樣會使得對話管理模組處理困難，降低對話系統處理的正確性，下面是一個代換錯誤的例子：

- 使用者：幫我轉【資訊所】
- 辨識後：幫我轉【制憲所】

3.少字(Deletion)的錯誤：

語者唸的時候，可能是唸太快或者是太小聲，而產生漏字的情形，例如：

- 使用者：可不可以請問一下從台北到【嘉義】的自強號票價多少
- 辨識後：可不可以請問一下從台北到【家　】的自強號票價多少

4.多字(Insertion)的錯誤：

語者唸的時候，可能是唸太慢或者是含混不清，而產生多字的情形，例如：

- 使用者：你好，請幫我接【王獻　章】
- 辨識後：你好，請幫我接【王獻煙章】

5. Bigram 語言模型的錯誤：

由於在通用語料和特定領域語料所合成的 bigram information 中，我們將特定領域的 bigram 的機率分數[10]權重調得比較高，而導致的錯誤，我們可以見下面的例子：

- 使用者：請幫我找【黃】教授好嗎

- 辨識後：請幫我找【煌】教授好嗎

上面發生錯誤的原因是因為語料庫中"郭耀煌教授"出現頻率很高，所以 bigram 的分數比較高，因而造成這種錯誤。

4.2 精鍊的處理

4.2.1 聲調(Tone)的再處理

聲調再處理的流程圖如圖 2.所示：

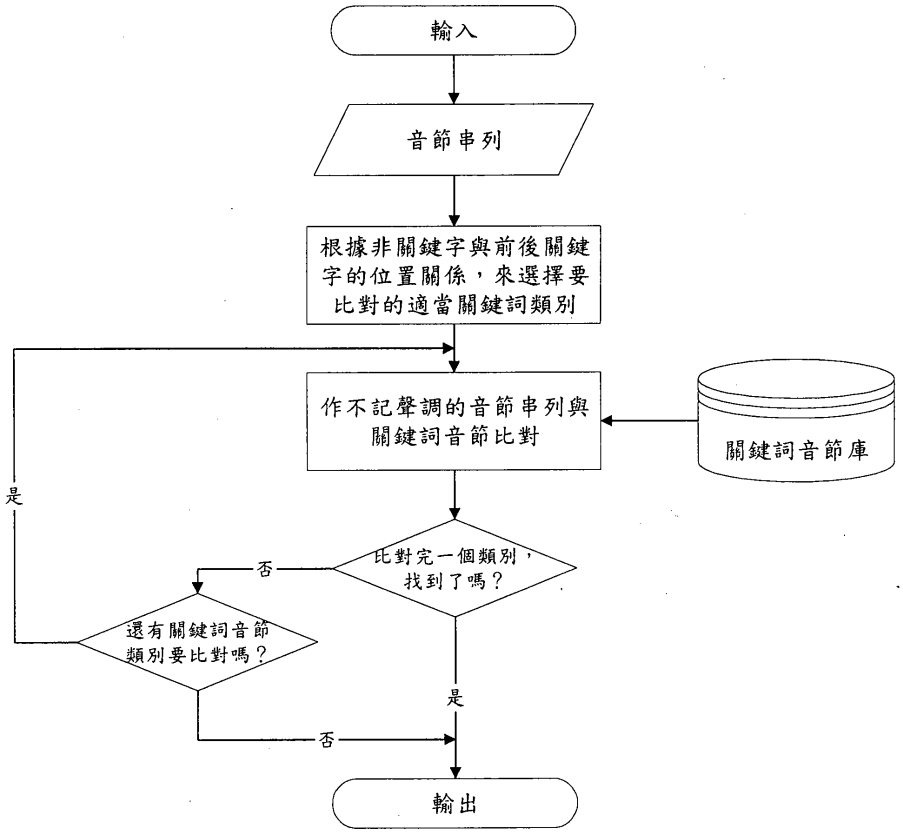


圖 2. 聲調再處理的流程圖

首先我們將關鍵詞庫的音節資訊建立成音節庫。接下來做比對的時候要挑選適當的關鍵詞類別來跟音節串列作比對。我們藉由候選詞與前後可能接的重要關鍵詞的關係來選擇比對的類別，然後我們開始作關鍵詞的比對。比對的方式是將關鍵詞音節與音節串列先做去聲調的處理，然後再比對出相吻合的，如果比對成功就跳出聲調處理的副程式；否則會繼續作下一個類別的比對，直到沒有任何關鍵詞音節。關鍵詞前後相連的規則，根據我們觀察的結果，定義如表 2 所示。

前後已經存在關鍵詞的數目	候選詞與關鍵詞的位置	組合情形	可能的候選詞類別
無	不用考慮	不用考慮	Fullname，Department，Sex，Title，Research，Prekeyword，Postkeyword
一個	候選詞在關鍵詞之前	候選詞＋F 候選詞＋D 候選詞＋S 候選詞＋T 候選詞＋R 候選詞＋Post	D,T,Pre D,Pre F,Pre F,D,R,Pre F,D,Pre F,D,S,T,R,Pre
	候選詞在關鍵詞之後	F+候選詞 D+候選詞 R+候選詞 Pre+候選詞	S,T,R,Post F,D,T,R,Post S,T,Post F,D,S,T,R,Post
兩個	候選詞在前後關鍵詞之間	Pre+候選詞+F Pre+候選詞+D Pre+候選詞+S Pre+候選詞+T Pre+候選詞+Post F+候選詞+D F+候選詞+R F+候選詞+Post D+候選詞+F D+候選詞+S D+候選詞+T D+候選詞+R D+候選詞+Post S+候選詞+D S+候選詞+Post T+候選詞+D T+候選詞+Post R+候選詞+Post	D,S,T,R F,D,T,R F,D,T,R F,D,S,R F,D,S,T,R D,S,T S,T S,T,R D,S,T F,D,T F,D,S F,D F,D,S,T,R D F,R D F,R S,T

表 2. 挑選適當關鍵詞類別方法表

其中 F 表示全名(Full-name)；D 表示單位(Department)；S 表示性別(Sex)；T 表示職稱(Title)；R 表示研究領域/工作性質(Researching Area / Working area)；Pre 表示前置詞(Pre-Keyword)；Post 表示後置詞(Post-Keyword)。

4.2.2 代換(Substitution)的處理

從這個步驟開始，一直到多字情形的處理，聲調的資訊就不再考慮。代換處理的流程圖如圖 3. 所示。

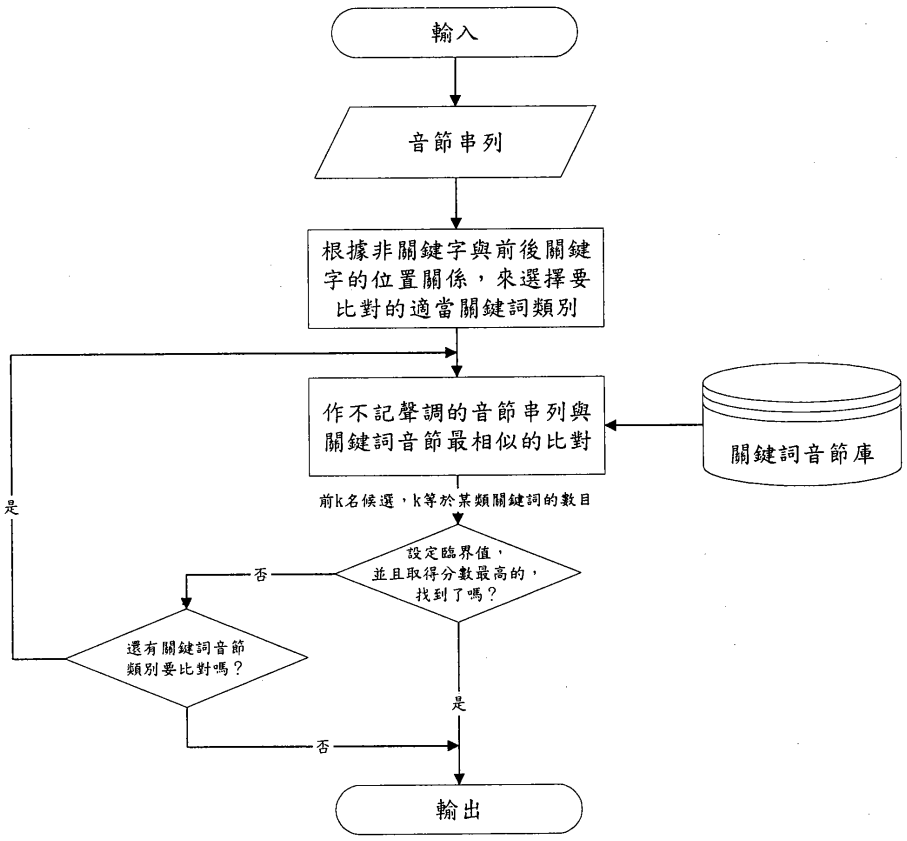


圖 3. 代換的處理流程圖

關鍵詞音節類別的選擇方法如上一節表 2.所示。接下來要作最大相似度的比對，我們定義了分數給定的原則如下表：

相似度	分數
完全正確(子音＋母音)	10
只有母音正確	7
沒有正確的	4

表 3. 音節相似度比對的分數定義表

我們紀錄每個關鍵詞音節的分數，然後選擇一個最高的分數：

$$S_{max} = Max(S_0, S_1, \ldots, S_n), \qquad S_i >= threshold \tag{1}$$

其中 S_i 表示第 i 個關鍵詞的分數。臨界值的計算方式如下，n 為候選音節長度：

$$\text{threshold} = \begin{cases} (n-1)*10+7, if & n=2 \\ (n-1)*10+4, if & n>2 \end{cases} \tag{2}$$

音節分數大於等於臨界值的關鍵詞才有可能是我們要的。在兩字時(n=2)可以處理一個音

節完全正確及另一個音節的母音資訊正確的情形；在三字以上時(n>=3)可以處理一個音節的代換(沒有正確的)或者 n-2 個音節完全正確而且兩個母音資訊正確的情形。圖 4.是比對過程示意圖。

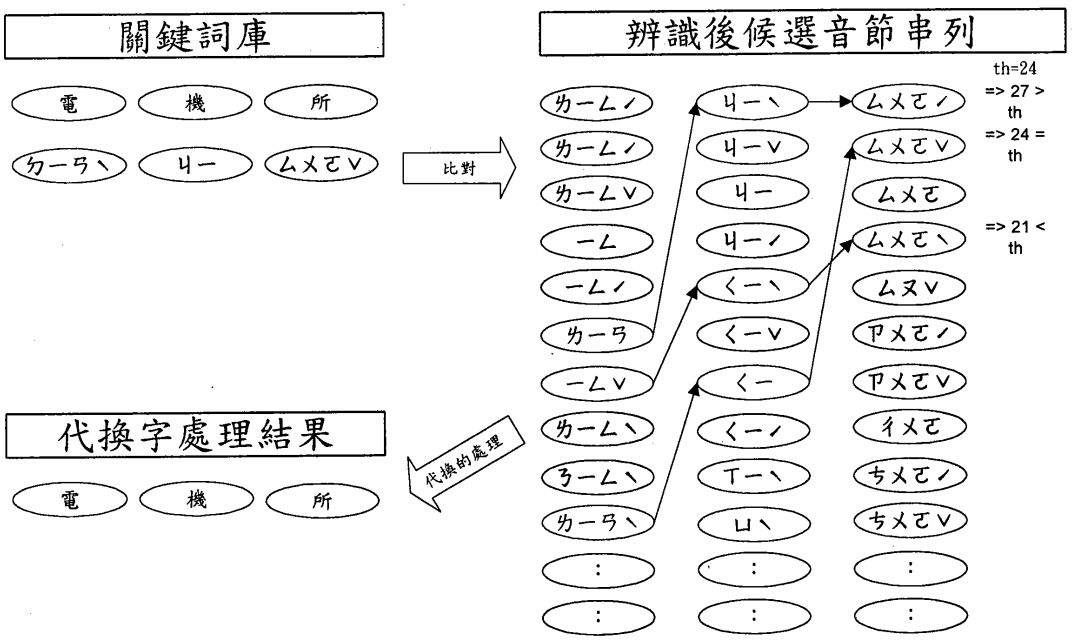


圖 4. 代換字的處理範例圖

4.2.3 少字(Deletion)的處理

少字處理的方式與代換相似，不同的是我們將關鍵詞音節從第一個到最後一個音，輪流刪掉一個之後再跟音節串列作比對，然後選擇一個最高的分數：

$$S_{max} = Max(S_0, S_1, \ldots, S_n), \qquad S_i >= threshold$$

臨界值公式如下，n 為候選音節長度：

$$threshold = n*10, n \geq 2 \tag{3}$$

表示要全部比對正確，加上剛才刪掉的一個音節，就成為少字的情形，找到了就會傳回關鍵詞的字串了。

4.2.4 多字(Insertion)的處理

在多字的處理方式同樣與代換相似。我們將關鍵詞音節從第一個到最後一個音，輪流插入一個音節之後再跟音節串列作比對，插入的音節我們設定為空白，因此分數設為 0，然後選擇一個最高的分數：

然後選擇一個最高的分數：

$$S_{max} = Max(S_0, S_1, \ldots, S_n), \qquad S_i >= threshold$$

臨界值的設定如下，n 為候選音節長度：

$$threshold = (n-1)*10, n \geq 3 \tag{4}$$

表示若比對 n-1 個正確，就可能是多字的情形，找到了就會傳回關鍵詞的字串了。

4.2.5 Bigram 語言模型錯誤的處理

Bigram 語言模型的錯誤，要能夠檢查的出來，必須使用文法的規則來檢查這個錯誤，我們若知道各個關鍵詞可能的連接規則，例如「姓+職稱」或者「姓+性別」等等那我們就可以修正這樣的錯誤。

我們在前面 4.2.1 到 4.2.4 小節的關鍵詞比對之前，會先依據我們所訂定的關鍵詞連接規則來作關鍵詞類別的選擇，這個動作就可以修正這類型的 Bigram 錯誤了。

5. 一字錯誤(1-error)快速演算法的建立

上一章提到的處理方法，需將音節串列與很大的資料庫作比對，雖然先對要比對的資料庫作一個篩選，但是其運算量還是很大，會增加系統處理的時間，我們希望能夠將資料庫比對的計算時間縮減。我們分析整個比對的過程，並找出能夠減少運算的方法。在只有一字錯誤的情形下，若以直覺式(straight forward)的方法，其計算量的分析如圖 5.所示。在此，我們計算比較的次數，因為這個數量佔了大多數的時間。圖 5.中，每個圓圈表示關鍵詞音節庫與音節串列比較一次，其中 X 表示可以為任意字。

1. **多字的情形：**

   四字時(n=4)時，例如「中華民國」，會有 n-1 個插入字的位置，而每行字則比較(n+1)次，所以總共(n-1)*(n+1)=n^2-1 次。

2. **少字的情形：**

   在 n=4 時，n 從 1 到 n 會各少一次，而每一行則剩下(n-1)次，所以總共比較了 n*(n-1)=n^2-n 次。

3. **代換的情形：**

   與少字的情形相似，在 n=4 時，n 從 1 到 n 會各被其他字所代換一次，而每一行維持 n 次，所以總共比較了 n*n=n^2 次。

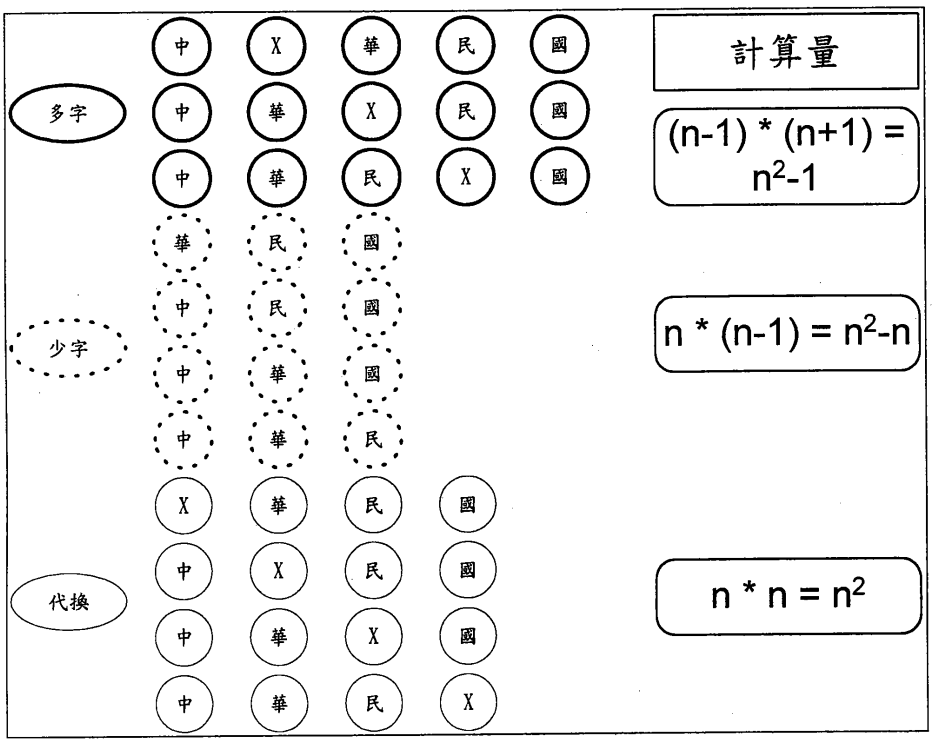


圖 5. 一字錯誤直覺式演算法計算量說明圖，以「中華民國」一詞為例。

接下來，我們將建立一個樹狀的資料結構，並且將上面的資料放入樹的節點，建成樹的方式如下：

1.依照橫列的方式，每一個橫列代表一個樹的分支。

2.將橫列依照從左到右的順序建立成樹的節點。

3.重複(2.)的步驟直到將所有橫列建立完畢。

假設樹高最大是 k 層，則第 k 層為多字，第(k-1)層為代換，而第(k-2)層為少字，範例如下圖：

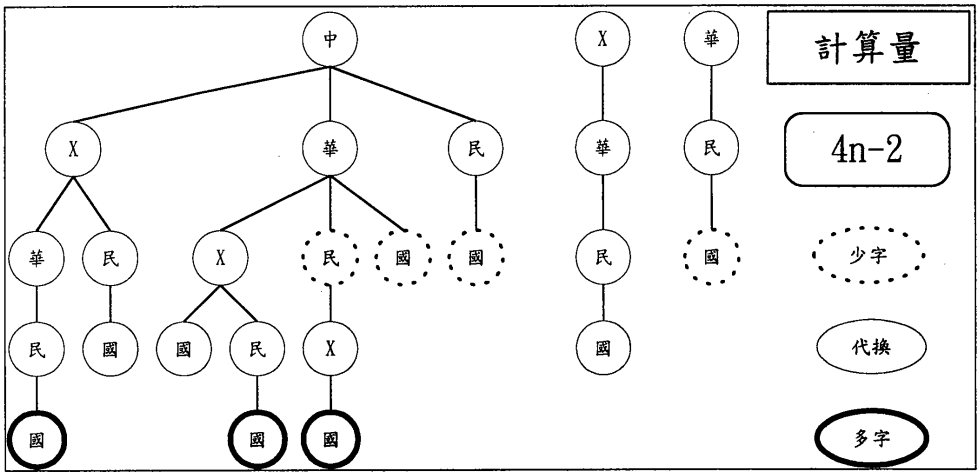


圖 6. 一字錯誤樹狀結構演算法計算量說明圖

舉例說明：從左到右邊，第二子樹，「中華 X 國」為代換，「中華 X 民國」為多字，「中華民」為少字，「中華民 X」為代換，「中華民 X 國」為多字依此類推。

接下來計算它的比較的運算量，設 Tn 代表樹高為 n 的那一層，我們計算每一層出現的不同位置的字的個數，它相當於在直覺式演算法中，每一個直行出現的不同字數，說明如下：

	四字詞「中華民國」	五字詞「資訊工程所」
第 Ti 層	Ti 層需比對的字元	
T0	中, X, 華	資, X, 訊
T1	X, 華, 民	X, 訊, 工
T2	X, 華, 民, 國	X, 訊, 工, 程
T3	X, 民, 國	X, 工, 程, 所
T4	國	X, 程, 所
T5		所

表 4. 一字錯誤情形中各層需要比對的字元範例表

四字時一共是 3+3+4+3+1=14，五字時一共是 3+3+4+4+3+1=18，每增加一個字時，增加 4 個比對時間，因此總計算量為 4n-2，時間複雜度為 O(n)。

6. 實驗

6.1 實驗環境

我們的實驗是以 Pentium 200 個人電腦加上 16 位元聲霸卡、麥克風做測試環境。我們的開發工具是 Microsoft Visual C++ 5.0。語音輸入端是採用連續音辨識程式的 API。文字翻語音的輸出端採用的是成大開發的語音合成 API [8]。

6.2 實驗方法與結果

測試的方式第一種是個別關鍵詞的正確率，我們所使用的資料庫是前面所建立的關鍵詞詞庫，一共有 20 個人名(full name) 、4 個性別稱謂(sex)、10 個職稱(title)、11 個單位(department)、33 個工作/研究領域(working/research area)。表 6 是測試結果，包括語音辨識端的正確率(未經精鍊處理，Base Line)和精鍊模組處理後的正確率(Refined Model)，表中的最後一列是所有關鍵詞正確率的平均，如下所示：

Type	Base Line	Refined Model
Full name	85.0%	92.5 %
Title	82.5 %	90.0 %
Sex	90.0 %	95.0%
Department	92.5 %	95.0 %
Researching area	80.0 %	87.5 %
Average	86.0 %	92.0%

表 6. 個別關鍵詞辨識結果比較表

測試的方式第二種是測試真實語料關鍵詞的正確率，我們測試了 100 句的真實語料。每一句語料都統計所有關鍵詞的個數以及辨認正確的個數；另外是測試整句的正確率，測試方式為，當整句的句子都正確時才算正確。測試結果如表 7 所示，包括語音辨識端的正確率和精鍊模組處理後的正確率。

Type	Base Line	Refined Model
Semantic slot	72.5%	82.5 %
Sentence	65%	78 %

表 7. 語意框的關鍵詞和整句的正確率比較表

測試方式的第三種是測試對話的完成率，我們當自己是使用者然後向系統進行查詢，我們一共測試了 105 組的對話過程。測試結果如表 8 所示：

Type	Base Line	Refined Model
# of dialog	105	105
# of success dialog	82	93
Success rate	78.10%	88.58%

表 8. 對話的完成率

7. 結論與討論

在本論文中，我們提出了解決語音辨識系統常見問題(聲調、代換、少字、多字與 Bigram 語言模型錯誤)的辦法。它可以有效地提高關鍵詞、句子、以至於對話系統的正確率。此外，我們也提出一個增加關鍵詞比對速度的演算法它可以將一字錯誤(1-error)的關鍵詞處理時間由 $O(n^2)$降低為 $O(n)$。

一套對話系統中，會產生問題的部分，除了本文所提到的之外，尚有很多種，例如 Out-of-Word、Out-of-Grammar、Out-of-Task 等等問題。另外，由於使用者輸入的遲疑、重複等等，會造成二字錯誤(2-error)以上的問題，如何提出有效的方法來解決這些問題，將是我們下一步研究的目標。

參考文獻

[1] Furui and M. M. Sondhi, Advances in Speech Signal Processing, Marcel Dekker, Inc., pp.652-699, 1992.

[2] Victor Zue, James Glass, etc., "Spoken Language System for Human/Machine Interfaces", DARPA N00014-89-J-1332, 1991.

[3] Hsien-Chang Wang, Jhing-Fa Wang, and Yi-Nan Liu, "A Conversational Agent for Food_ordering Dialog Based on VenusDictate", Proceedings of ROCLING X International Conference 1997, pp.325-334.

[4] Bennacef and L. Lamel et al., Dialog in the RAILTEL Telephone-Based System, ICSLP'96 Vol. 1.

[5] Helen M., Senis B, and Victor Zue, et al. WHEELS: A Conversational System in the Automobile Classification Domain, ICSLP'96 Vol. 1. pp. 542-545.

[6] Tsuboi and Y. Takebayashi, "A real-time task-oriented speech understanding system using keyword spotting," Proc. ICASSP, pp.197-200,1992.

[7] Hsien-Chang Wang and Jhing-Fa Wang, "A Telephone Number Inquiry System With Dialog Structure," Proceedings of 1997 Multimedia Technology and Applications Symposium. pp.263-270.

[8] Chung-Hsien Wu, J. F. Wang, etc, "Chinese Text-to-Speech System", National Science Community Project Report, NSC-84-2622-E00-006, 1996.

[9] J. Yang, L. F. Chien and L. S. Lee, Speaker Intention Modeling for Large Vocabulary Mandarin Spoken Dialogues. ICSLP'96, Vol. 2. pp. 713-716.

[10] Jyh-Shing Shyuu, Jhing-Fa Wang, "A Speech Input Interface for Web Page Query Based on A Dynamic Language Model Architecture", ICCE'98