Corpus-based Adaptation Mechanisms for Chinese Homophone Disambiguation

Chao-Huang Chang
E000/CCL, Building 11, Industrial Technology Research Institute
Chutung, Hsinchu 31015, Taiwan, R.O.C.
E-mail: changch@e0sun3.ccl.itri.org.tw

Abstract

Based on the concepts of bidirectional conversion and automatic evaluation, we propose two user-adaptation mechanisms, character-preference learning and pseudo-word learning, for resolving Chinese homophone ambiguities in syllable-to-character conversion. The 1991 United Daily corpus of approximately 10 million Chinese characters is used for extraction of 10 reporter-specific article databases and for computation of word frequencies and character bigrams. Experiments show that 20.5 percent (testing sets) to 71.8 percent (training sets) of conversion errors can be eliminated through the proposed mechanisms. These concepts are thus very useful in applications such as Chinese input methods and speech recognition systems.

1 Introduction

Corpus-based Chinese NLP research has been very active in the recent years as more and more computer readable Chinese corpora are available. Reported corpus-based NLP applications [10] include machine translation, word segmentation, character recognition, text classification, lexicography, and spelling checker. In this paper, we will describe our work on adaptive Chinese homophone disambiguation (also known as phonetic-input-to-character conversion or phonetic decoding) using part of the 1991 United Daily (UD) corpus of approximately 10 million Chinese characters (Hanzi).

It requires a coding method, structural or phonetic, to input Chinese characters into a computer, since there are more than 10,000 of them in common use. In the literature [3,7], there are several hundred different coding methods for this purpose. For most users, phonetic coding (Pinyin or Bopomofo) is the choice. To input a Chinese character, the user simply keys in its corresponding phonetic code. It is easy to learn, but suffers from the homophone problem, i.e., a phonetic code corresponding to several different characters. Therefore, the user needs to choose the desired character from a (usually long) list of candidate characters. It is inefficient and annoying. So, automatic homophone disambiguation is highly desirable. Several disambiguation approaches have been reported in the literature [3,7]. Some of them have even been realized in commercial input methods, e.g., Hanin, WangXing, Going. However, the accuracies of these disambiguators are not satisfactory. In this paper, we propose a corpus-based adaptation method for improving the accuracy of homophone disambiguation.

For homophone disambiguation, what we need as input is syllable (phonetic code) corpora instead of text corpora. For adaptation, what we need is personal corpora instead of general corpora (such as the UD corpus). Thus, we first design a selection procedure to extract articles by individual reporters. Ten personal corpora were set up in this way. An additional domain-specific corpus, translated AP news, was built up similarly. Then, we design a highly-reliable (99.7% correct) character-to-syllable converter [1] to transfer the text corpora into syllable corpora.

Our baseline disambiguator is rather conventional, composed of a word-lattice searching module, a path scorer, and a lexicon-driven word hypothesizer. Using the original text corpora and the corresponding syllable corpora, we propose a user-adaptation method, applying the concept of bidirectional conversion [1] and automatic evaluation [2]. The adaptation method includes two parts: character-preference learning and pseudo word learning. Given a personal corpus (i.e., sample text), the adaptation pro-

cedure is able to produce a user-specific character-preference model and a pseudo word lexicon automatically. Then the system can use the user-specific parameters in the two models for improving the conversion accuracy.

Extensive experiments have been conducted for (1) ten sets of local-news articles (one set per reporter) and (2) translated international news from AP News. Each set is divided into two subsets: one for training, the other for testing. The character accuracy of the baseline version is 93.46% on average. With the proposed adaptation method, the augmented version increases the accuracy to 98.16% for the training sets and to 94.80% for the test sets. In other words, 71.8% and 20.5% of the errors have been eliminated, respectively. The results are encouraging for us to further pursue corpus-based adaptive learning methods for Chinese phonetic input and language modeling for speech recognition.

2 Homophone Disambiguation

Mandarin Chinese has approximately 1300 syllables, 13,051 commonly used characters, and more than 100,000 words. Each character is pronounced as a syllable. Thus, it is clear that there are many syllables are shared by numbers of characters. Actually, some syllables correspond to more than 100 characters, e.g., the syllable [yi4] corresponds to 125 characters, 意、義、議、亦、易、益、etc. Thus, homophone (character) disambiguation is difficult but important in Chinese phonetic input methods and speech recognition systems.

The problem of homophone disambiguation can be defined as how to convert a sequence of syllables $S = s_1, s_2, ..., s_n$ (usually a sentence or a clause) into a corresponding sequence of characters $C = c_1, c_2, ..., c_n$ correctly. Here, each s_i stands for one of the 1300 Chinese syllables and each c_i one of the 13,051 characters.

Fortunately, when the characters are grouped into words (the smallest meaningful unit), the homophone problem is lessened. The number of homophone polysyllables is much less than that of homophone characters. (A Chinese word is usually composed of 1 to 4 characters.) For the disambiguation, longer words are usually correct and preferred. Thus, the homophone disambiguation problem is usually formulated as a word-lattice optimal path finding problem. (Note that there is the problem of unknown words, especially personal names, compound words, and acronyms, which are not registered in the lexicon.)

For example, a sequence of three syllables s1, s2, s3 involves six possible subsequences s1, s2, s3, s1-s2, s2-s3, s1-s2-s3, which can correspond to some words. Each subsequence could correspond to more than one word, especially in the case of monosyllables. Accordingly, a word lattice is formed by the words with one of the six subsequences as pronunciation. See Figure 1 for a sample word lattice.

Note that syllables are chosen as input units instead of word-sized units used in systems like TianMa. The major reason is: Chinese is a monosyllabic language; characters/syllables are the most natural units, while "words" are not well-defined in Chinese. It is difficult for people to segment the words correctly and consistently, especially according to the dictionary provided by the system. This is also the reason why newer intelligent Chinese input methods in Taiwan like Hanin, WangXing, and Going, all use syllables (for a sentence) as input units. In addition, our target system is an isolated-syllables speech recognition system.

3 The Baseline System

The proposed system (Figure 2) is composed of a baseline system plus two new features: character-preference learning (CPL) and pseudo word learning (PWL).

The baseline syllable-to-character converter consists of three components: (1) a word hypothesizer, (2) a word-lattice search algorithm, and (3) a score function. The basic model used in our system is: (1) a Viterbi search algorithm, (2) a lexicon-based word hypothesizer, and (3) a score function considering word length and word frequency.

The word hypothesizer matches the current input syllable candidates with the lexical entries in the lexicon (7,953 1-character words, 25,567 2-character, 12,216 3-character, 12,419 4-character, 58,155 words totally). All matched words are proposed as word hypotheses forming the word lattice. Currently, we consider only those words with at most four syllables (only less than 0.1% of words contain five or more syllables). In addition, Determinative-Measure (DM)

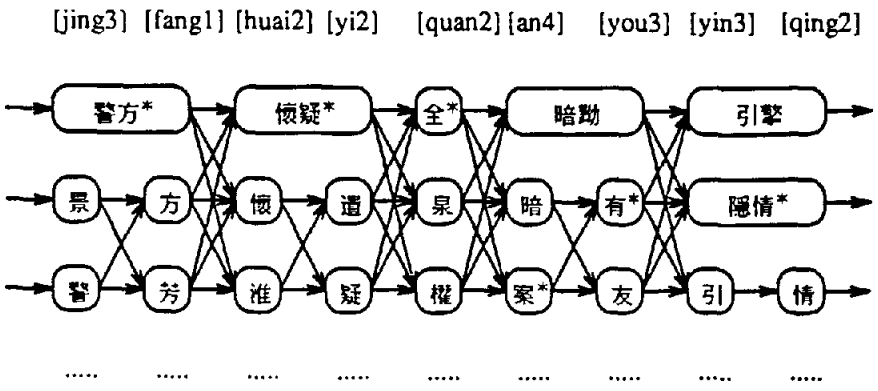

Figure 1: A Sample Word Lattice (*: correct words, ...: more monosyllabic words)

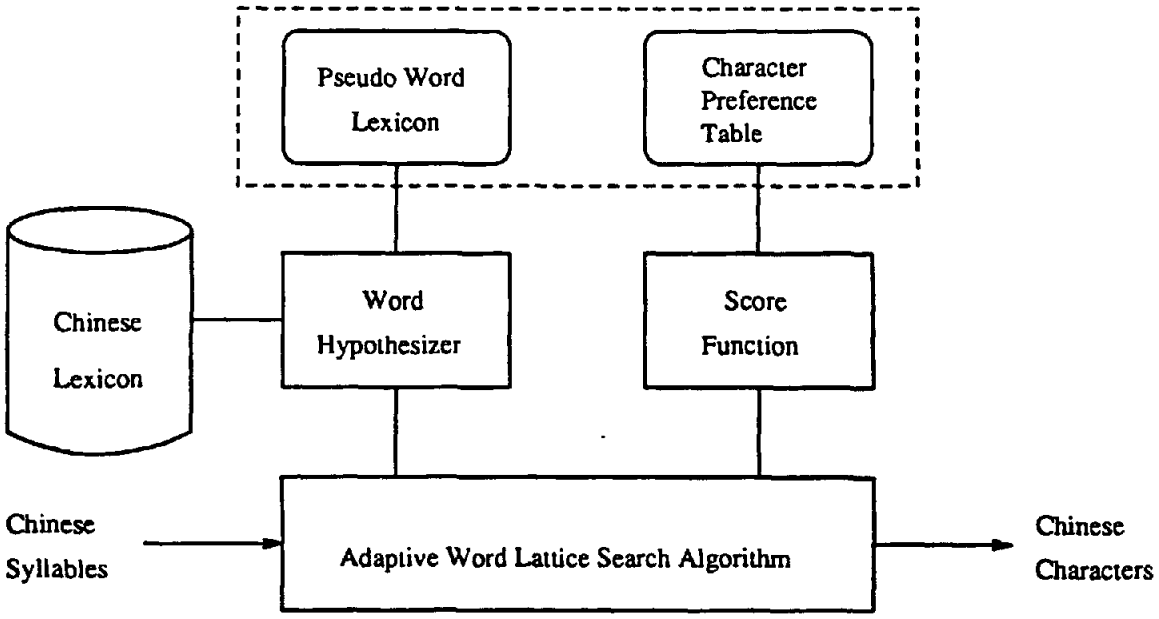

Figure 2: The Overall Structure

compounds are proposed dynamically, i.e., not stored in the lexicon.

Viterbi search is a well-known algorithm for optimal path-finding problems. The word lattice for a whole clause (delimited by a punctuation) is searched using the dynamic-programming-style Viterbi algorithm.

The score function is defined as follows: If a path P is composed of n words w_1, ..., w_n and two assumed clause delimiters w_0 and w_{n+1}, the path score for P is the sum of word scores for the n words and inter-word link scores for the n+1 word links (n−1 between-word links and 2 boundary links).

$$pathscore(P) = \sum_{i=1}^{n} wordscore(w_i) + \sum_{i=0}^{n} linkscore(w_i, w_{i+1})$$

The word score of a word is based on the word frequency statistics computed by counting the number of occurrences the word appears in the 10-million-character UD corpus. The word frequency is mapped into an integral score by taking its logarithm value and truncating the value to an integer. Lee *et al.* [11] recently presented a novel idea called *word-lattice-based Chinese character bigram* for Chinese language modeling. Basically, they approximate the effect of word bigrams by applying character bigrams to the boundary characters of adjacent words. The approach is simple (easy to implement) and very effective. Following the idea, we built a Chinese character bigram based on the UD corpus and used it to compute inter-word link scores. For two adjacent words, the last character of the first word and the first character of the second word are used to consult the character bigram which recorded the number of occurrences in the UD corpus. Inter-word link scores are then computed similarly to word scores.

4 Bidirectional Conversion and Automatic Evaluation

Here, we will only briefly review the concepts of bidirectional conversion and automatic evaluation [1,2]. For more details, see the cited papers.

Homophone disambiguation can be considered as a process of syllable-to-character (S2C) conversion. Its reverse process, character-to-syllable (C2S) conversion, is also nontrivial. There are more than 1000 characters, so-called Poyinzi (homographs), with multiple pronunciations. However, a high-accuracy C2S converter is achievable. Using an n-gram lookahead scheme, we have designed such a converter with 99.71% accuracy. Because of the high accuracy, the C2S converter can be used to convert a text corpus to a syllable corpus automatically. The two processes together form a bidirectional conversion model. The point is: If we ignore the 0.29% error (could be reduced if a better C2S system is used), many applications of the model appear.

We have applied the bidirectional model to automatic evaluation of language models for speech recognition. A more straightforward application is automatic evaluation of the S2C converter. A text is converted into a syllable sequence, which then is converted back to an output text. Comparing the input text with the output, we can compute the accuracy of the S2C converter automatically.

5 Corpus-based Adaptation Mechanisms

In the following, we describe how to apply the model to user-adaptation of homophone disambiguator.

5.1 Character-Preference Learning

Everyone has his own preference for characters and words. A chemist might use the special characters for chemical elements frequently. Different people uses a different set of proper names that are usually not stored in the lexicon. In this section, we propose an adaptation method based on the bidirectional conversion model.

From a sample text given by the user, the system first converts it to a sequence of syllables. Then, the baseline system is used to convert them back to Chinese characters. After that, we can compare them with the input to obtain the error records. From the comparison report, we will derive three indices for each character in the character set (say, 13,051 characters in the Big-5 coding used in Taiwan): A-count, B-count, and C-count. A-count is defined as

the number of times that the character is misrecognized. B-count the number of times it is wrongly used, while C-count the number of times it is correctly recognized. For example, if the user wants to input the string 李真真 and keys in the corresponding syllables [li3][zhen1][zhen1] while the output is 李珍珍, the indices would be: A(李)=0, B(李)=0, C(李)=1, A(真)=2, B(真)=0, C(真)=0, A(珍)=0, B(珍)=2, C(珍)=0. From these indices, we propose a character-preference learning procedure:

1. Convert the given sample text I_c into a syllable file I_s, using the character-to-syllable converter. Let the baseline version be V_0. Run V^0 with I_s to obtain an output O^0. From I_c and O^0, compute the initial accuracy a^0.
2. Initialize the 13051-entry character-preference table CPT^0 to zeroes. Set n to 1.
3. From I_c and O^{n-1}, compute the A, B, C indices for each character.
4. For each character c, add to the corresponding entry in CPT^{n-1} a preference score (according to a preference adjustment function pf of A(c), B(c), C(c)) to form CPT^n
5. Form a new version V^n of the syllable-to-character converter by considering CPT^n. Run V^n with I_s to obtain a new output O^n.
6. From I_c and O^n, compute the new accuracy rate a^n.
7. If $a^n > a^{n-1}$, set n to $n+1$ and repeat steps 3-6. Otherwise, stop and let CPT^{n-1} be the final CPT for the user.

Adjustment Functions

In step 4, the adjustment function pf is a function of A(c), B(c), C(c). Several versions have been tried in our experiments. Three of them are:

$$pf(c) = \begin{cases} +1 & \text{if } A(c) - B(c) > 0 \\ -1 & \text{if } A(c) - B(c) < 0 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$pf(c) = \begin{cases} +1 & \text{if } A(c) - B(c) + C(c) > 0 \\ -1 & \text{if } A(c) - B(c) + C(c) < 0 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$pf(c) = \begin{cases} +1 & \text{if } A(c) - B(c) + C(c) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Pf (1) is intuitive but easily suffers from over-training since it only considers error cases. To avoid the problem, we devise a new pf (2) taking the correct cases into account. After trying several combinations of A, B, C for pf, we observe that positive learning (3) is most effective, i.e., achieving the highest accuracy. Therefore, in the current implementation, pf (3) is used.

5.2 Pseudo Word Learning

The second adaptation mechanism is to treat error N-grams as new words (called *pseudo words*). An error N-gram is defined as a sequence of N characters in which at least (N - 1) characters are wrongly converted (from syllables) by the system. (In practice, $2 \leq N \leq 4$.) For example, if [fan4][zhen4][he2] (to input 范振和) is converted to 犯鎮和, three pseudo words are produced: 范振, 振和, and 范振和. There are two modes for generating pseudo words: corpus training and interactive correction. In the former, the user-specific text corpus (or simply a sample text) is used for generating the pseudo word lexicon (PW lexicon), applying the concept of bidirectional conversion. In the latter, pseudo words are produced through user corrections in an interactive input process. Both modes can be used at the same time.

In the following, we will describe how to build, maintain, and use the user-specific PW lexicon. The PW lexicon stores the M (lexicon size) pseudo words that are produced or referenced in the most recent period. It is structurally exactly the same as the general lexicon, containing Hanzi, phonetic code, and word frequency. The word frequency of a new PW is set to f_0 (3 in the implementation) and incremented by one when referenced. Once the word frequency exceeds an upper bound F, the PW would be considered as a real word and no longer liable to replacement.

The procedure is:

1. Segment the sample text into clauses (separated by punctuations). For each clause I_c, do steps 2-4.
2. Convert the clause into syllable sequence I_s using C2S, then convert I_s back to a character se-

quence O_c using baseline S2C. For each character C_n in O_c, do steps 3-4.

3. Compare C_n with the corresponding input character. Set the error flag if different.
4. If a pseudo word ending with C_n is found (according to error flags) then (1) increment the word frequency if it is already in the PW lexicon, and check the upper bound F; (2) replace the old entry and set frequency to f_0 if the lexicon has a homophone PW; (3) add a new entry if the lexicon has vacancies; (4) otherwise, remove one of the entries that have the least word frequency and add the new PW.
5. We have a new PW lexicon after the above steps are done.

We observe that 3-character pseudo words are very useful for dealing with the unregistered proper name problem, which is a significant source of conversion errors. The reasons are: (1) A large part of unknown words in news articles are proper names, especially three-character personal names; (2) It is not practical to store all the proper names beforehand; (3) The proper names usually contain uncommon characters which are difficult to convert from syllables. Therefore, the user (or author) can have a personalized PW lexicon which contains unregistered proper names he will use, simply by providing a sample text.

The parameters for both CPL and PWL can be trained by the bidirectional learning procedure. The only input the user needs to provide is a sample text similar to the texts he wants to input by the phonetic-input-to-character converter. The phonetic input file will be automatically generated by the character-to-syllable converter.

6 Experimental Results

6.1 The Corpora

Eleven sets of newspaper articles are extracted from the 1991 United Daily News Corpus (kindly provided by United Informatics, Inc., Taiwan). Ten of them are by specific reporters, i.e., one set per reporter. The other is translated AP News. These corpora are used to validate the proposed adaptation techniques.

We design an extraction procedure to select articles written by a specific reporter from the corpus with more than 10 million characters. First, collect character-trigrams after the word 記者 (ji4zhe3,'reporter') and sort them according to the number of occurrences. Most of these trigrams happen to be names of reporter. We use the top-10 names as the basis for selecting articles. Then, search the names in the corpus in order to built the article databases for the 10 reporters. The AP News corpus is built in a similar way (searching for the word 美聯社 mei3lian2she4). Table 1 lists some statistics for the article databases. The first column lists the set names, the second column the numbers of articles in the set, the third column the numbers of characters, and the fourth column the numbers of pronounceable characters.

Set	*#Articles*	*#Char1*	*#Char2*
lwy	83	35,000	31,163
lkq	52	22,424	20,214
llg	47	16,178	14,677
lxd	49	19,429	17,368
yft	44	18,647	16,423
yxj	45	14,791	13,132
fzh	45	17,154	15,299
tdc	44	15,323	13,638
lsq	44	18,804	17,150
ljn	44	19,065	17,122
ap	408	122,554	109,218

Table 1: The Article Databases

Each corpus is then divided into two parts according to publication date: a training set and a testing set. For example, the corpus `lwy` is divided into `lwy-1` and `lwy-2`.

In the following, we show the experimental results for training sets and testing sets, respectively.

6.2 Training Sets

Table 2 shows the adaptation results for the training sets. The R1 column lists the accuracy rates for the baseline system, while the R2 column lists those for the adapted (or personalized) system. To avoid the problem of over-training, we train the the system only by two iterations in practice. More iterations can improve the performance for training sets but hurt the performance for testing sets. The average character accuracy rate is improved by 4.68% (from 93.48% to 98.16%). That is, 71.8 percent of errors are eliminated.

Set	R1	R2	R2-R1	Ratio
lwy-1	94.32	98.35	4.03	70.9%
lkq-1	93.07	97.85	4.78	68.9%
llg-1	94.19	98.42	4.23	72.8%
lxd-1	95.49	98.53	3.04	67.4%
yft-1	94.75	98.32	3.57	68.0%
yxj-1	91.83	98.23	6.40	78.3%
fzh-1	91.20	98.15	6.95	78.9%
tdc-1	92.13	97.71	5.58	70.9%
lsq-1	93.63	98.23	4.60	72.2%
ljn-1	93.64	98.09	4.45	69.9%
ap-1	94.04	97.86	3.82	64.0%
Ave.	93.48	98.16	4.68	71.8%

Table 2: Accuracy Rates for Training Sets

6.3 Testing Sets

Table 3 shows the results for the testing sets. The average accuracy is improved by 1.34% (from 93.46% to 94.80%). That is, 20.5 percent of errors are eliminated.

Set	R1	R2	R2-R1	Ratio
lwy-2	94.06	95.00	0.94	15.8%
lkq-2	93.81	95.95	2.14	34.6%
llg-2	93.38	94.18	0.80	12.1%
lxd-2	95.97	96.42	0.45	11.2%
yft-2	95.21	96.17	0.96	20.0%
yxj-2	91.67	93.93	2.26	27.1%
fzh-2	91.28	92.29	1.01	11.6%
tdc-2	92.25	93.58	1.33	17.2%
lsq-2	92.86	94.27	1.41	19.7%
ljn-2	93.66	95.52	1.86	29.2%
ap-2	93.93	95.54	1.61	26.5%
Ave.	93.46	94.80	1.34	20.5%

Table 3: Accuracy Rates for Testing Sets

7 Related Work

The study of phonetic-input-to-character conversion has been quite active in the recent years. There are two different approaches for the problem: dictionary-based and statistics-based.

Matsushita (Taipei) developed a Chinese word-string input system, Hanin, as a new input method (Chen [4]) in which phonetic symbols are continuously converted to Chinese characters through dictionary lookup. Commercial systems TianMa and WangXing (ETen Corp.) also belong to this type. In the mainland, there have been several groups involving in similar projects [14,15] although most of them are pinyin-based and word-based.

In the statistics-based school are relaxation techniques (Fan and Tsai [6]), character bigrams with dynamic programming (Sproat [12]), constraint satisfaction approaches (JS Chang [3]), and zero-order or first-order Markov models (Gu *et al.* [7]).

Ni [9] mentioned a so-called self-learning capability for his Chinese PC called LX-PC. However, the method is (1) let the user define new words during the input process (2) dynamically adjust the word frequency of used words. Chen [4] also proposed a learning function that uses a learning file to store user-selected characters and words and the character before them. The entries in the learning file are favored over those in the regular dictionary. Lua and Gan [8] describe a simple error-correcting mechanism: increase the usage frequency of the desired word by 1 unit when the user corrects the system's output. These methods are either manual adaptation or simple word frequency counting.

Recently, Su *et al.* [5,13] proposed a discrimination oriented adaptive learning procedure for various problems, e.g., speech recognition, part-of-speech tagging, and word segmentation. The basic idea is: When an error is made, i.e., the first candidate is not correct, adjust the parameters in the score function based on subspace projection. The parameters for the correct candidate are increased, while those for the first candidate are decreased, both in an amount decided by the difference between the scores of the two candidates. This process continues until the correct candidate becomes the new first candidate; that is, the score of the correct candidate is greater than that of the old first one. Our learning procedure is different from theirs because (1) ours is increment-based while theirs is projection-based, (2) ours is not discrimination oriented, (3) ours is coarse-grained learning while theirs is fine-grained, and (4) the application domain is different.

8 Concluding Remarks

We have presented two corpus-based adaptation mechanisms for Chinese homophone disambiguation: character-preference learning and pseudo-word learn-

ing. Experimental results show that the error rates have been reduced significantly. This proves yet another success of corpus-based NLP research.

Future works include (1) more experiments using various texts, (2) study on more effective adjustment functions for CPL, (3) study on weighting of different lengths of pseudo words, (4) adaptation based on other parameters, e.g., parts-of-speech, semantic categories, and (5) application to linguistic decoding for speech recognition.

Acknowledgements

The author is grateful to the Chinese Lexicon group (CCL/ITRI) for the 90,000-word lexicon. This paper is a partial result of the project no. 37H2100 conducted by the ITRI under sponsorship of the Minister of Economic Affairs, R.O.C.

References

[1] C.-H. Chang. Bidirectional conversion between Mandarin syllables and Chinese characters. In *Proc. of 1992 International Conference on Computer Processing of Chinese and Oriental Languages*, Florida, USA, 1992.

[2] C.-H. Chang. Design and evaluation of language models for Chinese speech recognition. In *Proc. of 1992 CMEX Workshop on Chinese Speech Recognition*, Taipei, Taiwan, November 1992.

[3] J.-S. Chang, S.-D. Chen, and C.-D. Chen. Conversion of phonetic-input to Chinese text through constriant satisfaction. In *Proc. 1991 ICCPCOL*, pages 30–36, Taipei, 1991.

[4] S.-I. Chen, C.-T. Chang, J.-J. Kuo, and M.-S. Hsieh. The continuous conversion algorithm of Chinese character's phonetic symbols to Chinese character. In *Proc. of National Computer Symposium*, pages 437–442, 1987.

[5] T.-H. Chiang, J.-S. Chang, M.-Y. Lin, and K.-Y. Su. Statistical models for word segmentation and unknown word resolution. In *Proc. ROCLING V*, pages 121–146, Taipei, Taiwan, September 1992.

[6] C.-K. Fan and W.-H. Tsai. Relaxation-based word identification for removing the ambiguity in phonetic Chinese input. *Int. J. of Pattern Recognition and Artificial Intelligence*, 4(4):651–666, 1990.

[7] H.-Y. Gu, C.-Y. Tseng, and L.-S. Lee. Markov modeling of Mandarin Chinese for decoding the phonetic sequence into Chinese characters. *Computer Speech and Language*, 5:363–377, 1991.

[8] L.-S. Lee et al. Golden Mandarin (II) - an improved single-chip real-time Mandarin dictation machine for Chinese language with very large vocabulary. In *Proc. 1993 ICASSP*, pages II:503–506, April 1993.

[9] K.T. Lua and K.W. Gan. A touch-typing Pinyin input system. *Computer Processing of Chinese & Oriental Languages*, 6(1):85–94, June 1992.

[10] G. Ni. A Chinese PC features intelligent Hanzi input. In *Proc. 1986 Int. Conf. on Chinese Computing*, pages 155–159, August 1986.

[11] ROC Computational Linguistics Society. *Proc. of Workshop on Corpus-based Researches and Techniques for Natural Language Processing*, Taipei, Taiwan, September 1992.

[12] R. Sproat. An application of statistical optimization with dynamic programming to phonetic-input-to-character conversion for Chinese. In *Proc. ROCLING III*, pages 379–390, September 1990.

[13] K.-Y. Su and C.-H. Lee. Robustness and discrimination oriented speech recognition using weighted HMM and subspace projection approaches. In *Proc. ICASSP91*, pages 541–544, Toronto, Ontario, Canada, May 1991.

[14] X. Wang, K. Wang, and X. Bai. Separating syllables and characters into words in natural language understanding. *Journal of Chinese Information Processing*, 5(3):48–58, 1991.

[15] X. Zhong. A multiple phrase pinyin/Hanzi conversion mechanism. *Journal of Chinese Information Processing*, 4(2):55–64, 1990.