

The Properties and Further Applications of Chinese Frequent Strings

Yih-Jeng Lin^{*}, and Ming-Shing Yu⁺

Abstract

This paper reveals some important properties of CFSs and applications in Chinese natural language processing (NLP). We have previously proposed a method for extracting Chinese frequent strings that contain unknown words from a Chinese corpus [Lin and Yu 2001]. We found that CFSs contain many 4-character strings, 3-word strings, and longer n-grams. Such information can only be derived from an extremely large corpus using a traditional language model(LM). In contrast to using a traditional LM, we can achieve high precision and efficiency by using CFSs to solve Chinese toneless phoneme-to-character conversion and to correct Chinese spelling errors with a small training corpus. An accuracy rate of 92.86% was achieved for Chinese toneless phoneme-to-character conversion, and an accuracy rate of 87.32% was achieved for Chinese spelling error correction. We also attempted to assign syntactic categories to a CFS. The accuracy rate for assigning syntactic categories to the CFSs was 88.53% for outside testing when the syntactic categories of the highest level were used.

Keywords: Chinese frequent strings, unknown words, Chinese toneless phoneme-to-character, Chinese spelling error correction, language model.

1. Introduction

An increasing number of new or unknown words are being used on the Internet. Such new or unknown words are called “out of vocabulary (OOV) words” [Yang 1998], and they are not listed in traditional dictionaries. Many researchers have overcome problems caused by OOV words by using N-gram LMs along with smoothing methods. N-gram LMs have many useful applications in NLP [Yang 1998]. In Chinese NLP tasks, word-based bi-gram LMs are used by many researchers. To obtain useful probabilities for training, a corpus size proportional to 80000^2 (80000 is the approximate number of words in ASCED) = 6.4×10^9 words is required.

*

Department of Information Management, Chien Kuo Institute of Technology, Changhua, 500 Taiwan
E-mail: yclin@ckit.edu.tw Tel: 04-7111111 ext 3637 Fax:04-7111142

⁺ Department of Computer Science, National Chung-Hsing University, Taichung, 40227 Taiwan

However, it is not easy to find such a corpus at the present time.

A small-size corpus will lead too many unseen events when using N-gram LMs. Although we can apply some smoothing strategies, such as Witten-Bell interpolation or the Good-turing method [Wu and Zheng 2001] to estimate the probabilities of unseen events, this will be of no use when the size of training corpus is limited. From our observations, many the unseen events that occur when using N-gram LMs are unknown words or phrases. Such unknown words and phrases cannot be found in a dictionary. For example, the term “週休二日” (two days off per week) is presently popular in Taiwan. We cannot find this term in a traditional dictionary. The term “週休二日” is a 4-word string pattern which consists of four words: “週” (a week), “休” (to rest), “二” (two), and “日” (day). A word-based 4-gram LM and a large training corpus are required to record the data of such terms. Such a word-base 4-gram LM has not been applied to Chinese NLP in practice, and such a huge training corpus cannot be found at present. Alternatively, we can record the specifics of the term “週休二日” by using a CFS with relatively limited training data in which the specified term appear two or more times. Such training data could be recorded in one or two news articles containing hundreds of Chinese characters. Many researchers have shown that frequent strings can be used in many applications [Jelinek 1990; Suhm and Waibel 1994].

We have shown that adding Chinese frequent strings (CFSs), including unknown words, can improve performance in Chinese NLP tasks [Lin and Yu 2001]. A CFS defined based on our research is a Chinese string which appears two or more times by itself in the corpus. For example, consider the following fragment:

“國立中興大學，中興大學。” (National Chung-Hsing University, Chung-Hsing University.)

“中興大學” (Chung-Hsing University) is a CFS since it appears twice and its appearances are not brought out by other longer strings. The string “中興” (Chung-Hsing) appears twice, but it is not a CFS here since it is brought about by the longer string “中興大學”.

In our previous research, we showed that adding CFSs to a traditional lexicon, such as ASCED, can reduce the normalized perplexity from 251.7 to 63.5 [Lin and Yu 2001]. We also employed CFSs combined with ASCED as a dictionary to solve some Chinese NLP problems using the word-based uni-gram language model. We achieved promising results in both Chinese CTP and PTC conversion. It is well known that using a word-based bi-gram LM with a traditional lexicon can also improve accuracy in these two cases, especially in Chinese PTC conversion.

The organization of this paper is as follows. Section 2 gives some properties and distributions of CFSs, and we also make a comparison between CFS and an n-gram LM. Section 3 shows that by using a CFS-based uni-gram LM, we can achieve higher accuracy

than we can by using a traditional lexicon with a word-based bi-gram LM. We demonstrate this by using two challenging examples of Chinese NLP. In section 4, we assign syntactic categories to CFSs. Finally, section 5 presents our conclusions.

2. The Properties of CFS

We used a training corpus of 59 MB (about 29.5M Chinese characters) in our experiments. In this section, we will present the properties of CFSs. Compared with language models and ASCED, CFSs have some important and distinctive features. We extracted 439,666 CFSs from a training corpus.

2.1 Extracting CFSs from a Training Corpus

The algorithm for extracting CFSs was proposed in our previous work [Lin and Yu 2001]. We extracted CFSs from a training corpus that contained 29.5M characters. The training corpus also included a portion of the Academia Sinica Balanced Corpus [Chen *et al.* 1996] and many Internet news texts.

The length distribution of the CFSs is shown in the second column of Table 1. The total number of CFSs that we extracted was 439,666. Our dictionary, which we call CFSD, is comprised of these 439,666 CFSs. In contrast to the second column of Table 1, we show the length distribution of the words in ASCED in the forth column of Table 1. We found that three-character CFSs were most numerous in our CFS lexicon, while two-character words were most numerous in ASCED. Many meaningful strings and unknown words are collected in our CFSs. These CFSs usually contain more than two characters. Some examples are “小企鵝” (a little penguin), “西醫師” (modern medicine), “佛教思想” (Buddhist thought), “樂透彩券” (lottery), and so on. The above examples cannot be found in ASCED, yet they frequently appear in our training corpus.

2.2 Comparing CFSs with Word-Based N-Gram LMs

Since CFSs are strings frequently used by people, a CFS like “大學教授” (professors of a university) may contain more characters than a word defined in ASCED does. That is, a CFS may contain two or more words. If a CFS contains two words, we say that this CFS is a 2-word CFS. If a CFS contains three words, we say that this CFS is a 3-word CFS and so on. Figure 1 shows the distributions of CFSs according to word-based n-grams. The words are defined in ASCED. We also found 31,275 CFSs (7.11% of the CFSs in CFSD) that are words in ASCED.

From Figure 1, it can be shown that a CFS may contain more than 3 words. Many researchers in Chinese NLP have used word-based bi-gram LMs [Yang 1998] as a basic LM to

solve problems. A very large corpus is required to train a word-based 3-gram LM, while our CFS-based uni-gram model does not need such a large corpus. We also found that a CFS contains 2.8 words on average in CFSD. This shows that a CFS contains more information than a word-based bi-gram LM. In our experiment, we also found that the average number of characters of a word-based bi-gram was 2.75, and that the average number of characters of a CFS was 4.07. This also shows that a CFS contains more information than a word-based bi-gram LM.

Table 1. The length distributions of CFSs in CFSD and words in ASCED.

Number of characters in a CFS or a word	Number of CFSs of that length in our CFS dictionary	Percentage	Number of words of that length in ASCED	Percentage
1	3,877	0.88%	7,745	9.57%
2	69,358	15.78%	49,908	61.67%
3	114,458	26.03%	11,663	14.41%
4	113,005	25.70%	10,518	13.00%
5	60,475	13.75%	587	0.73%
6	37,044	8.43%	292	0.36%
7	19,287	4.39%	135	0.17%
8	11,494	2.61%	66	0.08%
9	6,588	1.50%	3	0.004%
10	4,080	0.93%	8	0.006%

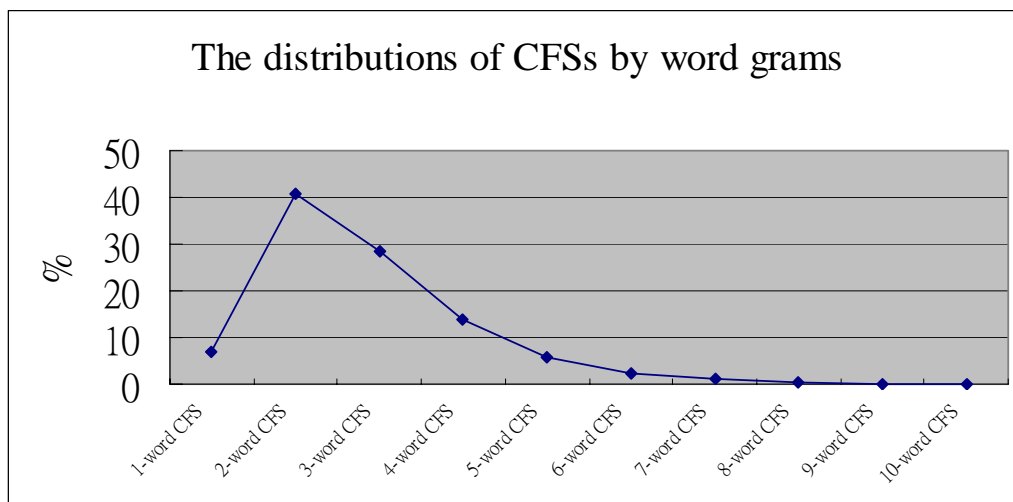


Figure 1. The distributions of CFSs by word-based grams

2.3 Compare the Distributions of CFSs and ASCED

In this subsection, we will make a comparison between our CFSs and ASCED. Table 1 and Figure 2 show the length distributions of our CFSs and ASCED. Comparing them, we find that the average number of characters in a word in ASCED is 2.36, while the average number of characters in a CFS is 4.07. Examining Figure 2, we notice that most of the words in ASCED are 2-character words, while the largest portion of CFSs are 2-character CFSs, 3-character CFSs, 4-character CFSs, and 5-character CFSs. This shows that our CFSs contain many 4-character and 5-character strings. To train character-based 4-gram and character-based 5-gram LMs requires a large training corpus. We also find that the number of one-character CFSs is fewer than that in ASCED. This shows that by using the CFSs, we can eliminate some ambiguities in Chinese PTC and Chinese CTP.

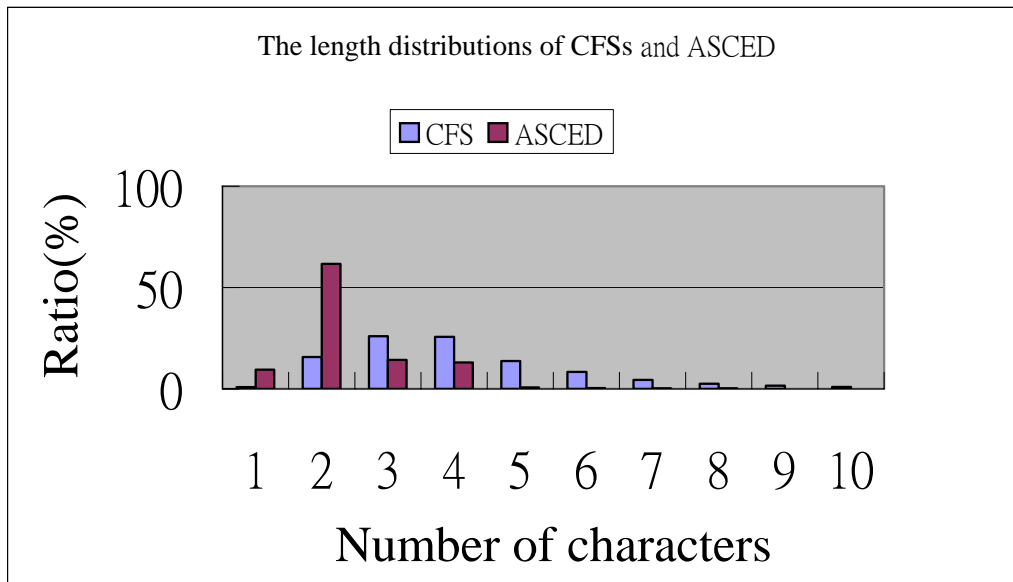
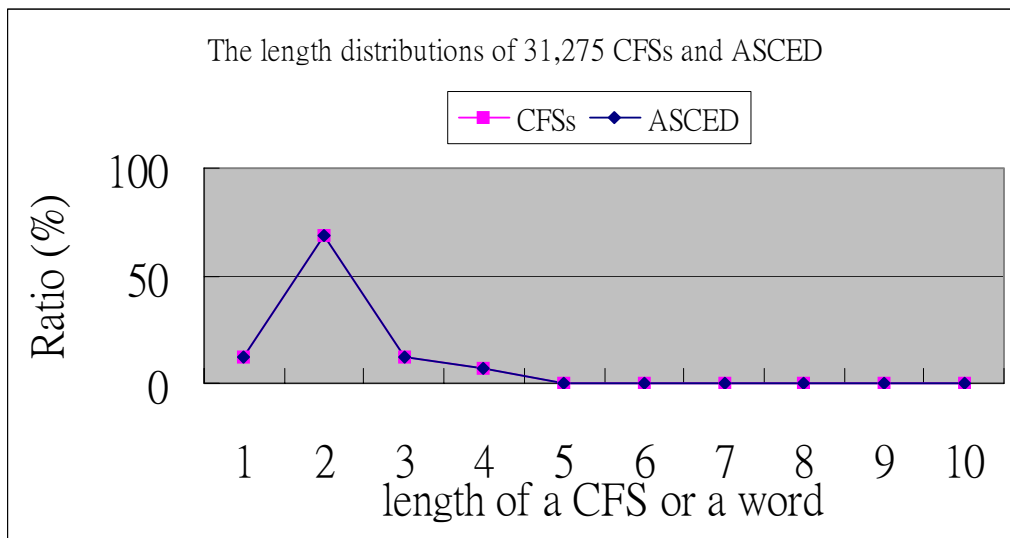


Figure 2. The length distributions of CFSs and ASCED.

We found 31,275 CFSs that were in ASCED. The length distribution of these 31,275 CFSs is shown in Table 2. We also compared the length distribution of these 31,275 CFSs with the length distribution in ASCED. Our comparison is shown in Figure 3. Note that the length distribution in ASCED is listed in the fifth column of Table 1. We find that the length distribution of these 31,275 CFSs is similar to the length distribution in ASCED. We conjecture that if the corpus is large enough, we can find most of the words in ASCED.

Table 2. The length distribution of 31,275 CFSs.

Number of characters in a CFS	Number of CFSs	Percentage
1	3,877	12.40%
2	21,411	68.46%
3	3,742	11.96%
4	2,089	6.68%
5	115	0.37%
6	33	0.105%
7	7	0.022%
8	1	0.003%
9	0	0%
10	0	0%

**Figure 3. The length distributions of 31,275 CFSs and ASCED.**

2.4 Comparing the Normalized Perplexity

Perplexity [Rabiner and Juang 1993] is an important and commonly used measurement of language models. Formula (1) provides a definition of perplexity. Since N_w , which is the number of words in the test corpus, in (1) is uncertain for Chinese, we normalize the

perplexity into characters by means of (2) [Yang 1998], producing is called the normalized perplexity (or relative perplexity):

$$PP = \Pr(W_1^{Nw})^{\frac{1}{Nw}}, \quad (1)$$

$$\text{where } \Pr(W_1^{Nw}) = \Pr(w_1) \bullet \Pr(w_2) \bullet \dots \bullet \Pr(w_{Nw}),$$

$$NP = PP^{\frac{Nw}{L(W)}}. \quad (2)$$

Here, $W_1^{Nw} = w_1 w_2 \dots w_{Nw}$ is the test sequence of the corpus and $\Pr(W_1^{Nw})$ is the probability that W_1^{Nw} will be computed within a given language model. $L(W)$ is the number of characters in W . PP is perplexity, and NP is the normalized perplexity.

We used a testing corpus to compute the normalized perplexities within the CFS-based uni-gram LMs and the word-based bi-gram LMs. The size of the testing corpus was 2.5M characters. We used the same training corpus mentioned in subsection 2.1 to extract CFSs and to train the word-based bi-gram LMs. Each word in the word-based bi-gram LM was defined in ASCED. We used the Good-Turing smoothing method to estimate the unseen bi-gram events. The normalized perplexity obtained using the word-based bi-gram LM was 78.6. The normalized perplexity became 32.5 when the CFS-based uni-gram LM was used. This shows that the CFS-based uni-gram LM has a lower normalized perplexity. That is to say, using the CFS-based uni-gram LM is better than using the traditional word-based bi-gram LM with a small-sized training corpus of 29.5M characters.

3. Application of CFS to Two Difficult Problems

In a previous study [Lin and Yu 2001], we showed that using CFSs and ASCED as the dictionary with the uni-gram language model can lead to good results in two Chinese NLP applications. These two applications are Chinese character-to-phoneme (CTP) conversion and Chinese phoneme-to-character (PTC) conversion. The achieved accuracy rates were 99.7% for CTP conversion and 96.4% for PTC conversion [Lin and Yu 2001]. The size of the training corpus in our previous research was 0.5M characters. There were 55,518 CFSs extracted from the training corpus. In this study, we solved two challenging Chinese NLP problems with a larger training corpus. The two problems were Chinese toneless phoneme-to-character (TPTC) conversion and Chinese spelling error correction (SEC).

The first task was Chinese TPTC conversion. Chinese TPTC tries to generate correct characters according to input syllables without tonal information. The second task was Chinese SEC (spelling error correction). In our study, we attempted to identify and correct the

possible errors in sentences with no more than one error that were input using the Cang-Jie (倉頡) input method.

3.1 Chinese Toneless Phoneme-to-Character Conversion

The first task was Chinese TPTC conversion. The lexicon we used was CFSD as mentioned in section 2.1. This task is more complex than traditional Chinese phoneme-to-character conversion. There are five tones in Mandarin. They are high-level (1st tone), high-rising (2nd tone), low-dipping (3rd tone), high-falling (4th tone), and the neutral tone [National Taiwan Normal University 1982]. There are a total of 1,244 possible syllables (combinations of phonetic symbols) in Mandarin, and there are a total of 408 possible toneless syllables. Therefore, each toneless syllable has about $1,244/408=3.05$ times the number of characters of a tonal syllable. The average length of a sentence in our training corpus is 8 characters per sentence. The number of possibilities for Chinese TPTC conversion is about $3.05^8=7489$ times that of Chinese PTC conversion. This shows that Chinese TPTC conversion is more difficult than Chinese PTC conversion.

The size of the outside testing data was 2.5M characters. In our TPTC module, we initially searched the system dictionary to access all the possible CFSs according to the input toneless phonemes. Such possible CFSs constitute a CFS lattice. We applied a dynamic programming methodology to find the best path in the CFS lattice, where the best path was the sequence of CFS-based uni-grams with the highest probability. The definition we employed of the probability $P(S)$ of each input sentence S was as follows:

$$S = CFS_1 CFS_2 \dots CFS_n,$$

$$P(S) = P(CFS_1) \cdot P(CFS_2) \cdot \dots \cdot P(CFS_n), \quad (3)$$

The achieved precision rate was 92.86%. The precision rate was obtained by using the formula (total number of correct characters) / (total number of characters). The processing time was 12 ms/character. We also applied the dictionary used in our previous research [Lin and Yu 2001] to test the data, which was 2.5M characters in size. The dictionary combines ASCDE with 55,518 CFSs. The achieved precision rate in solving the Chinese TPTC problem was 87.3%. This indicates that if we can collect more CFSs, we can obtain higher accuracy.

In this task, we also applied the word-based bi-gram LM with ASCED. The size of the training corpus was the same as that of the corpus mentioned in section 2.1, that is, 29.5M characters. The Good-Turing smoothing method was applied here to estimate the unseen events. The achieved precision rate was 66.9%, and the processing time was 510 ms/character. These results show that when the CFS-based uni-gram LM was used, the precision rate improved greatly (92.8 % vs. 66.9%) and the processing time was greatly reduced (12

ms/character vs. 510 ms/character) compared to the results obtained using the traditional word-based bi-gram LM.

3.2 The Chinese Spelling Error Correction Issue

We also applied the CFS-based uni-gram LM to the Chinese SEC problem [Chang 1994]. Chinese SEC is a challenging task in Chinese natural language. A Chinese SEC system should correct character errors in input sentences. To make the task meaningful in practice, we limited our Chinese SEC problem based on the following constraints: (1) the sentences were input using the Cang-Jie Chinese input method; (2) there was no more than one character error in an input sentence.

The reasons why we applied the above two constraints are as follows: (1) our Chinese SEC system is designed for practiced typists; (2) the Cang-Jie Chinese input method is a popular method widely used in Taiwan; (3) at most one character error is likely to be made in a sentence by a practiced typist; and (4) we can easily apply the methodology used this research to other Chinese input or processing systems. Our methodology for Chinese SEC is shown in Algorithm SEC.

Characters with similar Cang-Jie codes define a confusing set in Algorithm SEC. We constructed the confusing set for each Chinese character based on the five rules listed in Table 3. The longest common subsequence (LCS) algorithm is a well known algorithm that can be found in most computer algorithm textbooks, such as [Cormen *et al.* 1998].

Algorithm SEC.

Input: A sentence S with no more than one incorrect character.

Output: The corrected sentence for the input sentence S .

Algorithm:

- Step 1: For each i -th character in S , find the characters whose Cang-Jie codes are similar to the code of the i -th character. Let C be the set consisting of such characters. C is called the ‘confusing set’.
- Step 2: Replace each character in C with the i -th character in S . There will be a ‘maybe’ sentence S_i . Find the probability of S_i by using the CFS-based uni-gram LM. Record the maybe sentence with the highest probability.
- Step 3: For each character in S , repeat Step 1 and Step 2.
- Step 4: Output the ‘maybe’ sentence with the highest probability found in Steps 1, 2, and 3.

Table 3. Rules used to construct the confusing set based on the Cang-Jie Chinese input method.

Length of Cang-Jie code to the target character t	Each character s satisfying the conditions listed below is a similar character of t .
1	The characters whose Cang-Jie codes are the same as that of the target character.
2	A. The length of the Cang-Jie code of s is 2, and the length of the LCS of s and t is 1. B. The length of the Cang-Jie code of s is 3, and the length of the LCS of s and t is 2.
3	The length of the Cang-Jie code of s is greater than 1, and the length of the LCS of s and t is 2.
4	The length of the Cang-Jie code of s is greater than 2, and the length of the LCS of s and t is 3.
5	The length of Cang-Jie code of s is 4 or 5, and the length of the LCS of s and t is 4.

The uni-gram language model was used to determine the probability of each sentence. We used CFSD as our dictionary. There were 485,272 sentences for the outside test. No more than one character in each sentence was replaced with a similar character. Both the location of the replaced character and that of the similar character were randomly selected. The achieved precision rate was 87.32% for the top one choice. The precision rate was defined as (the number of correct sentences) / (the number of tested sentences). The top 5 precision rates are listed in Table 4. The precision rate of the top 5 choices was about 95%, as shown in Table 4. This shows that our approach can provide five possible corrected sentences for users in practice. The achieved precision rate in determining the location of the replaced character with the top one choice was 97.03%.

Table 4. The precision rates achieved using our Chinese SEC and the CFS-based uni-gram LM.

Top n	Precision rate
1	87.32%
2	90.82%
3	92.66%
4	93.98%
5	94.98%

We also applied ASCDE with word-based bi-gram LMs to compute the probability for each possible sentence. The size of the training corpus was 29.5M characters, which was the same as that of the training corpus mentioned in section 2.1. We also used the Good-Turing

smoothing method to estimate the unseen bi-gram events. The achieved precision rates are shown in Table 5. The achieved precision rate for the top one choice was 80.95%.

Table 5. The precision rates achieved using the Chinese SEC and the word-based bi-gram LM.

Top n	Precision rate
1	80.95%
2	82.58%
3	83.31%
4	83.77%
5	84.09%

From Table 4 and Table 5, we can find that using CFS-based uni-gram LM is better than using ASCED with a word-based bi-gram LM. The advantages are the high achieved precision rate (87.32% vs. 80.95%) and short processing time (55 ms/character vs. 820 ms/character).

4. Assigning Syntactic Categories to CFSs

A CFS is a frequently used combination of Chinese characters. It may be a proper noun, like “網際網路” (the Internet), a verb phrase, like “全力動員投入” (try one’s best to mobilize), and other word forms. If a CFS can be assigned to some syntactic categories, it can be used in more applications. The CYK algorithm is a well known method used to assign syntactic categories [Lin 1994]. In this study, we tried to assign syntactic categories to CFSs by a using dynamic programming strategy. If a CFS s is also a word w , we can assign the syntactic categories of w to s . When s is a combination of several words, we can attempt to find syntactic categories associated with it. We first find the probabilities of production rules. Then, we use these probabilities to determine the syntactic categories.

4.1 Extracting Production Rules from Sinica Treebank Version 1.0

We used the Sinica Treebank [Chen *et al.* 1994] as the training and testing data. The contents of the Sinica Treebank are composed of the structural trees of sentences. Structural trees contain the forms of words, the syntactic categories of each word, and the reductions of the syntactic categories of words. Figure 4 shows the structural tree of the sentence “你要不要這幅畫” (Do you want this picture?). The representation of this structural tree in the Sinica Treebank is as follows:

```
#S((agent:NP(Head:Nhaa: 你))|(Head:VE2(Head:VE2: 要))|(negation:Dc: 不))|(Head:VE2: 要))|(goal:NP(quantifier:DM: 這幅))|(Head:Nab: 畫)))#
```

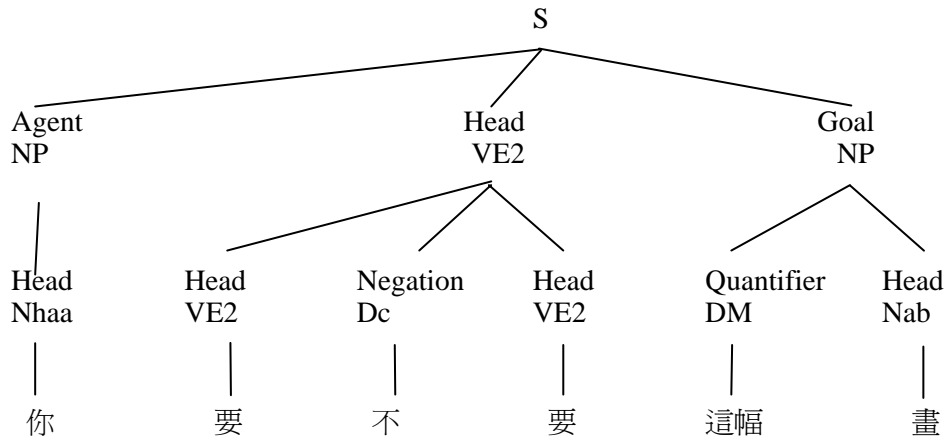


Figure 4. The structural tree of the sentence “你要不要這幅畫” (Do you want this picture?)

There are 38,725 structural trees in the Sinica Treebank version 1.0. They are stored in 9 files. We first used a portion of the 38,725 structural trees as the training data. We wanted to extract the production rules from each structural tree. These production rules were used to determine the syntactic categories of CFSs. Since each CFS could contain one or more words, the syntactic category of a CFS could be a portion of the structural tree. For example, four different production rules were extracted from the structural tree shown in Figure 4. They are “NP←Nhaa”, “VE2←VE2+Dc+VE2”, “NP←DM+Nab”, and “S←NP+VE2+NP”. The notations of syntactic categories are defined by the Chinese Knowledge Information Processing group (CKIP).

Examples of probabilities of production rules are listed in Table 6. We extracted 15,946 different production rules from 90% of the Sinica Treebank version 1.0. The other 10% of the structural trees are left for testing.

Table 6. Examples of production rules and their corresponding probabilities.

	Rule	Count	Probability
ADV	← A	1	1
ADV	← Dbaa	4	1
S	← Cbaa + S	15	0.9375
VP	← Cbaa + S	1	0.0625
NP	← NP + A + Nab	5	1
S	← Cbba + NP + VJ3	1	0.5
VP	← Cbba + NP + VJ3	1	0.5
NP	← NP + VG2 + NP	1	0.008
S	← NP + VG2 + NP	111	0.941
VP	← NP + VG2 + NP	6	0.051

4.2 Determining the Syntactic Categories of a CFS

We used the 15,946 production rules to determine the syntactic categories of CFSs. To perform this task, a lexicon with syntactic categories was required for each word. We used ASCED, provided by Academia Sinica, Taiwan, as the dictionary. ASCED is a well-defined dictionary which contains about 80,000 words. For an input CFS, we first looked in ASCED to get the syntactic categories for each substring word of the input CFS. We also used these syntactic categories and the 15,946 production rules to determine the syntactic categories of the input CFS. We tried to find the syntactic categories of a CFS by using the syntactic categories of the substrings of that CFS. The method we used is a dynamic programming method. As an example, Figure 5 shows the syntactic categories of the CFS “林小姐” (Miss Lin).

	1(林)	2(小)	3(姐)
A(林)	Nab, 0.5 Nbc, 0.5	NP, 1	NP, 1
B(小)		VH13, 0.25 V3, 0.25 Nv4, 0.25 VH11, 0.25	Nab, 1
C(姐)			B, 1

Figure 5. The syntactic categories of the CFS “林小姐” (Miss Lin).

As shown in Figure 5, we first looked in ASCED to find the syntactic categories of each possible word which was a substring of “林小姐”. Cell (A,1) contains the possible syntactic categories of the word “林”, cell (B,2) contains the possible syntactic categories of “小”, cell (C,3) contains the possible syntactic categories of “姐”, and cell (B, 3) contains the possible syntactic categories of “小姐”. The number following each syntactic category in a cell is the probability of that syntactic category.

Next, we tried to determine the syntactic categories of cell (A, 2) by using the production rules we extracted from the Sinica Treebank. The syntactic categories of cell (A, 2) could be derived using the information of cell (A, 1) and cell (B, 2). A total of $2 * 4 = 8$ possible production rules were derived. Examining the production rules we extracted, we found that only one of the 8 possible combinations existed in the production rules. This combination was $NP \leftarrow Nab + Nv4$. The result of cell (A, 2) was NP. The probability was 1 because $Nab + Nv4$ could only derive NP. The contents of cell (B, 3) could also be derived from the contents of cells (B, 2) and (C, 3).

Finally, we determined the syntactic categories of cell (A, 3) in the same way as in the preceding step. The syntactic categories of cell (A, 3) could be derived from cells (A, 1) and (B, 3), or cells (A, 2) and (C, 3) or cells (A, 1) and (B, 2) and (C, 3). The result was NP, which was derived from cell (A,1) and (B,3) by using the rule $NP \leftarrow Nbc + Nab$. The syntactic category of the CFS “林小姐” was NP, which was the only syntactic category derived by inspecting the contents of cell (A, 3).

4.3 Experimental Results

Our goal was to determine the syntactic categories of CFSs. The testing data we chose were in the bottom layer of each structural tree. Each level of the testing data contained many words. For example, we determined the syntactic categories of “要不要” and “這幅畫” as described for the example shown in Figure 4. We found that the syntactic category of “要不要” was VE2, and that syntactic category of “這幅畫” was NP. We retrieved 1,309 patterns and their related syntactic categories from the testing corpus. Among the 1,309 patterns, 98 patterns were our CFSs.

The structure of the notations of the syntactic categories defined by CKIP is a hierarchical one. There are a total of 178 syntactic categories with five layers in the hierarchical tree [CKIP 1993]. There are 8 categories in the first layer: N (noun), C (conjunction), V (verb), A (adjective), D (adverb), P (preposition), I (interjection), and T (auxiliary). The second layer contains 103 syntactic categories. For example, there are two sub-categories, Ca and Cb, in the second layer of category C in the first layer. Seven syntactic categories are defined in the Sinica Treebank. They are S (sentence), VP (verb phrase), NP (noun phrase), GP (direction phrase), PP (preposition phrase), XP (conjunction phrase), and DM (determinate phrase). We also put these 7 syntactic categories in the first layer of the hierarchical tree.

The achieved accuracy rates for determining the syntactic categories of these 98 CFSs by using all of the syntactic categories are shown in Table 7. When we used the syntactic categories in the first layer, the accuracy rate for the top one choice was 70.35%.

Table 7. The accuracy rate for 98 CFSs obtained by using all five layers of syntactic categories.

TOP n	Accuracy
TOP 1	63.26%
TOP 2	78.57%
TOP 3	91.67%
TOP 4	97.62%
TOP 5	97.62%

Because the size of training corpus was small compared with the hundreds of available syntactic categories, we also reduced the tags in each production tree to the second layer of the hierarchical tree. For example, when we reduced the syntactic categories of the production rule “S \leftarrow Cbca + NP + Dbb + VK2 + NP” to the second layer, we got the reduced production rule “S \leftarrow Cb + NP + Db + VK + NP “. We also determined the syntactic categories of the 98 patterns. The results are shown in Table 8. When we used the syntactic categories in the first layer, the accuracy rate for the top 1 choice was 76.28%.

Table 8. The accuracy rate for 98 CFSs obtained by using the syntactic categories reduced to the 2nd layer.

TOP n	Accuracy
TOP 1	71.02%
TOP 2	84.53%
TOP 3	92.86%
TOP 4	96.43%
TOP 5	98.81%

5. Conclusions

In this paper, we have presented some important properties of Chinese frequent strings. We used CFSs in several applications. We found that the CFS-based uni-gram LM was superior to traditional N-gram LMs when the training data was sparse. While the size of a corpus using the CFS-based uni-gram LM can be far smaller than that needed when using traditional N-gram LMs, for the applications studied here, the results obtained using the CFS-based uni-gram LM are better than those obtained using an n-gram LM.

Acknowledgements

We would like to thank Academia Sinica for providing its ASBC corpus, ASCED dictionary, and Sinica Treebank. We also extend our thanks to the many news companies for distributing their files on the Internet.

References

- C. H. Chang, “A Pilot Study on Automatic Chinese Spelling Error Correction,” *Communication of COLIPS*, Vol. 4, No. 2, 1994, pp. 143-149.

- K. J. Chen, C. R. Huang, L. P. Chang, and H. L. Hsu, "SINICA CORPUS: Design Methodology for Balanced Corpora," *Proceeding of PACLIC 11th Conference*, 1996, pp. 167-176.
- F. Y. Chen, P. F. Tsai, K. J. Chen, and C. R. Huang, "Sinica Treebank," *Computational Linguistics and Chinese Language Processing*, Vol. 4, No. 2, 1994, pp. 75-85.
- CKIP(Chinese Knowledge Information Processing Group, 詞庫小組) , "Analysis of Chinese Part-of-Speech (中文詞類分析), Technical Report of CKIP #93-05(中文詞知識庫小組技術報告 #93-05)," Academia Sinica, Taipei, Taiwan, 1993.
- T. H. Cormen, C. E. Leiserson, R. L. Rivest, "Introduction to Algorithms," The MIT Press, 1998.
- F. Jelinek, "Self-organized Language Modeling for Speech Recognition," *Readings in Speech Recognition*, Ed. A. Wabel and K. F. Lee. Morgan Kaufmann Publishers Inc., San Mateo, California, 1990, pp. 450-506.
- Y. C. Lin, "A Level Synchronous Approach to Ill-formed Sentence Parsing and Error Correction," Ph.D. Thesis, Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, June 1994.
- Y. J. Lin and M. S. Yu, "Extracting Chinese Frequent Strings Without a Dictionary From a Chinese Corpus And its Applications," *Journal of Information Science and Engineering*, Vol. 17, No. 5, 2001, pp. 805-824.
- National Taiwan Normal University, "Mandarin Phonetics," National Taiwan Normal University Press, Taipei, Taiwan, 1982.
- L. R. Rabiner and B. H. Juang, "Fundamentals of Speech Recognition," Prentice Hall Co. Ltd., 1993.
- B. Suhm and A. Waibel, "Toward Better Language Models for Spontaneous Speech," *Proc. ICSLP*, 1994, pp. 831-834.
- Jian Wu and Fang Zheng, "On Enhancing Katz-Smoothing Based Back-Off Language Model," *International Conference on Spoken Language Processing*, 2001, pp. I-198-201.
- K. C. Yang, "Further Studies for Practical Chinese Language Modeling," Master Thesis, Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, June 1998.