Text Segmentation for Chinese Spell Checking

Kin Hong Lee and Mau Kit Michael Ng
Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, Republic of China. E-mail: khlee@cse.cuhk.edu.hk, mkng@cse.cuhk.edu.hk

Qin Lu
Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Hong Kong. E-mail: csluqin@comp.polyu.edu.hk

Chinese spell checking is different from its counterparts for Western languages because Chinese words in texts are not separated by spaces. Chinese spell checking in this article refers to how to identify the misuse of characters in text composition. In other words, it is error correction at the word level rather than at the character level. Before Chinese sentences are spell checked, the text is segmented into semantic units. Error detection can then be carried out on the segmented text based on thesaurus and grammar rules. Segmentation is not a trivial process due to ambiguities in the Chinese language and errors in texts. Because it is not practical to define all Chinese words in a dictionary, words not predefined must also be dealt with. The number of word combinations increases exponentially with the length of the sentence. In this article, a Block-of-Combinations (BOC) segmentation method based on frequency of word usage is proposed to reduce the word combinations from exponential growth to linear growth. From experiments carried out on Hong Kong newspapers, BOC can correctly solve 10% more ambiguities than the Maximum Match segmentation method. To make the segmentation more suitable for spell checking, user interaction is also suggested.

Introduction

Spelling checkers for Western languages are very mature tools in word processing. However, the counterpart for Chinese word processing is very different. Chinese text has no natural delimiters such as spaces between words, which are meaningful sequences of characters. Every Chinese character input must be a valid ideograph, but the sequence of Chinese characters may not make sense. For example, if the word 時間 (which means time) is mistyped as 時閒, both characters 時 and 閒 are correct characters, although the character sequence 時閒 is not a correct word. Spell checking in Chinese text is designed to identify the wrong use or misuse of characters in text composition. In other words, it is error correction at the word (a meaningful sequence of characters) level rather than at the character level. Chinese spell checking is usually divided into two steps: segmentation of text and error detection. Segmentation is a process that divides a string of characters into words. The segmented text is then checked against a dictionary or thesaurus. Grammatical rules are also sometimes used to detect errors. In this article, the focus is on the segmentation process.

Segmentation is not a trivial process due to morphological complexities and ambiguities in the Chinese language (Wu & Tseng, 1995). A sentence may be segmented in several legitimate ways, yielding different meanings. It is not easy to determine which of the possible segmentations is the best. Chinese segmentation problems have been reported in many information retrieval systems. Also, there is a problem with unknown words. Unknown words are words that are not predefined in the system. It is not practical to define all Chinese words in a dictionary because new words can be created by combining characters or words (Nie, Hannan, & Jin, 1995). For example, 足球場 (Football Field) is a combination of 足球 (Football) and 場 (Place). Among the unknown words, there are morphologically derived words, personal names, and transliterated foreign names (Chang, Sproat, Shih, & Gale, 1994).

In Chinese spell checking, it cannot be assumed that texts are free of errors. This further complicates the process of segmentation. There are four main kinds of errors. They are (Chang, 1994):

1. Misuses of characters due to same or similar sounds; e.g., 按「步」就班 should be corrected as 按部就班 (an idiom that means following the prescribed order), where both the character 步 (step) and 部 (department) in Chinese phonetic system Pinyin are "bu4."
2. Misuses of characters due to similar shapes; e.g., 桿「茵」should be corrected as 桿菌 (*Bacillus*); e.g., 「荼」「壼」should be corrected as 茶壺 (teapot). Note that the character pairs (茵, 菌), (荼, 茶), (壼, 壺) are similar in shapes.

Received March 25, 1998; revised July 15, 1998; accepted October 22, 1998.

3. Misuses of characters due to similar meanings; e.g., 名「符」其實 should be corrected as 名副其實 (an idiom that means not just in name only, but also in reality). The character 符 means "in accordance with" and 名符其實 can be interpreted as "the name is in accordance with the reality." Some people argue that both are correct in meaning, but it is incorrect because the original characters must be preserved in an idiom; e.g., 既往不「究」should be corrected as 既往不咎 (an idiom that means let bygones be bygones). The character 究 means "to investigate," and 既往不究 can be interpreted as "not to investigate for somebody's past misdeeds," However, it is incorrect because the original characters must be preserved in an idiom.
4. Typing errors related to Chinese input methods.

However, unlike text analysis for translation or semantic analysis, sometimes it is not necessary for a spelling checker to find a unique segmentation solution.

For example, the sentence 發展中國家用電器換取外匯 may be segmented as:

1. 發展 中國 家用電器 換取 外匯
   develop / China / household-appliance / exchange / foreign currency
   The translation is; "to develop China's household-appliance industry to exchange for foreign currency."
2. 發展中國家 用 電器 換取 外匯
   developing country / use / appliance / exchange / foreign currency
   The translation is: "developing countries use appliances to exchange foreign currency." Both segmentation results are correct from the point of view of spell checking, and there is no need to solve the ambiguity.

Many segmentation methods exist (Bai, 1994; Chang, 1994; Chang & Chen, 1993; Gao & Chen, 1996; Liang & Zheng, 1991; Nie, Hannan, & Jin, 1995; Sproat & Shih, 1990; Wu & Tseng, 1995). Most segmentation methods find a unique solution without interacting with the user, even if suspected errors occur. Some of these methods assume that the texts to be segmented are correct, while others automatically choose the most likely segmentation as the solution. They are suitable for applications such as semantic analysis for Information Retrieval. However, they are not suitable for spell checking in text processing because errors and unknown words are not dealt with. Because there is no accurate way to distinguish errors from unknown words, the best solutions obtained may not match the original writer's intended meaning, as in the example shown above.

In this article, a Block-of-Combinations (BOC) segmentation method based on frequency of word usage is proposed. To make the method more suitable for spell checking, user interaction is also introduced into the system. Interaction is possible because the spell checker is intended for on-line text checking. When suspected errors occur, the system will allow the user to make the final decision. Based on the user's response, the segmentation can be refined to fit the user's interpretation, and unknown words can also be learned by the system during the spell checking process.

The rest of this article is organized as follows. Section 2 discusses related work in segmentation. Section 3 gives an overview of the segmentation process and the interactive model of this system. A word-frequency based segmentation method (BOC) is described in Section 4. Section 5 is the conclusion.

Related Work

There are many different approaches to Chinese text segmentation. Basically, they can be classified into rule-based, statistics-based, or hybrid methods based on the combination of rules and statistics.

In rule-based segmentation approaches, dictionaries are often used. However, the arrangement of the dictionaries varies from one design to another. Some segmentation methods use dictionaries of words. The sequences of characters to be segmented are checked against the dictionaries. Among those methods, maximum match (Liang & Zheng, 1991) is the most commonly used because it is simple and efficient. The idea of maximum match is to select the longest word among all possibilities when there is an ambiguity. On the other hand, some methods use dictionaries of word components. It is based on the idea that most of the words exceeding two characters can be formed by one-character or two-character words. Grammatical rules can be incorporated to combine the word components (Wu & Tseng, 1995).

Besides dictionaries of words, other information is also considered in some designs. In the segmentation method proposed by Chang (1994), a character table for similar shape, sound, meaning, and input-method-code characters are proposed. In Chang's method, all the combinations are proposed based on the character table and scores are given to the combinations. This approach tries to "guess" all the possible errors. The problem with this method is its limitations in handling errors other than single substitution errors. Also, the performance is highly dependent on the size of the character table. A knowledge base containing grammatical and semantic knowledge for word segmentation is suggested by Liang and Zheng (1991). However, it is not easy to construct a complete knowledge base for all Chinese words. The semiword method was introduced by Bai (1994). A semiword is a one-character word that is seldom used as a word. Instead, semiwords are used to form words with other characters. Examples of semiwords are 確 (real), 實 (real), 理 (reason). They are seldom used as words on their own, but they form words such as 確實 (really) and 真理 (the truth), which are used more often. A set of semiwords was compiled by Bai (1994). The best segmentation is chosen using a set of scoring principles.

In statistics-based methods, probabilities such as word frequency and character co-occurrence frequency are considered. Lua (1990) proposed to use information theory in word formation. New words are formed if there is a significant change in entropy, in terms of word frequency. A method based on mutual information was proposed by Sproat and Shih (1990). This method gives a measure of how strongly two characters are associated based on the probability of the occurrence of the characters. Also, bigrams are considered in finding word boundaries. An *n*-gram is a string of *n* adjacent characters that may or may not be words (while in some other approaches (Chang & Chen, 1994) an *n*-gram is defined as a string of *n* adjacent words). In particular, a bigram is a string of two adjacent characters, and is often used in statistical and hybrid approaches. Bigrams and trigrams were used in Chang and Chen (1994) for classifying Chinese words. Markov models and bigrams are used together as an evaluation of a segmentation solution in Yeh and Lee's method (1991). Gao and Chen (1996) considered all the combinations of *n*-grams, in which *n* is a variable, in performing segmentation.

Some of the articles suggested that it may be better to solve the problem of segmentation together with other goals. The integration of word segmentation and part-of-speech tagging was proposed by Chang and Chen (1993). From their results, it was found that the segmentation-dominated approach is better than the tagging-dominated approach. Thus, a good segmentation can improve tagging. Nie, Hannan, and Jin (1995) suggested that unknown word detection can be integrated with segmentation. The phrase to be handled is first segmented as much as possible based on a dictionary. The unsegmented portions are then examined and candidate unknown words are proposed. Based on statistics, unknown words are found.

Quite a number of these efforts at segmentation achieve a high level of accuracy in tests. However, most of the methods assume that there are no unknown words, and automatically choose a segmentation among the possibilities as the solution without interacting with the user, even if there are suspected errors. This approach should not be applied to spell checking because the segmentation chosen may not match the original writer's intentions.

The Segmentation Process and System Interaction Model

The segmentation process BOC proposed is dictionary based. It makes use of the statistical data about Chinese words published by the Education Department of Hong Kong (香港教育署, 1986). The dictionary in BOC contains 60,000 words, in which the 2,000 most frequently used words are grammatically tagged. BOC also includes a user dictionary, and a temporary dictionary for the storage of unknown words. The user dictionary stores user-defined words, which are not predefined in the system and can be updated from time to time, whereas the temporary dictionary automatically stores the unknown words until the whole segmentation process terminates.

When a piece of text is to be handled, it is first divided into sentences. Punctuation marks are used as delimiters to separate sentences. Some of the sentences may contain symbols, alphabetic symbols, and numerals. These types of characters are skipped without checking, and are used as unnatural delimiters to further divide sentences into phrases. The phrases are then segmented into words by the BOC method presented in the next section. Because incorrect characters such as mistyped, characters often cannot form a multicharacter word with adjacent characters (Leung & Kan, 1996), single-character words are considered as suspected errors. Table 1 of Lua (1990) shows, however, monosyllabic words have the highest frequencies, although the number of different monosyllabic words (i.e., single-character words) is much less than that of disyllabic words (i.e., words consisting of two characters).

TABLE 1. Distribution of monosyllabic and polysyllabic words (Lua, 1990, p. 306).

Word length	Number	Usage
Monosyllabic	12.1%	64.3%
Disyllabic	73.6%	34.3%
Trisyllabic	7.6%	0.4%
4-Syllabic	6.4%	0.4%
5-Syllabic or more	0.2%	0%

In fact, a number of single-character words are used quite frequently. If all the single-character words are treated as suspected errors and presented to the user, there will be a lot of false alarms. To reduce false alarms, occurrences of the first 200 most frequently used single-character words such as 的 (of), 一 (one), 是 (is), 不 (not), 有 (have), 在 (in), 我(I), 個 (unit, quantity) should not be considered as suspected errors.

When a suspected error is detected, it is presented to the user for clarification. A continuous string of suspected errors is considered as one suspicious unit, and is highlighted for the user. Words that are similar to a suspicious unit are fetched from the dictionaries and displayed as suggested corrections. The process that determines the suggested corrections is beyond the scope of this article. Because a suspicious unit may not correspond to a word, the user can adjust the boundaries of the suspicious unit by highlighting it again. The corresponding list of suggested corrections is then obtained.

A suspected error may be an error or an unknown word. Errors can be corrected at once. The user can replace a suspicious unit by a suggested correction, or edit the erroneous sentence directly. On the other hand, unknown words are automatically stored in the temporary dictionary, or they can also be stored in the user dictionary for future use based on the user's indication. When those words are encountered again in the text, they will not be treated as suspected errors because they are not unknown words any more. After a suspected error is handled, the segmentation process continues at the position immediately after the word handled.

Block-of-Combinations (BOC) Segmentation Method

Before describing the BOC method, let's consider the following phrase first: 誰都不知道它的確實用途 (no one knows its real use).

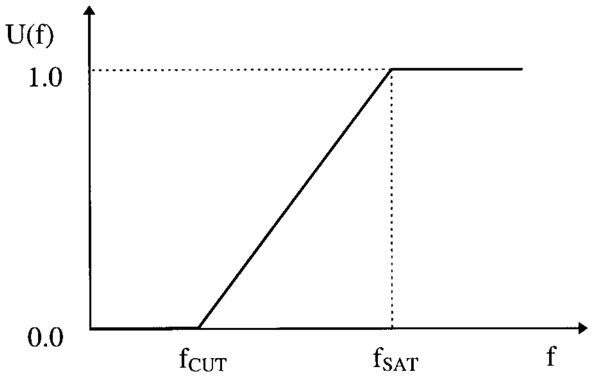

FIG. 1. The relation between single-character word function $U(f)$ and word frequency f.

Assume that the words 知道 (know), 確實 (really), 確實 (really), 實用 (practical), 用途 (use) are in the dictionary. Then there are several possible combinations. For example,

1. 誰 都 不 知道 它 的確 實用 途
2. 誰 都 不 知道 它 的確 實 用途
3. 誰 都 不 知道 它 的 確實 用途

The correct segmentation should be combination 3:

誰 都 不 知道 它 的 確實 用途

(no one knows its real use).

The word by word translation is: who / also / not / know / it / of / real / use. The other two segmentations are not meaningful.

This may be deduced from so-called word formation power (香港教育署, i.e., the Education Department of Hong Kong, 1986), in which the word formation power of the character 實 is higher than that of the character 的, and so it is more likely that the character sequence 確實 is a word. Also, the co-occurrence probability suggested by Sproat and Shih (1990) may be used to choose the correct segmentation. However, the correct segmentation would be totally different if one of the characters is changed. For example: 誰都不知道它的確實用嗎 (the last character is changed from 途 to 嗎) (Does no one know that it is really practical?)

The correct segmentation of this phrase should be:

誰 都 不 知道 它 的確 實用 嗎

(the word by word translation is: who / also / not / know / it / really / practical / question tag), which is very different from the previous example.

In fact, the segmentation can be viewed in another way. The first phrase is segmented as above because it is unlikely that either the character 實 or the character 途 is a single-character word. Therefore, the segmentation can be considered as choosing the combination that has the smallest number of "unusual" single-character words. This is similar to the semiword method proposed by Bai (1994). Recall that a semiword is a one-character word that is seldom used as a word. However, in the semiword method, a character is either in the set of semiwords or not, and takes a binary value. For the BOC method proposed, word frequency is considered rather than binary value.

In the BOC segmentation method proposed, single-character-word function U is defined as follows:

$$U(f) = \begin{cases} 1 & \text{if } f \geq f_{\text{SAT}} \\ (f - f_{\text{CUT}})/(f_{\text{SAT}} - f_{\text{CUT}}) & \text{if } f_{\text{CUT}} < f < f_{\text{SAT}} \\ 0 & \text{if } f \leq f_{\text{CUT}} \end{cases}$$

where f is the occurrence frequency of the character as a single-character word; f_{CUT} is the threshold frequency below the range in which the characters are considered as semiwords; f_{SAT} is the threshold frequency above which the characters often appear a single-character words.

$U(f)$ can be visualized by Figure 1.

The score of a segmentation is defined as: Score-S $= \Sigma$ $(1 - U(f_j))$, where j is a single character appearing in the segmentation.

Thus, the best segmentation is the one with the smallest Score-S.

Heuristic for Finding the Best Segmentation

To find the segmentation with the smallest Score-S, all the possible combinations of the words are considered. Theoretically, any sequence of Chinese characters can form a word, if unknown words are also considered. For a phrase of L characters, if the maximum word length is restricted to m, there are m possible words starting with the first character. For example, in the phrase 天鵝湖是 . . . (swan lake is . . .), when $m = 3$, at most 3 words can be formed with the first character 天: 天鵝湖 (swan lake), 天鵝 (swan), and 天 (sky) (Fig. 2).

If it is provided that the first character 天 is segmented as a monosyllabic word, the number of combinations of the phrase is reduced to that of the remaining $L - 1$ characters 鵝湖是 . . . Similarly, for the case that the first two characters 天鵝 are segmented as on disyllabic word, the number of combinations is dependent on the $L - 2$ characters 湖是 . . . . If the first m characters are segmented as one word, the number of combinations of the phrase will be reduced to that of the L-m characters. Hence, (1) for a phrase of length 1 (i.e., a character), the maximum number of different segmentation is 1. (2) From Figure 2, for a phrase of L characters, the maximum number of segmentations is:

$$\begin{aligned} & \text{max. no. of segmentation of } L - 1 \text{ characters} \\ + \; & \text{max. no. of segmentation of } L - 2 \text{ characters} \\ + \; & \cdots \\ + \; & \text{max. no. of segmentation of } L - m \text{ characters} \end{aligned}$$

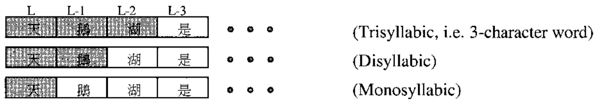

FIG. 2. The words formed with the first character in a phrase.

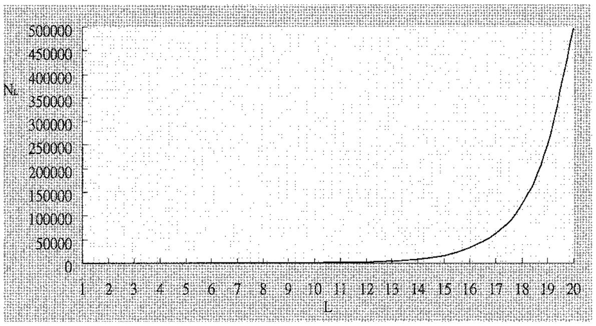

FIG. 3. Exponential growth of the number of segmentations.

Thus, the maximum number of segmentations for a phrase of L characters is:

$$N_L = \sum_{i=1}^{i=m} N_{L-i}$$

where $N_{L-i} = 0$ if $L - i < 0$; and $N_{L-i} = 1$ if $L = i$.

For the case of maximum word length $m = 7$, the maximum number of segmentations N_L for phrase length L are computed and plotted in Figure 3.

From Figure 3 it can be observed that the maximum number of segmentations increases exponentially with the phrase length, and there is a risk of combinatory explosion. Therefore, segmentation will cause long delays if L is large.

To avoid the risk of combinatory explosion, a heuristic is designed. It is observed that although there are long-distance dependency phenomena in Chinese, most of the ambiguities can be solved by considering a few adjacent characters. Also, from Table 1, most of the words in Chinese are monosyllabic and disyllabic words. Thus, the probability of ambiguities involving long words is much lower than those involving disyllabic words or trisyllabic words. The exception is when long words are composed of shorter ones, e.g., 交通工具 (vehicle) is composed by 交通 (traffic) and 工具 (tool). The heuristic proposed is based on these assumptions.

Instead of considering all the combinations of a long phrase at one time, the segmentation process considers text under a sliding window. In each iteration, the process looks ahead several characters and generates combinations to choose the best solution. Because there may be several ambiguities adjacent to each other, it may not be able to find a common ending position for evaluation without considering a long series of characters. So the concept of Terminators is designed. A Terminator is the starting position of the words that follow the words considered in the current iteration. Informally speaking, they are words that will be considered in the next iteration. For example, when considering the phrase 筆畫就是構成漢字字形的各種點和線 (strokes are the dots and lines that construct the shape of the Chinese characters), if the current iteration initially considers the first five characters, then the Terminators are the starting positions of the words located behind the first five characters. Because 漢字 (Chinese character) and 字形 (character shape) are words, the phrase becomes:

筆畫就是構成漢字字形的各種點和線,

where the arrows indicate two of the possible Terminators.

To perform evaluation of a word combination, a Score-S is calculated from the starting position of current iteration to the nearest Terminator behind the combination. Thus, if a combination terminates immediately before a Terminator, then the combination matches well with a potential solution in the next iteration. Note that even in the same iteration, the number of characters considered in calculating the Scores for different combinations may be different, but are restricted to a certain range. The detail of the heuristic is described as follows:

1. The segmentation process scans the phrase from left to right.
2. A number Max_W is predefined (e.g., $\text{Max}_W = 5$), which is the maximum length of character strings that will be considered in solving ambiguities. Whenever a word longer than Max_W is encountered, the word is chosen as the result. Thus, in the following steps, the lengths of all the words are assumed to be Max_W or less.
3. At a certain position P of the phrase, all the words beginning with character P are found. (For simplicity, here "words" means "multicharacter words.") (a) If there is no word beginning at P, it is segmented as a single character and the next iteration starts. (b) Otherwise, for the longest word W starting at P, its length L_W is found. All the words starting within P and $P + (L_W - 1)$ inclusively are found. They are denoted as $\{W_{\leq Lw}\}$. If there is only one word, W will be accepted as the result. Then the next iteration begins. If there is more than one word, it is considered to be ambiguous, and further analysis is carried out in the following steps.
4. All the words starting within P and $P + (\text{Max}_W - 1)$ inclusively are found. This set is denoted by $\{W_{\leq max}\}$. Because all the words under consideration are within Max_W characters, the maximum extent they can span is from P to $P + 2(\text{Max}_W - 1)$. For example, if $\text{Max}_W = 5$, the maximum extent the words can span is from P to $P + 8$. This is illustrated in Figure 4.
5. All the possible combinations of the words in $\{W_{\leq \max}\}$ are generated so that each of these combinations starts at P and ends between $P + (\text{Max}_W - 1)$ and $P + 2(\text{Max}_W - 1)$. In particular, if $\text{Max}_W = 5$, the number of combinations is upper bounded by 65, which will be explained in the next section.
6. All the words starting within $P + \text{Max}_W$ and $P + (\text{Max}_W - 1) + \text{Max}_W$ inclusively are found and used as Terminators. For $\text{Max}_W = 5$, the Terminators are within $P + 5$ and $P + 9$.

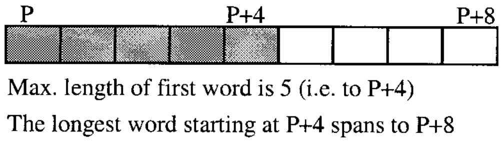
P
P+4
P+8
Max. length of first word is 5 (i.e. to P+4)
The longest word starting at P+4 spans to P+8

FIG. 4.

7. For each of the combinations generated, the corresponding Score-S is evaluated from P to the smallest Terminator after the combination.
8. After considering all the combinations generated, the one with the smallest Score-S is chosen as the best solution, and the word combination from P to the first word starting between P and $P + (\text{Max}_W - 1)$ is the segmentation result of the current iteration. Ties in the smallest Score-S are broken by first choosing the longest word combination, and then combination consists of the smallest number of words.

Maximum Number of Combinations in Each Iteration

Recall from the previous section that the length of all the words are Max_W or less, and P is the first character considered in the current iteration. For a phrase of L characters, if $L \leq \text{Max}_W$, then the number of possible word combinations C_L is 2^{L-1}. For example, the phrase ABC can be segmented as {A/B/C, A/BC, AB/C, ABC}. That is, $L = 3$ and $C_L = 2^{3-1} = 4$.

According to point 4 of BOC in the previous section, the number of possible combinations in an iteration is maximum when the words can span from P to $P + 2(\text{Max}_W - 1)$.

That is, $L \geq P + 2(\text{Max}_W - 1) - P + 1; \geq 2\text{Max}_W - 1$.

For a phrase of L Characters where $L \geq 2\text{Max}_W - 1$, because all the combinations start at P and end between $P + (\text{Max}_W - 1)$ and $P + 2(\text{Max}_W - 1)$ as mentioned in point 5 of BOC, the maximum number of combinations:

$$\text{Max} - C = \sum_{i=\text{Max}_W}^{2\text{Max}_W - 1} \text{number of combinations of length } i$$

Recall that all combinations are formed by words in $\{W_{\geq \max}\}$.

1. For any combination longer than Max_W, the last word in the combination must start with a character within $P + 1$ and $P + \text{Max}_W - 1$.
   If the last word starts at position $P + 1$, the number of combinations $= C_1 \times 1$, as in Figure 5.
   If the last word starts at position $P + 2$, the number of combinations $= C_2 \times 2$, as in Figure 6.
   Similarly, if the last word starts at position $P + (\text{Max}_W - 1)$, the number of combinations is $C_{\text{Max }w-1} \times (\text{Max}_W - 1)$.
   Thus, the total number of combinations longer than

$$\text{Max}_W = \sum^{\text{Max}_W - 1} j \times C_j$$

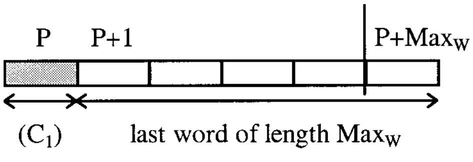

FIG. 5.

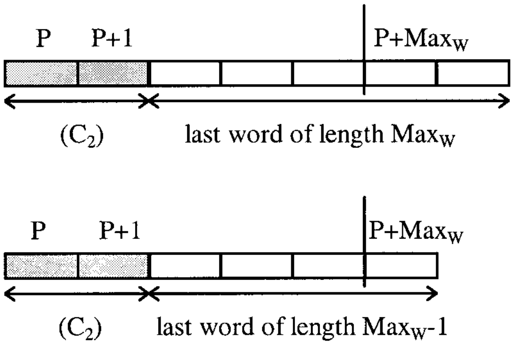

FIG. 6.

2. The number of combinations of length $\text{Max}_W = C_{\text{Max }w}$. Therefore, the maximum number of combinations

$$\text{Max} - C = C_{\text{Max }w} + \sum_{j=1}^{\text{Max}_W - 1} j \times C_j$$

   For example, if $\text{Max}_W = 5$,

$$\text{Max} - C = 2^{5-1} + \sum_{j=1}^{5-1} j \times 2^{j-1} = 65$$

The maximum length of words involving ambiguities (i.e., Max_W) determines the maximum number of combinations Max-C. The larger the maximum length, the more combinations have to be considered, and hence, the more computation time is needed. Therefore, the value of Max_W should be kept as small as possible. On the other hand, because most of the words in Chinese are monosyllabic and disyllabic, most of the ambiguities involve disyllabic words. An ambiguity of two disyllabic words consists of three characters. As the algorithm uses adjacent multicharacter words for solving ambiguities, at least two more characters have to be considered. Therefore, the preferable value of Max_W should be at least 5. It is also a fact that there are not many Chinese words consisting of six of more characters. If such words are encountered, it is very likely that they are desired even when ambiguities occur because long words are often composed of shorter ones. Thus, the maximum length of words involving ambiguities (i.e., Max_W) is set to 5.

For each iteration, if $\text{Max}_W = 5$, the maximum number of combinations is 65 when there are ambiguities. For the worst case, the result of each iteration is a disyllabic word (i.e., word consisting of two characters). Therefore, for a long phrase of 30 characters, 15 iterations are needed and the number of combinations is 65 × 15, i.e., 975. The exponential growth of combinations when considering all the possibilities at one time is reduced to linear by this BOC method.

Example

Assume that the phrase 發展中國家庭電器換取外匯 (to develop China’s household-appliance industry to exchange for foreign currency) is to be segmented. The corresponding words in the dictionary are 發展中國家 (developing country), 發展 (develop), 中國 (China), 國家 (country), 家庭電器 (household-appliance), 家庭 (family), 電器 (appliance), 換取 (exchange), and 外匯 (foreign currency). Also, $\text{Max}_W = 5$ throughout the iterations (refer to step 2). That is, if a word of six or more characters is encountered, the word will be chosen as the result of that iteration.

Iteration 1

The first character 發 is considered ($P = 1$). There are two words starting with it: 發展中 國家 and 發展. The longest word $W =$ 發展中國家 and its length $L_W = 5$ (refer to step 3b). All the words starting within P (i.e., 1) and $P + (L_W - 1)$ (i.e., 5) inclusively are put in $\{W_{\le Lw}\}$. $\{W_{Lw}\}$ = {發展中國家, 發展, 中國, 國家, 家庭電器, 家庭}.

There are six words in $\{W_{\le Lw}\}$, so it is considered as ambiguous.

Then, because $P = 1$ and $P + (\text{Max}_W - 1) = 5$, all the words starting between the first character 發 and the fifth character 家 are found (refer to step 4). There are six words denoted by:
$\{W_{\text{max}}\}$ = {發展中國家, 發展, 中國, 國家, 家庭電器, 家庭}.

Combinations of words are generated based on $\{W_{\le\text{max}}\}$ (refer to step 5). Examples are:

1. 發展中國家
2. 發展　中　國家
3. 發展　中國　家庭電器
4. 發展　中國　家庭

Terminators are then found. Because $P + \text{Max}_W = 6$, and $P + (\text{Max}_W - 1) + \text{Max}_W = 10$, the Terminators start between the sixth character and the 10th character in the iteration (refer to step 6). Therefore, the Terminators are within the underlined portion of the sentence: 發展中國家庭電器換取外匯. Thus, the Terminators are at the characters 電 and 換,, because 電器 and 換取 are words in the dictionary.

To evaluate the Score-S, the Terminators are used to limit the length of the combinations (refer to step 7). The corresponding combinations to be evaluated are:

1. 發展中國家　庭 ↑ Terminator
$U(f)$: 0.0025 Score-S = 0.9975
2. 發展　中　國家　庭 ↑ Terminator
$U(f)$: 0.0915 0.0025 Score-S = 1.9060
3. 發展　中國　家庭電器 ↑ Terminator Score-S = 0.0
4. 發展　中國　家庭 ↑ Terminator Score-S = 0.0

Because the single character 庭 appears in combinations 1 and 2, their Score-S will be higher. Recall that the best combination is the one with the lowest Score-S, combinations 3 and 4 are better. Because combination 3 is longer, it is considered as the best combination (refer to step 8). The segmentation result of this iteration is the first word 發展. The next iteration starts immediately after this word, and is at the character 中.

Iteration 2

The character 中 is considered and, thus, $P = 3$. The only word starting with 中 is 中國. Therefore, $W =$ 中國 and $L_W = 2$ (refer to step 3b).

Then, because $P = 3$ and $P + (L_W - 1) = 4$, all the words starting between the third character 中 and the fourth character 國 are found. There are two words, denoted by: $\{W_{\le Lw}\}$ = { 中國, 國家 }.

Thus, the sequence is ambiguous.

Because $P = 3$ and $P + (\text{Max}_W - 1) = 7$, all the words starting between the third character 中 and the seventh character 電 are found (refer to step 4). There are five words denoted by:
$\{W_{\le\text{max}}\}$ = { 中國, 國家, 家庭電器, 家庭, 電器 }.

Combinations are generated based on $\{W_{\le\text{max}}\}$ (refer to step 5). For example,

1. 中國　家庭　電器
2. 中　國家　庭　電器
3. 中國　家庭電器

Because $P + \text{Max}_W = 8$ and $P + (\text{Max}_W - 1) + \text{Max}_W = 12$, the Terminators are words starting between the 8th character 器 and the 12th character 匯 in the iteration (refer to step 6). Thus, the Terminators are at the starting position of the words 換取 and 外匯.

Therefore, the evaluation results are (refer to step 7):

1. 中國　家庭　電器 ↑ Terminator Score-S = 0.0
2. 中　國家　庭　電器 ↑ Terminator
$U(f)$: 0.0915 0.0025 Score-S = 1.9060
3. 中國　家庭電器 ↑ Terminator Score-S = 0.0

After evaluating the Score-S of the combinations, it is found that combinations 1 and 3 are equally good. Because combination 3 consists of fewer words, it is the best combination (refer to step 8). The segmentation result of this iteration is the word 中國. Then in the next iteration, it will start with the character 家

Iteration 3

Character considered: 家 and $P = 5$ (step 3).

Words beginning at $P = 5$ are 家庭電器 and 家庭. Therefore, $W =$ 家庭電器 and $L_W = 4$ (step 3b). $\{W_{\le Lw}\}$ = { 家庭電器, 家庭, 電器 }. Thus, ambiguous. $\{W_{\le\text{max}}\}$ = { 家庭電器, 家庭, 電器, 換取 } (step 4).

Combinations based on $\{W_{\le\text{max}}\}$ are generated (step 5):

1. 家庭電器　換取
2. 家庭　電器　換取

Terminator is at the starting position of 外匯 (step 6).

After consideration, combination 1 is better and the result of this iteration is 家庭電器 (steps 7 and 8).

Iteration 4

Character considered: 換 and $P = 9$.

The only word starting at $P = 9$ is 換取. Thus, $W =$ 換取 and $L_W = 2$. $\{W_{\leq Lw}\} = \{$ 換取 $\}$. Therefore, it is not ambiguous (step 3b).

Result of this iteration is 換取.

Iteration 5

Character considered: 外 and $P = 11$.

The only word starting at $P = 11$ is 外匯. Thus, $W =$ 外匯 and $L_W = 2$. $\{W_{\leq Lw}\} = \{$外匯 $\}$. Therefore, it is not ambiguous (step 3b).

Result of this iteration is 外匯..

The final segmentation result of the phrase will be 發展 中國 家庭電器 換取 外匯 . The translation is: "to develop China's household-appliance industry to exchange for foreign currency."

With this method, the large number of combinations for a long phrase is broken down into blocks of combinations of shorter phrases. Thus combinatory explosion is avoided.

Evaluation Results

The BOC segmentation method proposed is evaluated in terms of accuracy and speed. A segmentation method called Forward Maximum Match (Liang & Zheng, 1991) is used as a control for comparison. Forward Maximum Match scans sentences from left to right. Character sequences consisting of the first character are first checked against a dictionary. If only one word is matched, it is considered as a segment. It is ambiguous if more than one word is matched. Ambiguities are resolved by choosing the longest word among all the possibilities. The process continues after the word matched. Forward Maximum Match is simple and efficient. However, it is not designed to handle unknown words and errors.

Accuracy

Tests have been performed with articles retrieved over the Internet from the newspaper Ming Pao in Hong Kong. Eleven pieces of articles, which are main news, with a total of 6,518 characters are segmented. The BOC segmentation method is compared with the Forward Maximum Match segmentation method. We found a total of 100 ambiguities. Among the 100 ambiguities, 68 of them can be solved correctly by both methods, and 5 of them cannot be solved correctly by both methods. Among the remaining 27 ambiguities, 19 of them can only be solved by BOC, while 8 of them can only be solved by Forward Maximum Match. Therefore, among the 100 ambiguities, BOC can solve 87 of them, while the Forward Maximum Match can only solve 76 of them. Thus, BOC can solve more ambiguities than Maximum Match by more than 10%. Some examples from the articles are as follows:

1. 可 爲 乘客 帶來 更 多 方便 (by Block-of-Combinations BOC, correct)
   (It can bring more convenience to the passengers)
2. 可 爲 乘客 帶來 更 多方 便 (by Forward Maximum Match FMM, incorrect)
   港 府 上述 提議 仍然 有效 (by BOC, reasonable)
   (The suggestion, which is proposed by the Hong Kong Government, mentioned above is still valid)
   港 府上 述 提議 仍然 有效 (by FMM, incorrect)
   where 港府 is an unknown word to the system which means "Hong Kong Government". Note how an unknown word affects the segmentation result.
3. 美 聯 物 業 主席 黃 建業 表示 (by BOC, reasonable)
   (Mr. Wong, the chairperson of Midland Realty, said)
   美 聯 物 業主 席 黃 建業 表示 (by FMM, incorrect)
   where 美聯物業 (Midland Realty) is a company name and 黃建業 (the full name of Mr. Wong) is a person name. Both of them are unknown words in the system.
4. However, some ambiguities cannot be solved correctly by BOC:
   以便 七月 一 日後 (by BOC, incorrect)
   以便 七月 一日 後 (by FMM, correct)
   (in order to . . . after the first of July).
   The incorrect resolution of ambiguities is because the character 一 often appears as a single-character word.
5. Further experiments are performed with errors randomly injected into the articles. Original sentence:
   該 批 貨 當時 分別 存放 在 十一 個 箱 內.
   (Those goods were separately placed in eleven boxes at that time)
   Error injected:
   該 批 貨 人 時 分別 存放 在 十一 個 箱 內 (by BOC, reasonable)
   該 批 貨 人 時分 別 存放 在 十一 個 箱 內 (by FMM, incorrect) where the character 人 is a substitution error randomly injected. Note how an error affects the segmentation result.

Speed

As a segmentation process of an on-line spell checker, speed is important. The computing time of the tests is recorded. In the test, the program is run on a Pentium 100 MHz personal computer with 32 M RAM. The operating system is UNIX. The BOC approach is compared with the Forward Maximum Match (FMM) method, which is very simple and efficient. Each set of data is tested twice. The results are shown in Table 2.

From the result, it is observed that the time performance of BOC and Maximum Match are very close to each other. Note that words in the dictionary are fetched by sequential search in the experiment. The speed can be significantly improved through indexing the dictionary or hashing.

Conclusion

In this article, a Block-of-Combinations (BOC) segmentation method based on single-character word occurrence

TABLE 2. Speeds of the FMM and BOC segmentation methods.

Case	Numbers of characters	FMM method time (seconds)			BOC method time (seconds)		
		Test 1	Test 2	Mean	Test 1	Test 2	Mean
1	610	135.49	135.21	135.350	135.29	135.35	135.320
2	806	165.99	166.04	166.015	166.21	166.19	166.200
3	293	68.51	68.77	68.640	68.66	68.49	68.575
4	250	57.53	57.62	57.575	57.47	57.51	57.490
5	740	167.85	168.25	168.050	168.05	167.91	167.980
6	1299	304.59	304.99	304.790	304.71	304.80	304.755
7	505	104.00	103.76	103.880	103.89	104.00	103.945
8	486	108.23	108.61	108.420	108.19	108.62	108.405
9	569	120.77	120.89	120.830	120.62	120.71	120.665
10	593	131.12	131.44	131.280	131.55	131.30	131.425
11	367	75.76	75.78	75.770	75.77	75.88	75.825
Total	6518			1440.600			1440.585

frequency is proposed. To find the best solution, a long phrase is broken into shorter ones, and a small number of word combinations are considered in each iteration so as to avoid the risk of combinatory explosion. The result of tests on newspapers retrieved over the internet shows that BOC is more accurate than the Forward Maximum Match approach. The computing time of BOC and Maximum Match are found to be very close to each other. With BOC, unknown words and errors can be taken into consideration during segmentation. As it is needed to deal with errors and unknown words in Chinese Spell Checking, it is also proposed to introduce user interaction into the system.

References

Bai, S. (1994). “Semi-word” method for Chinese word segmentation. International Conference on Chinese Computing ’94 (ICCC94), 304–309.

Chang, C.H. (1994). A pilot study on automatic Chinese spelling error correction. Communication of COLIPS, 4(2), 143–149.

Chang, C.H., & Chen, C.D. (1993). A study on integrating Chinese word segmentation and part-of speech tagging. Communications of COLIPS, 3(2), 69–77.

Chang, C.H., & Chen, C.D. (1994). A study on corpus-based classification of Chinese words. International Conference on Chinese Computing ’94 (ICCC94), 310–316.

Chang, N., Sproat, R., Shih, C., & Gale, W. (1994). A stochastic finite-state word-segmentation algorithm for Chinese. Proceedings of ACL 94.

Gao, J., & Chen, X. (1996). Automatic word segmentation of Chinese texts based on variable distance method. Communication of COLIPS, 6(2), 87–94.

Leung, C.H., & Kan, W.K. (1996). Difficulties in Chinese typing error detection and ways to the solution. Computer Processing of Oriental Languages, 10(1), 97–113.

Liang, N., & Zheng, Y. (1991). A Chinese word segmentation model and a Chinese word segmentation system PC-CWSS. Communications of COLIPS, 1(1), 51–55.

Lua, K.T. (1990). From character to word—An application of information theory. Computer Processing of Chinese & Oriental Languages, 4(4), 304–313.

Nie, J.Y., Hannan, M.L., & Jin, W. (1995). Unknown word detection and segmentation of Chinese using statistical and heuristic knowledge. Communications of COLIPS, 5(1&2), 47–57.

Sproat, R., & Shih, C. (1990). A statistical method for finding word boundaries in Chinese text. Computer Processing of Chinese & Oriental Languages, 4(4), 336–351.

Wu, Z., & Tseng, G. (1993). Chinese text segmentation for text retrieval: Achievements and problems. Journal of the American Society for Information Science, 44(9), 532–542.

Wu, Z., & Tseng, G. (1995). ACTS: An automatic Chinese text segmentation system for full text retrieval. Journal of the American Society for Information Science, 46(2), 83–96.

Yeh, C.L., & Lee, H.J. (1991). Rule-based word identification for Mandarin Chinese sentence—A unification approach. Computer Processing of Chinese & Oriental Languages, 5(2), 97–118.

香港教育署. (1986). 教育研究處, *香港初中學生 中文詞匯研究*. 香港教育署.