Computational Linguistics and Chinese Language Processing
vol. 2, no. 2, August 1997. pp. 25-46.

Longest Tokenization

JIN GUO*

Abstract

Sentence tokenization is the process of mapping sentences from character strings into strings of tokens. This paper sets out to study *longest tokenization* which is a rich family of tokenization strategies following the general *principle of maximum tokenization*. The objectives are to enhance the knowledge and understanding of the principle of maximum tokenization in general, and to establish the notion of longest tokenization in particular. The main results are as follows: (1) Longest tokenization, which takes a *token n-gram* as a tokenization object and seeks to maximize the object *length* in characters, is a natural generalization of the *Chen and Liu Heuristic* on the *table of maximum tokenizations*. (2) Longest tokenization is a rich family of distinct and unique tokenization strategies with many widely used maximum tokenization strategies, such as *forward maximum tokenization, backward maximum tokenization, forward-backward maximum tokenization*, and *shortest tokenization*, as its members. (3) Longest tokenization is theoretically a true subclass of *critical tokenization*, as the essence of maximum tokenization is fully captured by the latter. (4) Longest tokenization is practically the same as *shortest tokenization*, as the essence of length-oriented maximum tokenization is captured by the latter. Results are obtained using both mathematical examination and corpus investigation.

Keywords: sentence tokenization, tokenization disambiguation, maximum tokenization, critical tokenization, word segmentation, word identification.

1. Introduction

Sentence tokenization[1] is the task of converting a sentence from a character string into a string of word-like tokens. It is widely agreed to still be an open problem in Chinese Language Processing (Chen, 1996; Gan, Palmer and Lua, 1996; Huang, Chen and Chang,

*Institute of Systems Science, National University of Singapore, Kent Ridge, Singapore 119597. E-mail: guojin@iss.nus.sg.

1. Also known in the literature as *word segmentation* or *word identification*.

1996; Huang and Xia, 1996; Sproat, Shih, Gale and Chang, 1996; Su, Chiang and Chang, 1996; Sun and Huang, 1996) and is getting recognition as a general problem in Computational Linguistics (Guo, 1997; Wu, 1997). It has been stated in the literature that major obstacles lie in tokenization ambiguities and unknown words in real text, and in the lack of a consensus on standard for tokenization which includes an operational definition of the notion of words and a complete and consistent set of tokenization rules (Liu, Tan and Shen, 1994; Huang, Chen and Chang, 1996).

As to tokenization ambiguity resolution, what is interesting is the fact that, at least for Chinese, a quite intuitive heuristic, referred to here as *the principle of maximum tokenization*[2], alone can deliver closed-dictionary tokenization accuracy anywhere above 98% (Liang, 1986; Liu, 1986). It is believed that the principle of maximum tokenization is "the most powerful and commonly used disambiguation rule" (Chen and Liu, 1992, 104). However, "there are a few variations of the sense" (Chen and Liu, 1992, 104) of the principle, and different realizations "were invented one after another and seemed inexhaustible" (Webster and Kit, 1992, 1108). This implies that the principle is still not well understood, and that much better realizations might be waiting to be discovered.

Therefore, we set for ourselves the task of enhancing our knowledge and understanding of this vague but powerful tokenization principle. The starting point for this study was the following **Chen and Liu Heuristic** (Chen and Liu, 1992, 104):

(1) "The most plausible segmentation is the three word sequence with the maximal length."

According to Chen and Liu (1992, 104), this is adopted after having "done the experiments with each of different variations" of the principle, and "achieves as high as 99.69% accuracy". However, we have never seen evidence in the literature that it has been studied except for the original proposal; thus, we believe it is worth being investigated further.

What we will establish in this paper is a set of tokenization strategies, collectively referred to as *longest tokenization*. Both mathematical examination and corpus investigation will be conducted to explore its theoretical implications and practical behaviors.

In particular, we will demonstrate that (1) Longest tokenization, which takes a *token n-gram* as a tokenization object and seeks to maximize the object *length* in characters, is a natural generalization of the *Chen and Liu Heuristic* (Chen and Liu, 1992) on the *table of maximum tokenizations*. (2) Longest tokenization is a rich family of distinct

2. Also known in the literature as *maximal matching*.

and unique tokenization strategies with many widely used maximum tokenization strategies, such as *forward maximum tokenization, backward maximum tokenization, forward-backward maximum tokenization* (Liu, 1986; Liang, 1986), and *shortest tokenization* (Wang, 1989), as its members. (3) Longest tokenization is theoretically a true subclass of *critical tokenization* (Guo, 1997). (4) Except for ordinary cases where longest tokenization is forward and/or backward tokenization by definition, longest tokenization is practically the same as shortest tokenization.

While the proper solution for sentence tokenization need not necessarily be in the form of shortest tokenization or critical tokenization, this study is nevertheless informative in understanding the principle of maximum tokenization, and is instructive with regard to sentence tokenization practice. It is believed proper to claim that the two most *significant implications* of this study are that: (1) the *essence* of the principle of maximum tokenization has been fully captured by critical tokenization; and (2) the *essence* of length-oriented realizations of the principle has been fully captured by the token-based forward and/or backward maximum tokenizations at one extreme, and by the sentence-based shortest tokenization at the other.

The rest of this paper is organized as follows. We will first review in Section 2 various maximum tokenization strategies proposed in the literature, and develop in Section 3 the notion of longest tokenization. We will then analyze in Section 4 theoretical relationships among members of the longest tokenization family and those representative maximum tokenization strategies, and investigate in Section 5 practical relationships through detailed data examination on a large representative corpus. Major results achieved will be summarized in Section 6. There is also an appendix showing some indications on the proof of several theorems in Section 4.

2. Maximum Tokenization

Numerous sentence tokenization strategies following the general *principle of maximum tokenization* have been proposed in the literature. Among them, *forward maximum tokenization, backward maximum tokenization, forward-backward maximum tokenization* (Liang, 1986; Liu, 1986), and *shortest tokenization* (Wang, 1989) have generally been regarded as the most representative ones. It has also been claimed that *critical tokenization* is "the only type of tokenization completely fulfilling the principle of maximum tokenization." (Guo, 1997, 590)

In this section, we will review each of these tokenization strategies by presenting a brief description followed by two simple examples adapted from Guo (1997). Before

that, however, let us examine *exhaustive tokenization.*

Exhaustive tokenization (**ET**), as the name implies, aims to produce all possible tokenizations. More precisely, a word string is an *ET* tokenization of a character string, if the concatenation of the former reproduces the latter. *ET* is not a type of maximum tokenization strategy, but is the base for all types of tokenization strategies.

EXAMPLE 1: Given the mini-English dictionary *D={a, d, f, n s, u, fund, funds, and, sand}*, the character string *S=fundsand* has the set of all possible tokenizations E_D*(fundsand)={f/u/n/d/s/a/n/d, fund/s/a/n/d, f/u/n/d/s/and, funds/a/n/d, f/u/n/d/sand, find/s/and, funds/and, fund/sand}*, where $E_D(S)$ denotes the set of exhaustive tokenizations for character string *S* over dictionary *D*.

EXAMPLE 2: Given the dictionary *D={a, b, c, d, ab, bc, cd, abc, bcd}*, the character string *S=abcd* has the set of exhaustive tokenizations E_D*(abcd)={a/b/c/d, ab/c/d, a/bc/d, a/b/cd, abc/d, ab/cd, a/bcd}* This set can be depicted as in Figure 1 below.

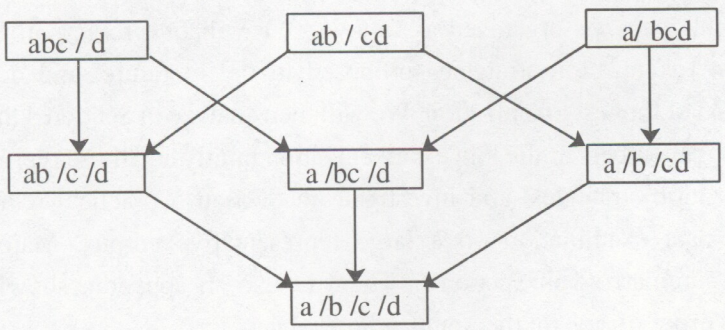

Figure 1 *The exhaustive tokenization set*

E_D(abcd)={a/b/c/d, a/b/cd, a/bc/d, a/bcd, ab/c/d, ab/cd, abc/d}.

Also illustrated in Figure 1 is the **cover relation** between different tokenizations: a word string covers another if the concatenation of some words in the latter reproduces the former. For instance, *ab/cd covers* both *ab/c/d* and *a/b/cd* but not *a/bc/d*. As has been proven in (Guo, 1997), the cover relation is a (**reflexive**) **partial order**. Thus, a set of tokenizations forms a **partially ordered set**, or simply a **poset** (e.g., Kolman and Busby, 1987), on the cover relation. Poset is an important type of mathematical structure with many neat mathematical properties. In short, the collection of different tokenizations for a character string is not merely a mixture but is well structured.

The procedure of **forward maximum tokenization** (**FT**) is: given a character string to be tokenized and a tokenization dictionary, match the string against the dictionary, find the first longest match from the beginning of the string, take it out as the first token, and then repeat the procedure until no more tokens can be taken out.

EXAMPLE 1 (CONT.): The character string *S=fundsand* has the unique forward maximum tokenization *funds/and*, i.e., $F_D(fundsand)=\{funds/and\}$, where $F_D(S)$ denotes the set of forward maximum tokenizations for character string *S* over dictionary *D*. Note that three dictionary tokens, *f, fund*, and *funds*, match the character string *S=fundsand* from its beginning, but that *funds* is the longest one among the three and, hence, the first token produced.

EXAMPLE 2 (CONT.): The character string *S=abcd* has the word string *abc/d* as its sole *FT*, i.e., $F_D(S)=\{abc/d\}$.

The procedure of **backward maximum tokenization** (**BT**) is the same as that of forward maximum tokenization except that, while the dictionary matching and tokenization process goes from left to right in *FT*, it goes in the reverse direction, from right to left, in *BT*.

EXAMPLE 1 (CONT.): $B_D(fundsand)=\{fund/sand\}$, where $B_D(S)$ denotes the set of backward maximum tokenizations for character string *S* over dictionary *D*. Note that three dictionary tokens, *d, and*, and *sand*, match the character string *S=fundsand* from its ending, but that *sand* is the longest one among the three and, hence, the first token produced.

EXAMPLE 2 (CONT.): $B_D(abcd)=\{a/bcd\}$.

The procedure of **forward-backward maximum tokenization** (**FBT**) is actually not independent but is the union of *FT* and *BT*. That is, a word string is an *FBT* tokenization if it is either *FT* or *BT*. *FBT* is also known as **dual-direction maximum tokenization** (**DT**).

EXAMPLE 1 (CONT.): $D_D(fundsand)=\{funds/and, fund/sand\}$, where $D_D(S)$ denotes the set of forward-backward maximum tokenizations for character string *S* over dictionary *D*. Note that the tokenization *funds/and* is from *FT*, and that *fund/sand* is from *BT*.

EXAMPLE 2 (CONT.): $D_D(abcd)=\{abc/d, a/bcd\}$.

Note that, by definition, it is always true that $D_D(S)=F_D(S)\cup B_D(S)$.

A word string is a **shortest tokenization** (**ST**) if it contains the minimum number of words possible; i.e., it has the shortest word string length.

EXAMPLE 1 (CONT.): S_D*(fundsand)={funds/and, fund/sand}*, where $S_D(S)$ denotes the set of shortest tokenizations for character string *S* over dictionary *D*. Note that both tokenizations *funds/and* and *fund/sand* have word string length 2, and that there is no shorter valid tokenization for *S=fundsand*. The three-word tokenization *fund/s/and* is not an *ST*.

EXAMPLE 2 (CONT.): S_D*(abcd)={abc/d, ab/cd, a/bcd}*.

A word string is a **profile tokenization** (**PT**), if it, by itself, is a profile token, or if it contains a profile token and the left and right substrings with respect to the profile token are profile tokenizations. A dictionary token is a **profile token** of a character string to be tokenized, if it matches the character string and is not part of any other dictionary token matching the same string at the same position. Simply put, profile tokens are the most prominent tokens in a sentence, and profile tokenization segments a sentence by repeatedly identifying profile tokens. Profile tokenization takes a type of *island-driven* strategy with profile tokens as islands.

EXAMPLE 1 (CONT.): P_D*(fundsand)={funds/and, fund/sand}*, where $P_D(S)$ denotes the set of profile tokenizations for character string *S* over dictionary *D*. Note that both *funds* and *sand* are profile tokens of the character string *S=fundsand*, but that *fund* and *and* are not.

EXAMPLE 2 (CONT.): P_D*(abcd)={abc/d, a/bcd}*. Note that only *abc* and *bcd* are profile tokens; thus, *ab/cd* is not a profile tokenization as it does not contain any profile token.

A word string is a **critical tokenization** (**CT**) if it is not covered by any other tokenization. This implies that no valid tokenization can be produced by concatenating adjacent words in any critical tokenization. In terms of the tokenization *poset* described above, a critical tokenization (Guo, 1997) is precisely a **minimal poset element** (Kolman and Busby, 1987).

EXAMPLE 1 (CONT.): C_D*(fundsand)={funds/and, fund/sand}*, where $C_D(S)$ denotes the set of critical tokenizations for character string *S* over dictionary *D*. Note that *fund/s/and* is not a critical tokenization since by concatenating its first two words *fund* and *s*, the valid tokenization *funds/and* can be reproduced.

EXAMPLE 2 (CONT.): C_D*(abcd)={abc/d, ab/cd, a/bcd}*. Note that *a/b/cd* is not a critical

tokenization since by concatenating its first two words *a* and *b*, the valid tokenization *ab/cd* can be reproduced.

The complex example below is purposely crafted to further clarify the above reviewed maximum tokenization strategies.

EXAMPLE 3: Given the dictionary *D={a, b, c, d, e, f, g, h, i, j, k, l, abc, abcd, cdefgh, defg, ghijkl, hij}*, the character string *S=abcdefghijkl* has the *token graph* (Wang, 1989) shown in Figure 2, and has the different types of tokenizations summarized in Figure 3.

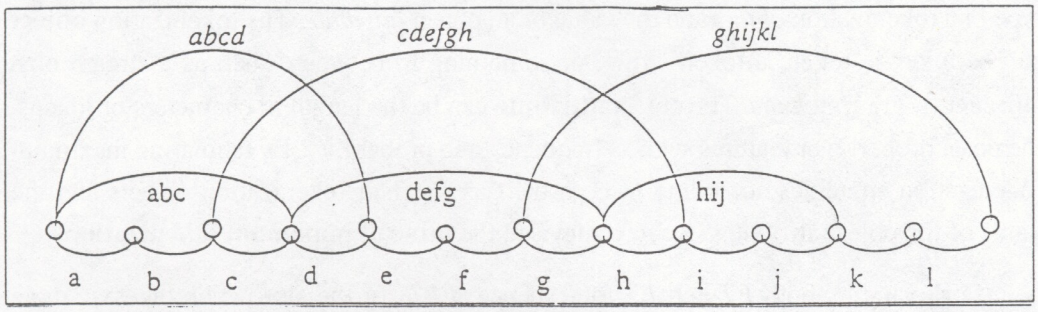

Figure 2 Token graph.

```
Character String
   S = abcdefghijkl

Tokenization Dictionary
   D = {a, b, c, d, e, f, g, h, I, j, k, l, abc, defg, hij,
      abcd, cdefgh, ghijkl }

Forward Maximum Tokenization
   FD(S) = { abcd/e/f/ghijkl }

Backward Maximum Tokenization
   BD(S) = { abcd/e/f/ghijkl }

Forward-Backward Maximum Tokenization
   DD(S) = { abcd/e/f/ghijkl }

Shortest Tokenization
   SD(S) = { abcd/e/f/ghijkl }

Profile Token
   {abcd, cdefgh, ghijkl}

Profile Tokenization
   PD(S) = { abcd/e/f/ghijkl, a/b/cdefgh/i/j/k/l }

Critical Tokenization
   CD(S) = { abcd/e/f/ghijkl, abc/defg/hij/k/l,
             a/b/cdefgh/i/j/k/l}
```

Figure 3 Different maxium tokenizations

3. Longest Tokenization

In the preceding section, we reviewed various representative tokenization strategies following the *principle of maximum tokenization*. This leads us to postulate here that the *core* of the principle is the search for certain kinds of *extremes* - the longest affix tokens for *FT, BT* and *FBT*, the shortest token string length for *ST*, the profile tokens for *PT*, the minimum poset elements for *CT*, and the like.

Furthermore, we find that such *extremes* can be categorized according to both the type of a tokenization *object* and the value of an object *attribute*. The tokenization **object** can be a single token, a token string, or something in between, such as a stream of n adjacent (n-gram) tokens. The object **attribute** can be the length in characters or tokens, the poset property, or features such as frequency and probability. By tabulating maximum tokenization strategies according to both the type of their tokenization objects and the value of the object attributes, we have devised the **table of maximum tokenizations**.

For example, both *FT* and *BT* (and, hence, *FBT*) fit the slot which takes a single token as a tokenization object and seeks to maximize the object length in characters. *ST* takes a complete token string as an object but goes on to minimize the object length in tokens. On the other hand, *PT* takes a single token while *CT* takes a whole token string as an object but both seek minimum poset elements. These are tabulated in Table 1. (The row labeled *Token N-gram* will be explained shortly.)

Extreme		*Attribute*	
		Object Length	*Poset Property*
Object	*Single Token*	*FT, BT, FBT*	*PT*
	Token N-gram	*LR(n), RL(n), DD(n)*	*PT(n)*
	Token String	*ST*	*CT*

Table 1. *Table of maximum tokenizations.*

By studying Table 1, it becomes trivial to understand that the *Chen and Liu Heuristic* also fits the table as a tokenization strategy that takes a token trigram (consecutive triple tokens) as an object and seeks to maximize the object length in characters. Since the *Chen and Liu Heuristic* was originally implemented (Chen and Liu, 1992) in

the forward direction from left to right, it will hereafter be referred to as **left-to-right token trigram maximum tokenization** or **LR(3).**

After found the position of *LR(3)* in the *table of maximum tokenizations*, some natural generalizations arise. Firstly, instead of a token trigram, the tokenization object can be a *token n-gram* (consecutive *n* tokens) of any order *n*. This leads to the general **left-to-right token n-gram maximum tokenization** or **LR(n)**. Notice that *LR(1)* is, by definition, the same as *FT*, i.e., *LR(1)=FT*. It is also trivial to demonstrate that, for each character string *S*, there exists a sufficiently large number *N* such that *LR(n)=ST* holds for any $n \geq N$, i.e., $LR(\infty)=ST$. That is, *FT* and *ST* are the two polar members of the *LR(n)* family.

Secondly, parallel to what *BT* is to *FT*, by taking a token n-gram as an object and by seeking to maximize the object length in characters in the reverse direction *from right to left*, we obtain the general **right-to-left token n-gram maximum tokenization** or **RL(n)**. Similarly, *BT* and *ST* are the two polar members of the *RL(n)* family, i.e., *RL(1)=BT* and $RL(\infty)=ST$.

Thirdly, parallel to what *FBT* is to *FT* and *BT*, by taking a token n-gram as an object and by seeking to maximize the object length in characters in the dual directions *from both ends*, we obtain the general **dual-direction token n-gram maximum tokenization** or **DD(n)**. That is, *DD(n)=LR(n)+RL(n)*. As in the above, *FBT* and *ST* are the two polar members of the *DD(n)* family, i.e., *DD(1)=FBT* and $DD(\infty)=ST$.

EXAMPLE 1 (CONT.): For any $n \geq 2$, we have $F_D(n, fundsand)=$ $B_D(n, fundsand)=$ $D_D(n, fundsand)=\{funds/and, fund/sand\}$, where $F_D(n,S)$ denotes the set of *LR(n)* tokenizations for character string *S* over dictionary *D*. Similarly, we have $B_D(n,S)$ *for RL(n)* and $D_D(n,S)$ *for DD(n)*. Note that, by definition, $F_D(1,S)=F_D(S)$, $B_D(1,S)=B_D(S)$, and $D_D(1,S)=D_D(S)$.

EXAMPLE 2 (CONT.): $F_D(n, abcd)=B_D(n, abcd)=D_D(n, abcd)=\{abc/d, a/bcd\}$ for $n \geq 2$.

EXAMPLE 3 (CONT.): The results of *LR(n)*, *RL(n)* and *DD(n)* are in Figure 4 as follows.

```
Left-to-Right Token N-gram Maximum Tokenization
   F_D(1,S) = { abcd/e/f/ghijkl }
   F_D(2,S) = { abc/defg/hij/k/l }
   F_D(3,S) = { abc/defg/hij/k/l }
   F_D(n,S) = { abcd/e/f/ghijkl }      for n≥4

Right-to-Left Token N-gram Maximum Tokenization
   B_D(n,S) = { abcd/e/f/ghijkl }      for any n

Dual-Direction Token N-gram Maximum Tokenization
   D_D(1,S) = { abcd/e/f/ghijkl }
   D_D(2,S) = { abc/defg/hij/k/l, abcd/e/f/ghijkl }
   D_D(3,S) = { abc/defg/hij/k/l, abcd/e/f/ghijkl }
   D_D(n,S) = { abcd/e/f/ghijkl }      for n≥4
```

Figure 4 *Longest tokenization results.*

EXAMPLE 4: The character string S is composed of $2k$ different characters, $S=c_1...c_k...c_{2k}$. The tokenization dictionary D is made up of these $2k$ different characters plus the two special tokens $w_a=c_1...c_k$ *and* $w_b=c_k...c_{2k}$, $D=\{c_1, ..., c_k, ..., c_{2k}, w_a=c_1...c_k, w_b=c_k...c_{2k}\}$. Figure 5 lists the tokenization results.

```
Left-to-Right Token N-gram Maximum Tokenization
   F_D(n,S)={ c1/.../c_{k-1}/w_b }      for n≥k
   F_D(n,S)={ w_a/c_{k+1}/.../c_{2k} }  for n<k

Right-to-Left Token N-gram Maximum Tokenization
   B_D(n,S)={ c1/.../c_{k-1}/w_b }      for all n and k

Dual-Direction Token N-gram Maximum Tokenization
   D_D(n,S)={ c1/.../c_{k-1}/w_b }              for n≥k
   D_D(n,S)={ c1/.../c_{k-1}/w_b, w_a/c_{k+1}/.../c_{2k} } for n<k

Shortest Tokenization
   S_D(S)={ c1/.../c_{k-1}/w_b }

Profile Tokenization
   C_D(S)={ c1/.../c_{k-1}/w_b, w_a/c_{k+1}/.../c_{2k} }

Critical Tokenization C_D(S)={ c1/.../c_{k-1}/w_b,
w_a/c_{k+1}/.../c_{2k} }
```

Figure 5 *different maximum tokenization results.*

With the *table of maximum tokenizations*, many more maximum tokenization strategies can be naturally devised. For instance, instead of conducting tokenization in a fixed direction of either left-to-right or right-to-left, we can also do by repeatedly

searching for the object that has the *global* maximum object length in the substring to be tokenized. To have the paper more focused, however, we will restrict ourselves only to the above three types of generalizations.

It is worth noting that the pursuit of the *maximum length* is the single essential common characteristic inherited from the *Chen and Liu Heuristic* and shared by all the members of *LR(n), RL(n)* and *DD(n)*. It is, thus, logical to collectively refer to these tokenization strategies as **Longest Tokenization** (**LT**).

It is also worth noting the association between *LR(n)* tokenization and *LR(n)* parsing (Aho and Ullman, 1972); i.e., both make decisions by looking ahead several tokens. As tokenization has been stated in the literature as a special type of parsing, it is reasonable to regard *LR(n)* tokenization as a result of the transplantation of the general *LR(n)* parsing strategy to sentence tokenization.

4. Theoretical Relationships

This section explores logical relationships both between members of the longest tokenization family and between longest tokenizations and all the representative maximum tokenization strategies reviewed in Section 2. The results will be presented as four theorems, while some indications on the proof of these theorems will be given in Appendix.

Within the longest tokenization family, by definition, we always have $D_D(n,S)= F_D(n,S) \cup B_D(n,S)$, which implies $F_D(n,S) \subseteq D_D(n,S)$ and $B_D(n,S) \subseteq D_D(n,S)$. That is, both left-to-right and right-to-left token n-gram maximum tokenizations are always subclasses of dual-direction token n-gram maximum tokenization of the same order. The following theorem makes it clear that these are the only universally held logical relationships between members of the longest tokenization family.

THEOREM 1: For any positive n and m, none of the following relationships universally holds:

(1) $F_D(n,S) \subseteq F_D(m,S)$, $n \neq m$, (2) $B_D(n,S) \subseteq F_D(m,S)$, (3) $D_D(n,S) \subseteq F_D(m,S)$,

(4) $F_D(n,S) \subseteq B_D(m,S)$, (5) $B_D(n,S) \subseteq B_D(m,S)$, $n \neq m$, (6) $D_D(n,S) \subseteq B_D(m,S)$,

(7) $F_D(n,S) \subseteq D_D(m,S)$, $n \neq m$, (8) $B_D(n,S) \subseteq D_D(m,S)$, $n \neq m$, (9) $D_D(n,S) \subseteq D_D(m,S)$, $n \neq m$

That is, for each of the relationships listed above, and for any posivite n and m, there exist a character string S and dictionary D such that the relationship does not hold.

This theorem implies that, except for *LR(n)* and *RL(n)* to *DD(n)*, no tokenization strategy of the longest tokenization family is always part of or the same as another tokenization strategy of the family. In short, all the members of the longest tokenization family are distinct and unique.

By definition, both $F_D(n,S)=F_D(S)$ *and* $B_D(n,S)=B_D(S)$ hold for $n=1$. Since every longest affix token is also a profile token by definition, there always exist $F_D(S)\subseteq P_D(S)$ and $B_D(S)\subseteq P_D(S)$. Hence, both $F_D(n,S) \subseteq P_D(S)$ and $B_D(n,S) \subseteq P_D(S)$ hold for $n=1$. In addition, as $D_D(n,S)=D_D(S)$ for $n=1$ and $D_D(n,S)=F_D(n,S)\cup B_D(n,S)$, there is $D_D(n,S)\subseteq P_D(S)$ for $n=1$. That is, for order $n=1$, members of the longest tokenization family are always subclasses of profile tokenization. The following theorem makes it clear that these are the only universally held logical relationships between profile tokenization and members of the longest tokenization family.

THEOREM 2: For $n>1$, none of the following relationships universally holds:

(1) $P_D(S) \subseteq F_D(n,S)$, (2) $F_D(n,S) \subseteq P_D(S)$,
(3) $P_D(S) \subseteq B_D(n,S)$, (4) $B_D(n,S) \subseteq P_D(S)$,
(5) $P_D(S) \subseteq D_D(n,S)$, (6) $D_D(n,S) \subseteq P_D(S)$.

That is, for each of the relationships listed above, there exist a character string S and dictionary D such that the relationship does not hold.

This theorem implies that, except for *LR(1)*, *RL(1)* and *DD(1)*, which are true subclasses of profile tokenization, no tokenization strategy of the longest tokenization family is *always* part of or the same as profile tokenization, nor the other way around.

For the relationship between shortest tokenization and members of the longest tokenization family, it has been seen in the previous section that $LR(\infty)=RL(\infty)=DD(\infty)=ST$. Let us refer a character string as having **at least** N **tokens** over a tokenization dictionary, if the shortest tokenization of the character string has N tokens. There is the following theorem.

THEOREM 3: For any character string S with at least N tokens over a tokenization dictionary D, the following three relationships hold true for any $n\geq N$:

(1) $F_D(n,S)=S_D(S)$, (2) $B_D(n,S)=S_D(S)$, (3) $D_D(n,S)=S_D(S)$.

However, for each of the three relationships, and for any $n<N$, there exists a character string S, which has at least N tokens over a tokenization dictionary D, such that the

relationship does not hold. That is, N is the *supremum* (the least upper bound) that makes all the longest tokenization strategies with an order the same or higher than it equivalent to shortest tokenization. In the next section, we will try to answer how *high* (or actually how *low*) the supremum can be in practice by investigating a representative corpus.

For the relationship between critical tokenization and members of the longest tokenization family, it has been proven in (Guo, 1997) that both $F_D(S) \subseteq C_D(S)$ and $B_D(S) \subseteq C_D(S)$ hold. Thus, $D_D(S) \subseteq C_D(S)$ also universally holds true. That is, for order $n=1$, members of the longest tokenization family are subclasses of critical tokenization. This is actually a fact for longest tokenization on the whole.

THEOREM 4: $\bigcup_{n=1}^{\infty} F_D(n,S) \cup B_D(n,S) \subseteq C_D(S)$ holds for any character string S and tokenization dictionary D. Moreover, there exist a character string S and tokenization dictionary D such that $\bigcup_{n=1}^{\infty} F_D(n,S) \cup B_D(n,S) \neq C_D(S)$

This theorem implies that the family of longest tokenization on the whole still can not produce any word string that is not a critical tokenization. In other words, at least for longest tokenization, all tokenizations obeying the principle of maximum tokenization have already been discovered by critical tokenization.

While the first two theorems confirm that longest tokenization has really contributed a rich set of *distinct* and *unique* maximum tokenization strategies, the last two theorems reveal that *no surprise* can be expected as to shortest tokenization and critical tokenization. The theoretical relationships discovered in this section, together with those given in (Guo, 1997), can be figuratively summarized as shown in Figure 6 below. Arrows in this figure are pointing from super-class tokenization strategies to their respective sub-classes. For instance, *LR(n)* is a subclass of *CT*.

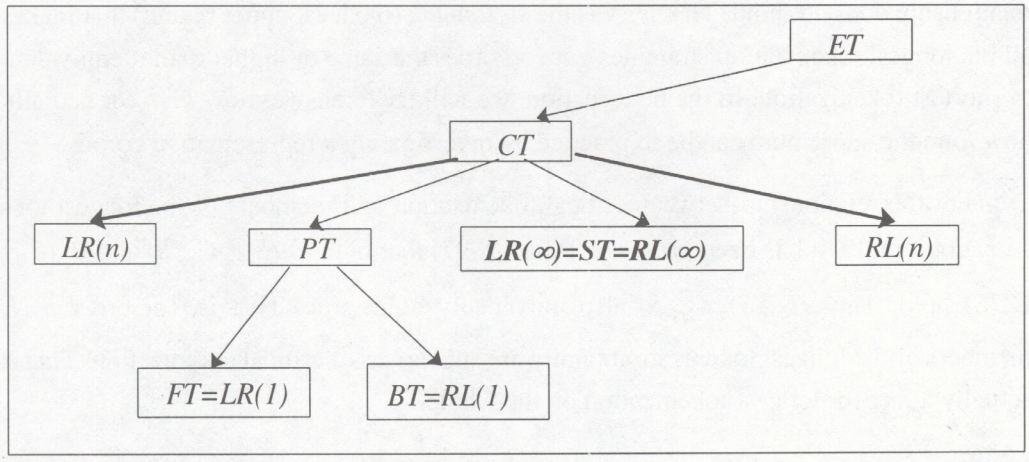

Figure 6 *Theoretical relationships among different tokenization strategies.*

5. Practical Relationships

This section demonstrates practical behaviors of different longest tokenization strategies as compared to each other and to other maximum tokenization strategies. In particular, we will see in this section that, at least on the Chinese *PH* corpus, except for trivial cases of $n=1$ where the longest tokenization strategy is by definition the forward and/or backward maximum tokenization, there is practically no difference between longest tokenization and shortest tokenization.

The two resources used in this study are the Chinese *PH* corpus (Guo, 1993) and the *Beihang* dictionary (Liu and Liang, 1989). The Chinese *PH* corpus is a collection of *4* million Chinese characters of news articles from the *Xinhua* News Agency, China. The *Beihang* dictionary is a collection of about *50,000* word-like tokens, each of which occurs at least *5* times in a balanced collection of more than 20 million Chinese characters.

What is unique in the *PH* corpus is that all *unambiguous* token boundaries with respect to the *Beihang* dictionary have been marked. For this study, we have extracted from the *PH* corpus all multi-character fragments in between adjacent unambiguous token boundaries that are not entries in the *Beihang* dictionary. This is the same as extracting all maximum length fragments with *critical ambiguities* (Guo, 1997), or *disjunctive* or *overlapping* type (Webster and Kit, 1992) tokenization ambiguities. In this paper, such fragments are referred to as **critical fragments** (**CF**).

There are *14,984* distinct critical fragments which cumulatively occur *49,308* times in the *PH* corpus. Their length distribution is given in Table 2 below. It has been observed

that most critical fragments are very short. About 90% of the critical fragments have merely 3 or 4 characters each, and there are only two fragments each with *11* characters. The average length of these critical fragments is only *3.62 (=178,341/49,308)* Chinese characters.

(1) Length (char.)	(2) Occurrence (num.)	(3)=(2)/49308 Occurrence (%)	(4)=(1)*(2) Volume (char.)	(5) Type (num.)	(6)=(5)/14984 Type (%)
3	25492	51.70	76476	6539	43.64
4	19770	40.09	79080	6504	43.41
5	2391	4.85	11955	1184	7.90
6	1125	2.28	6750	585	3.90
7	258	0.52	1806	105	0.70
8	216	0.44	1728	43	0.29
9	18	0.04	162	16	0.11
10	34	0.07	340	6	0.04
11	4	0.01	44	2	0.01
Total	**49308**	**100.00**	**178341**	**14984**	**100.00**

Table 2. *Critical fragment length distribution.*

We will only use these critical fragments rather than the whole *PH* corpus to compare various maximum tokenization strategies. This unique experimental configuration, as compared with the general practice in the literature, can be justified with the following three observations: (1) All the maximum tokenization strategies studied in this paper make the same correct tokenization decision at all *unambiguous* token boundaries.[3] (2) They make the same tokenization decision for all dictionary-entry character fragments in between unambiguous token boundaries.[4] (3) They each produces invariant tokenization results for any critical fragment regardless of the specific context in which the critical fragment occurs.[5]

---

3. However, almost none of them can tell where the unambiguous boundaries are.

4. In other words, they have the same performance at the resolution of *hidden ambiguity* (Guo, 1997), or *combinational* or *conjunctive* type ambiguity (Webster and Kit, 1992).

5. That is, all critical fragments are *self-contained* with respect to all the maximum tokenization strategies studied in this paper. Note that this does *not* imply *context-independence* for any of these maximum tokenization strategies. The self-containment in tokenization only holds for *critical fragments* (Guo, 1997), *not* for arbitrary character strings. After all, the sole purpose for employing larger tokenization objects is to utilize more context constraints. Nevertheless, recognition of the self-containment property of critical fragments reveals the limitation of the mainstream thinking of introducing context restrictions by enlarging tokenization objects. In fact, for critical fragments, not only does the self-containment property hold for maximum tokenization, but it also holds for its correct tokenizations. More precisely, we have observed the very strong tendency of *one tokenization per source* (Guo 1997).

Moreover, by comparing maximum tokenizations exclusively on the set of critical fragments, inflated performance reports can largely be avoided. For instance, it has been noted that more than 98% of the token boundaries in the *PH* corpus are unambiguous, and that almost all critical fragments have only two alternative critical tokenizations each. Consequently, every maximum tokenization strategy should be able to achieve tokenization accuracy no worse than 99%, which becomes the dominant denominator that makes differences among various maximum tokenization strategies insignificant. In contrast, by concentrating on the set of critical fragments, the large denominator (the known common part) is purposely removed; thus, differences among various maximum tokenization strategies can be better highlighted.

Recall that only *14,984* distinct critical fragments cumulatively occur *49,308* times in the *PH* corpus. Guided by the theoretical results given in the previous section, their shortest tokenizations were generated, and the results are summarized in Table 3 below. At one extreme, close to 98% of the critical fragment occurrences have exactly two tokens each in their shortest tokenization. At the other extreme, only one critical fragment (see Table 4 below) with 5 tokens in its shortest tokenization occurs once in the *PH* corpus.

Length	Occurrences	%	Types	%
2	48110	97.57	14221	94.91
3	1171	2.37	742	4.95
4	26	0.05	20	0.13
5	1	0.00	1	0.01
Total	**49308**	**100.00**	**14984**	**100.00**

Table 3. *Shortest tokenization length distribution.*

Based on the theorems given in the previous section, the first observation is that $LR(n)=RL(n)=DD(n)=ST$ holds for any $n\geq 5$ on the *PH* corpus over the *Beihang* dictionary. In addition, for the single critical fragment with at least 5 tokens, the tokenization results are listed in Table 4. In short, $LR(4)=ST$ holds with one miss; $RL(4)=ST$ holds with no exception; and $DD(4)=ST$ holds perfectly in the *PH* corpus.

LR(2)	RL(2)	LR(3)	RL(3)	LR(4)	RL(4)	ST	Tokenizations
1	0	1	0	1	1	1	國 外交部 長進 行會 談
1	0	1	0	1	1	1	國 外交部 長進 行 會談
0	1	1	1	1	1	1	國 外交部 長 進行 會談
0	1	0	1	1	1	1	國 外交 部長 進行 會談
0	1	0	1	0	1	1	國外 交 部長 進行 會談

Table 4. *Tokenizations for the single critical fragment with at least 5 tokens.*

Similarly, to compare *ST* with *LT* for $n=3$, it is only necessary to investigate fragments with at least 4 tokens. In addition to the single critical fragment with at least 5 tokens presented in Table 4 above, tokenization results for the only 26 critical fragment occurrences with at least 4 tokens under *ST, LR(3)* and *RL(3)* are listed in Table 5. Note that the shaded cells in the leftmost column are for the 17 *ST*'s that are not *RL(3)*, and the shaded cells in the rightmost column are for the 16 *ST*'s that are not *LR(3)*. There is no critical fragment found that has an *LR(3)* or *RL(3)* tokenization which is not an *ST* tokenization. Moreover, $LR(3) \cup RL(3) = ST$ holds in the *PH* corpus.

Shade for none RL(3)			**Shade for none LR(3)**
收入 超過 當地 平均			
鐵路 局面 臨近 年來			
中國 人民 政治 生活			
全國 人民代表大會 審議 和		全國 人民代表大會 審 議和	
全國 人民代表大會 審議 和		全國 人民代表大會 審 議和	
全國 人民代表大會 審議 和		全國 人民代表大會 審 議和	
日中 國人 民 政治協商會議	日中 國人民 政治協商會議		日 中國人民 政治協商會議
項目 的確 立 意味著	項目 的 確立 意味著		項 目的 確立 意味著
中國 人民 和 解放軍	中國 人 民和 解放軍		中 國人 民和 解放軍
並進 行部 分設 備	並進 行部 分 設備	並進 行 部分 設備	並 進行 部分 設備
出國 內人 文學科	出國 內人 文 學科	出國 內 人文 學科	出 國內 人文 學科
對外 來人 口實 施	對外 來人 口 實施	對外 來 人口 實施	對 外來 人口 實施
廣大 海內 外僑 胞	廣大 海內 外 僑胞	廣大 海 內外 僑胞	廣 大海 內外 僑胞
國內 外公 開發 行	國內 外公 開 發行	國內 外 公開 發行	國 內外 公開 發行
國內 外公 開發 行	國內 外公 開 發行	國內 外 公開 發行	國 內外 公開 發行
和文 化工 作戰 線	和文 化工 做 戰線	和文 化 工作 戰線	和 文化 工作 戰線
切切 實實 地利 用	切切 實實 地 利用	切切 實實地 利用	切 切實 實地 利用
取得 了解 放大 陸	取得 了解 放 大陸	取得 了 解放 大陸	取 得了 解放 大陸
上街 頭發 表演 說	上街 頭發 表 演說	上街 頭 發表 演說	上 街頭 發表 演說
提高 人民 生活 水	提高 人民 生 活水	提高 人民 生活 水	提 高人 民生 活水
提高 人民 生活 水	提高 人民 生 活水	提高 人民 生活 水	提 高人 民生 活水
提高 人民 生活 水	提高 人民 生 活水	提高 人民 生活 水	提 高人 民生 活水
修建 中小 學校舍	修建 中小 學 校舍	修建 中 小學 校舍	修 建中 小學 校舍
修建 中小 學校舍	修建 中小 學 校舍	修建 中 小學 校舍	修 建中 小學 校舍
職工 人人 爲生 產	職工 人人 爲 生產	職工 人 人爲 生產	職 工人 人爲 生產
東經 營房 地 產並	東經 贏 房地產 並	東 經營 房地產 並	東經 營房 地產 並

Table 5. *LR(3), RL(3) and ST for CF's of at least 4 tokens.*

The same procedure was also applied to compare *ST* and *LT* for $n=2$. The overall results are summarized in Table 6 below. In short, the *49,308* critical fragments extracted

from the *PH* corpus altogether produce *77,935* shortest tokenizations. Taking shortest tokenization as a reference, longest tokenizations produce no extra and miss few. For instance, the only flaw in dual-direction token n-gram maximum tokenization with respect to shortest tokenization is *DD(2)*, which has 7 absences.

Theory	*Correct*	*Miss*	*Extra*	*Recall (%)*	*Precision (%)*
			LR(n) to ST		
LR(2)=ST	77253	682	0	99.13	100.00
LR(3)=ST	77918	19	0	99.98	100.00
LR(4)=ST	77934	1	0	100.00	100.00
LR(n≥5)=ST	77935	0	0	100.00	100.00
			RL(n) to ST		
RL(2)=ST	77273	662	0	99.15	100.00
RL(3)=ST	77919	18	0	99.98	100.00
RL(4)=ST	77935	0	0	100.00	100.00
RL(n≥5)=ST	77935	0	0	100.00	100.00
			DD(n) to ST		
DD(2)=ST	77928	7	0	99.99	100.00
DD(3)=ST	77935	0	0	100.00	100.00
DD(4)=ST	77935	0	0	100.00	100.00
DD(n≥5)=ST	77935	0	0	100.00	100.00

Table 6. *Longest tokenizations are practically the same as shortest tokenization.*

In summary, what has been confirmed in this section is that except for *n=1*, where *LR(1)=FT, RL(1)=BT* and *DD(1)=FBT* by definition, there is practically no difference between *LT* and *ST*; i.e., *LR(n)=RL(n)=DD(n)=ST* for any *n≥2*.

6. Summary

The objective of this paper has been to enhance our knowledge and understanding of the powerful **principle of maximum tokenization**. The actual work has been to establish the notion of **longest tokenization**, a rich set of tokenization strategies following the principle of maximum tokenization. This has been done in four steps.

The first step was to form the **table of maximum tokenizations** through a critical review of several representative maximum tokenization strategies frequently seen in the literature. According to the *table of maximum tokenizations*, all the maximum tokenization strategies can be viewed as searching for *extremes* that are values of certain *attributes* of certain tokenization *objects*.

The second step was to propose the notion of **longest tokenization** through

identification and generalization of the specific tokenization object and the object attribute used in the original *Chen and Liu Heuristic*. Briefly, longest tokenization takes a token n-gram as an object and seeks to maximize the object length in characters. Variations come from factors such as the *order* of the token n-gram (number of tokens) and the *direction* (left-to-right, right-to-left, or both) of tokenization.

The third step was to establish **theoretical positions** for longest tokenization in the already crowded family of maximum tokenizations. This was done by studying *theoretical* relationships both within the longest tokenization family and between longest tokenization and other maximum tokenization strategies. It has been proven that: *forward maximum tokenization, backward maximum tokenization, forward-backward maximum tokenization*, and *shortest tokenization* are all special cases of the longest tokenization family, and that all the members of the longest tokenization family are true subclasses of *critical tokenization*. Other than that, all the members of the longest tokenization family are distinct and unique for all known maximum tokenization strategies studied in this paper, including *profile tokenization*.

The fourth step was to find **practical positions** for longest tokenization. This was done through a detailed data investigation on the Chinese *PH* corpus. It has been verified that, except for trivial cases where longest tokenization is the same as forward and/or backward maximum tokenization, there is virtually no difference between longest tokenization and shortest tokenization.

In conclusion, a rich set of maximum tokenization strategies called longest tokenization has been well established in this paper. However, this paper has also revealed that **no surprise** can be expected from longest tokenization as the *essence* of the principle of maximum tokenization is fully captured by critical tokenization, and the *essence* of length-oriented realization of the principle is fully captured by token-based forward/backward maximum tokenization and sentence-based shortest tokenization.

In this paper, we have also shown, in the *table of maximum tokenizations*, the **token n-gram profile tokenization PT(n)**, which is another rich family of maximum tokenization strategies each of which taking a token n-gram as a tokenization object and searches for the minimum element of the object poset. By studying profile tokenization in a way analogue to what has been done in this paper, results parallel to what have been achieved for longest tokenization can be established: theoretically, all token n-gram profile tokenization strategies are distinct and unique, but all are subclasses of critical tokenization; practically, there is virtually no difference between profile tokenization and critical tokenization. In particular, except for a single miss in *PT(1), PT(n)=CT* holds

perfectly true for any positive n in the Chinese *PH* corpus.

It is worth noting that, while logically not universally hold true, both $LR(n)<LR(n+1)$ and $RL(n)<RL(n+1)$ are actually realized in the *PH* corpus without exception. We are thus interested in knowing the sufficient and necessary conditions under which the monotonic relationships hold true.

References

Aho, Alfred V.; and Ullman, Jeffrey D. *The Theory of Parsing, Translation, and Compiling, Volume 1: Parsing*. Prentice-Hall, Inc, 1972.

Chen, Keh-Jiann "Chinese Sentence Parsing, " A Tutorial at the 1996 International Conference on Chinese Computing (ICCC'96). Singapore, 1996.

Chen, Keh-Jiann and Shing-Huan Liu "Word Identification for Mandarin Chinese Sentences." In *Proceedings of the 14th International Conference on Computational Linguistics* (COLING'92):101-107, Nantes, France, 1992.

Gan, Kok-Wee; Palmer, Martha; and Lua, Kim-Teng . A Statistically Emergent Approach for Language Processing: Application to Modeling Context Effects in Ambiguous Chinese Word Boundary Perception. *Computational Linguistics* 22.4(1996): 531-553.

Guo, Jin , "PH - A Free Chinese Corpus," *Communications of COLIPS*, 3.1 (1993): 45-48.

Guo, Jin, "Critical Tokenization and its Properties," *Computational Linguistics*, 23.4 (1997):569-596.

Huang, Changning; and Xia, Ying (eds.) *Essays on Language Information Processing*, Tsinghua University Press, Beijing, 1996.

Huang, Chu-Ren, Keh-Jiann Chen, and Lili Chang. "Segmentation Standard for Chinese Natural Language Processing," In the *Proceedings of the 16th International Conference on Computational Linguistics* (COLING'96), Copenhagen, Denmark, 1996.

Kolman, Bernard; and Busby, Robert C. *Discrete Mathematical Structures for Computer Science*, 2nd edition. Prentice-Hall, Inc, 1987.

Liang, Nanyuan. "On Computer Automatic Word Segmentation of Written Chinese," *Journal of Chinese Information Processing*, 1.1(1986).

Liu, Yongquan. *Language Modernization and Computer*. Wuhan University Press, 1986.

Liu, Yuan and Nanyuan Liang. *Contemporary Chinese Common Word Frequency Dictionary* (Phonetically Ordered Version). Yuhang Press, Beijing, 1989.

Liu, Yuan, Tan, Qiang, and Shen, Xukun. *Contemporary Chinese Language Word Segmentation*

Specification for Information Processing and Automatic Word Segmentation Methods, Tsinghua University Press, Beijing, 1994.

Sproat, Richard, Chilin Shih, Villiam Gale, and Nancy Chang. "A Stochastic Finite-State Word-Segmentation Algorithm for Chinese," *Computational Linguistics*, 22.3(1996): 377-404.

Su, Keh-Yih, Tung-Hui Chiang and Jing-Shin Chang. "An Overview of Corpus-Based Statistics-Oriented (CBSO) Techniques for Natural Language Processing," *International Journal of Computational Linguistics and Chinese Language Processing*, 1.1(1996): 101-158.

Sun, Maosong and Changning Huang. "Word Segmentation and Part-of-Speech Tagging for Unrestricted Chinese Texts," A Tutorial at the 1996 International Conference on Chinese Computing (ICCC'96). Singapore, 1996.

Wang, Xiaolong. "Automatic Chinese Word Segmentation," in *Word Separating And Mutual Translation of Syllable and Character Strings*. Chapter 3, pages 31-48, Ph.D. Dissertation, Dept. of Computer Science and Engineering, Harbin Institute of Technology, 1989.

Webster, Jonathan J. and Chunyu Kit. "Tokenization as the Initial Phase in NLP," In the *Proceedings of the 14th International Conference on Computational Linguistics* (COLING'92): 1106-1110, Nantes, France, 1992.

Wu, Dekai. "Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora," *Computational Linguistics*, Vol. 23.3(1997): 377-403.

Appendix A: Hints for Theorem Proving

This appendix lists some indications on the proof of the four theorems presented in Section 4.

THEOREM 1: This theorem can be proved by crafting, for each pair of tokenization strategies of the longest tokenization family, a specific character string together with its tokenization dictionary, and by demonstrating that the two tokenization strategies under question produce different tokenization results for the character string.

Actually, Example 4 in Section 3, together with its symmetry which is formed by inverting both the character string to be tokenized and all the tokens in the tokenization dictionary, serves the purpose well. Let us denote $A=\{w_a/c_{k+1}/.../c_{2k}\}$, $B=\{c_1/.../c_{k-1}/w_b\}$, and $AB=\{c_1/.../c_{k-1}/w_b, w_a/c_{k+1}/.../c_{2k}\}$, and denote the outcome of relationship testing (1 for true and 0 for false) as *x/y*, where *x* is for Example 4 and *y* is for the symmetry. Table 7 gives the results for all the nine relationships under all the three possible situations, $n>m$, $n<m$, and $n=m$.

For instance, in Situation 1 where $n>m$, by taking $n=k$ and $k>m$, since $LR(n)=B$ and $LR(m)=A$ in Example 4 and $LR(n)=B$ and $LR(m)=B$ in the symmetry, the test results of the first relationship, which is $LR(n) \subseteq LR(m)$, are 0/1, which effectively demonstrate that the given relationship does not universally hold true. From Table 7, it is clear that the Example 4, together with its symmetry, works in almost all the situations except for the shaded three, which can be readily tackled with another similar example.

Relationship			**Situation 1**			**Situation 2**			**Situation 3**		
Left ⊆ Right			**n>m**			**n<m**			**n=m**		
Rel.	Left	Right	Test	Left	Right	Test	Left	Right	Test	Left	Right
	n	M		n=k	k>m		n<k	k=m		n<k	m<k
1	LR(n)	LR(m)	0/1	B/B	A/B	0/1	A/B	B/B			
2	RL(n)	LR(m)	0/1	B/B	A/B	1/0	B/A	B/B	0/0	B/A	A/B
3	DD(n)	LR(m)	0/1	B/B	A/B	0/0	AB/AB	B/B	0/0	AB/AB	A/B
4	LR(n)	RL(m)	1/0	B/B	B/A	0/1	A/B	B/B	0/0	A	B/A
5	RL(n)	RL(m)	1/0	B/B	B/A	1/0	B/A	B/B			
6	DD(n)	RL(m)	1/0	B/B	B/A	0/0	AB/AB	B/B	0/0	AB/AB	B/A
7	LR(n)	DD(m)	1/1	B/B	AB/AB	0/1	A/B	B/B			
8	RL(n)	DD(m)	1/1	B/B	AB/AB	1/0	B/A	B/B			
9	DD(n)	DD(m)	1/1	B/B	AB/AB	0/0	AB/AB	B/B			

Table 7. *Relationship testing for Example 4 and its symmetry.*

THEOREM 2: This theorem can also be proved by showing concrete (counter) examples. For the relationship between $LR(n)$ and PT, the example can be a character string of $2n+3$ different characters, $S=c_1...c_{n+1}c_{n+2}...c_{2n+2}c_{2n+3}$, and a tokenization dictionary which is made of these $2n+3$ different single characters plus the following four tokens: $w_a = c_1...c_{n+1}$, $w_b=c_1...c_{n+2}$, $w_c=c_{n+1}...c_{2n+3}$ *and* $w_d=c_{n+2}...c_{2n+2}$. In this case, there exist $F_D(n,S)=\{w_a w_d c_{2n+3}\}$ for any $n>1$, and $P_D(S)=\{w_b c_{n+3}...c_{2n+3}, c_1...c_n w_c\}$.

THEOREM 3: By definition.

THEOREM 4: It is straightforward to verify that $\bigcup_{n=1}^{\infty} F_D(n,S) \cup B_D(n,S) \neq C_D(S)$ holds for Example 3 in Section 2. Assume that the character string $S=c_1...c_n$, together with its tokenization D, has one $LR(n)$ tokenization, $W \in F_D(n,S)$, which is not a critical tokenization, $W \notin C_D(S)$. Denote $W=w_1...w_m$. By the definition of critical tokenization, there must exist i and j, $1 \le i < j \le m$, such that $w_i...w_j \in D$. This is in conflict with the definition of $LR(n)$.