Liwen Qiu
School of Library & Information Science
University of Western Ontario, London, Ont., Canada

Applicability of String Indexing to the Chinese Language with Special Reference to NEPHIS

Qiu, L.: **Applicability of string indexing to the Chinese language with special reference to NEPHIS.**
Int.Classif. 16(1989)No.2, p.89-94, 3 refs.

The applicability of the three types of string indexing, as defined by Tim Craven, to the Chinese language is investigated. It was found that KWIC and KWOC indexing cannot be used for the Chinese language. Term list input string indexing is directly applicable to Chinese, but it has not been used. The applicability of coded input strings varies with different systems: the application of PRECIS to the Chinese language requires much effort while the adoption of NEPHIS to the Chinese language requires fewer changes. Author

0. Introduction

Although string indexing has been used in English successfully for many years, it has not yet been applied to Chinese. There are two reasons why string indexing has not been applied to Chinese: first, the special characteristics of the Chinese language hinder the application of string indexing; and second, there is a general lack of research in this area. This paper describes an attempt to explore the applicability of string indexing to the Chinese language. It begins by examining the applicability to Chinese of three types of string indexing, namely, ordinary-language input strings, term list input strings, and coded input strings. The remainder of the paper focuses on NEPHIS, since it appears as the system most adaptable to the requirements of the Chinese language.

1. Ordinary-language Input Strings

The simplest and most common kind of string indexing software is designed to use expressions in ordinary language as input strings. These expressions may be unmodified titles of documents, descriptions composed by an indexer or hybrids of the two. Typical examples of this type of string indexing are the well-known KWIC (Key Word In Context), KWOC (Key Word Out of Context), and CYCLING.

The software for these ordinary-language systems distinguishes words as strings of characters set off by space; a word is picked out from the index string and compared with a stoplist or a golist (A stoplist is a list of terms which cannot be access terms; a golist is a list of terms which should be access terms). The access terms recognized are usually individual words, and these words are referred to as keywords. This method is language- or writing system-dependent and may not be directly applicable to non-alphabetic languages or languages with different writing conventions. For example, it is completely unworkable when confronted with the Chinese language.

The basic unit of the Chinese language is a character. Although a single character has a certain meaning, it is not a searchable term from the point of view of information retrieval. A searchable term consists of several characters, but there is no space between terms in a Chinese sentence. For example, the phrase in (1)

(1) 情报检索

means information retrieval. The first two characters stand for "information" and the latter two characters mean "retrieval", but the combination of the second and the third character is meaningless.

How could a computer be programmed to separate Chinese characters properly so that meaningful terms can be picked out to form index entries? One way of dealing with this problem would be to store a dictionary of terms so that each character in the text to be indexed can be scanned and matched against the dictionary. However, because the possible combinations of characters are so great in number, the size of the dictionary would be extremely reduced. For this reason, this method has been used only in experimental systems but not in real systems. The other problem with dictionary or artificial intelligence based systems is that the whole point of KWIC and KWOC is lost, namely, the quick and cheap production of indexes.

There is as yet no commercial Chinese version of KWIC or KWOC indexing. Thus, the first type of string indexing is not applicable to the Chinese language at present and its applicability will depend on the development of appropriate artificial intelligence or natural language processing technology.

2. Term List Input Strings

A second type of string indexing system is designed for input strings consisting of unconnected words, examples are SLIC and TABLEDEX. In English indexing systems, simple lists of keywords can be used quite successfully as input strings in permuted systems; and some index string generators of term-list input strings are quite similar to those for ordinary-language input strings. The key problem of separating Chinese characters properly by computer to form meaningful terms is solved completely in term-list input strings, since the input string consists of a group of already separated terms. Therefore term-list input string indexing is quite applicable to the Chinese language.

For example, in a SLIC (Selected Listing In Combination) system, the index strings produced for the terms listed in (2)

(2) EFFECTIVENESS
INDEXING
RETRIEVAL
THEORY

are (for English) given in (3)

(3) 1. EFFECTIVENESS : INDEXING : RETRIEVAL : THEORY
2. EFFECTIVENESS : INDEXING : THEORY
3. EFFECTIVENESS : RETRIEVAL : THEORY
4. EFFECTIVENESS : THEORY
5. INDEXING : RETRIEVAL : THEORY
6. INDEXING : THEORY
7. RETRIEVAL : THEORY
8. THEORY

The corresponding Chinese version has the same number of output entries but the terms in these eight entries will be different because Chinese characters are sorted according to "Pin Yin" (the Chinese phonetic system which can romanize Chinese characters). So, the output entries will be as in (4)

(4) 标引：检索：理论：效益
标引：检索：效益
标引：理论：效益
标引：效益
检索：理论
检索：效益
理论：效益
效益

The romanised forms are given in (5).

(5) 1. Biaoyin : Jiansuo : Lilun : Xiaoyi
2. Biaoyin : Jiansuo : Xiaoyi
3. Biaoyin : Lilun : Xiaoyi
4. Biaoyin : Xiaoyi
5. Jiansuo : Lilun : Xiaoyi
6. Jiansuo : Xiaoyi
7. Lilun : Xiaoyi
8. Xiaoyi

Although this type of string indexing is applicable to Chinese, there are as yet no Chinese indexes using string indexing with term-list input strings, probably because this type of string indexing is not as well-known as KWIC and KWOC indexing.

3. Coded Input Strings

A third type of index system uses codes added to title-like phrases or lists of terms to increase the indexer's control over the output. String indexes with coded input strings include Statement Indexing, automated library catalog display systems, PRECIS, POPSI, NEPHIS, LIPHIS, and NETPAD. The applicability of this type of string indexing to Chinese is different for different systems. I have chosen to study PRECIS and NEPHIS, because they offer interesting possibilities for comparison in that the output strings are fairly comparable, although the input processes are very different from one another.

4. PRECIS

Perhaps the most recognized string indexing system based on coding of input strings is PRECIS (PREserved Context Index System), developed for the British National Bibliography by Austin and others (1, p.32).

A PRECIS index string has three basic parts. The first two, the "lead" and the "qualifier", together form the heading; the lead is in boldface and is separated from the qualifier by a period-plus-space. The third part is a subheading, called the "display". This general pattern may be represented as

(6) Lead. Qualifier
Display

In PRECIS; a role operator is a code symbol which indicates the grammatical role or function of the term which follows it, and which regulates the order of terms in a string. Two PRECIS roles important for my analysis are the downwards connective "$V" and the upwards connective "$W". A connective coded $V is printed only when the string is being read in downwards order, i.e., when the term to which it is attached is in display position or in the lead. A connective coded $W is printed only when the string is being read in an upwards order; i.e., when the term to which it is attached is in the qualifier position (2, p.28).

Some simple PRECIS strings can be translated into Chinese directly, as shown in (7)

(7) SUBJECT: Feeding habits of common birds

input string:

(1) 鸟 $h 普通
birds $h common

(p) 习惯
habits

(2) 喂食
feeding

output entries:

鸟
BIRDS

普通，习惯，喂食
COMM BIRDS. Hab. Feeding

习惯，普通 鸟
HABITS. common birds

喂食
Feeding

喂食，习惯，普通 鸟
FEEDING. Habits, Common birds

But most English PRECIS strings cannot be translated into Chinese directly, because the syntax of PRECIS is based on the English syntax.

The biggest problem is prepositions, which are widely

used in PRECIS. For example, the preposition "of" is widely used with the upwards connective "$W". The corresponding Chinese character for "of" is as written in (8). But the word sequence is reversed, as in (9a) and (9b)

(8) 的

(9a)

儿童 的 营救
children of rescue
(rescue of children)

(9b)

书 的 订购
books of acquisition
(acquisition of books)

The preposition "by" is heavily used in English with the downwards connective "$V". The corresponding Chinese character of "by" is (10). But its different positions in a phrase give rise to different meanings as in (11)

(10) 被

(11a)

狗 被 救
dog by rescue
(dog was rescued)

(11b)

被 狗 救
by dog rescue
(rescued by dogs)

Another problem in Chinese is that many prepositions like "in", "on", and "to" are expressed by several characters which are separated by intervening words as the example in (12) shows.

(12)

在 教室 里
classroom
in
(in the classroom)

在 书桌 上
desk
on
(on the desk)

到 学校 去
school
to
(to the school)

Therefore, a very simple English PRECIS string containing prepositions can not be translated into Chinese using PRECIS codes. Consider (14), the entries which would result from the Chinese version of the familiar PRECIS string in (13)

(13) STRING: (1) children
(2) rescue $V by $W of
(3) dogs

(14) Chinese version:

a. 儿童
CHILDREN
营救 被 狗
Rescue by dogs

b. 营救. 儿童
RESCUE. Children
被 狗
By dogs

c. 狗
DOGS
营救 的 儿童
Rescue of children

Only (14b), the second Chinese entry, is correct. The first Chinese entry (14a) is grammatically wrong and the third Chinese entry (14c) is ambiguous in meaning between "children were rescued" and "children were rescuers".

If we change the connective for the preposition "of" to conform with Chinese syntax, we get the string in (15)

(15) 儿童 $V 的
children $V of
营救 $V 被
rescue $V by
狗
dog

The resulting entries are given in (16)

(16a) 儿童
的营救. 被 狗

(16b) 营救. 儿童
被狗

(16c) 狗
营救. 儿童

Now, the grammatical problem is solved, so both the first and second entries are correct. But (16c), the third entry, is still ambiguous.

There is no simple way to solve this kind of problem. In order to have a Chinese version of PRECIS, some changes in the existing connectives and rules must be made. The complicated rule system of PRECIS makes this task very difficult. One Chinese researcher has spent

many years trying to work out a Chinese version of PRECIS.

6. NEPHIS

The acronym NEPHIS stands for NEsted PHrase Indexing System (3), and it is the principle of nesting (or embedding) upon which the NEPHIS system is based. A NEPHIS input string is a phrase in ordinary language with added coding symbols. Only four different coding symbols are used: the left and right angular brackets (〈) (〉), the question mark (?), and the at sign (@).

Although the input string is in ordinary language, terms are separated by the added coding symbols. Thus, NEPHIS can be applied to Chinese without any additional mechanism to handle the problem of separating characters to form meaningful terms. In addition, the NEPHIS system is in practice much less language dependent than the PRECIS system. The four coding symbols are less related to specific language phenomena, as contrasted with PRECIS, in which $V and $W directly relate to some prepositions and serve their requirements in English.

7. The Applicability of the Four Coding Symbols of NEPHIS to Chinese.

7.1 The coding symbols "〈" and "〉"

The basic coding symbols of NEPHIS are "〈" and "〉",which are used to indicate that one phrase is nested in another. A number of phrases may be marked off by the indexer as being nested in a larger phrase. Moreover, the system of nesting is recursive; that is, the indexer may indicate that a phrase is nested within another phrase which is itself nested within a third phrase. This recursive nesting is used to improve the order of elements in one or more of the permutations. Each of the indexer-defined phrases in the input string becomes in turn the beginning of a permutation. The phrase is then examined by the program to see whether it is nested within a larger phrase in the input string. If so, the rest of the large phrase is appended to it by a period-plus-space, the point where the smaller phrase is omitted being indicated by a dash. The larger phrase is then itself examined to see whether it is nested within a phrase in the input string, the process being repeated until the entire input string has been covered, at which point the permutation is complete (3).

The coding symbols "〈" and "〉" also function effectively within Chinese strings, as in (17).

(17a) input string:

睡眠 研究者 的〈研究 成效〉
sleep researcher of rese. pro.
(Research productivity of sleep researchers)

(17b) output entries:

睡眠 研究者 的 研究 成效
sleep researcher of rese. pro.

研究 成效, 睡眠 研究者 的
rese. pro. sleep researchers of

Both entries are valid; that is, they are meaningful strings of Chinese. Because the word order in a sentence is different in English than in Chinese, the same string will have different ways of coding in the two languages; however, the NEPHIS codes work just as they should for Chinese. Another example is (18).

(18)
政府 资助 〈项目〉的〈出版物〉
gove. funded pro. of publi.
(publication of government-funded projects)

7.2 The Coding Symbol "@"

The symbol "@" is used at the beginning of a phrase to indicate that the permutation beginning with that phrase is not to be performed. Some English strings which need the symbol "@" may not need this symbol in their Chinese version since the word order of sentences are quite different in these two languages. For the same reason, there are some Chinese strings which need the symbol "@" while their English version does not need it. For example, the English input string in (19a) would be in Chinese (19b), which needs no symbol "@" to produce an output entry equivalent to the English one.

(19a) Equations for 〈@design of 〈retrieval system〉〉

(19b) 检索 系统 设计 〈方程〉
retr. system desi. equa.

In contrast, the English string (20a) does not need the "@" coding, while the Chinese version (20b) does.

(20a) Documents on 〈Canadian〈Agriculture〉〉

(20b) @关于〈加拿大〈农业〉〉的〈文献〉
Cana. agri. docu.
on

In short, the symbol "@" is applicable and very useful to Chinese strings.

7.3 The Forward-Reading Connective and the Backward-Reading Connective "?"

These two additional features were introduced in order to make the permutations read somewhat more

smoothly and intelligibly. Both involve the use of the question mark to alert the program that the word or words that follow are to be included in the permutation produced only under certain circumstances.

In the Chinese language, there are also some prepositions which should be omitted under certain circumstances in order to get a more comfortable reading. So, these two connectives are useful in Chinese strings. In addition, these connectives are not related to any specific words or sentence order; i.e., they are not language dependent. Therefore, they are applicable to Chinese strings. Of course, because of the difference of sentence structure between English and Chinese, the same string will be coded differently in the two languages in order to get output entries with comparable meanings. The Chinese input string for (21) will be (22a) and its output (22b).

(21) Opinions? of ⟨Users⟩? on
⟨Printed ⟨Subject ⟨index⟩⟩⟩

(22a) 用户 ? 对 ⟨书本式 ⟨主题
users to printed subject

索引⟩⟩? 的 ⟨意见⟩
index of opinions

(22b)

用户 对 书本式 主题 索引 的 意见
user to prin sub index of opin

书本式 主题 索引, 用户 的 意见
print sub index. user of opin

主题 索引, 书本式-, 用户 的 意见
sub index. prin-. user of opin

索引, 主题-, 书本式-, 用户 的 意见
index. sub-. prin-. user of opin

意见. 用户 对 书本式 主题 索引
opinions. user to prin sub inde

The string "rescue of children by dogs" which is difficult to implement in Chinese PRECIS is easy in NEPHIS with the proper use of the symbol "?" as in (23)

(23a) input string:

儿童 被 ? ⟨狗⟩ ⟨营救⟩
children by dos rescue

(23b) output entries:

儿童 被 狗 营救
Children by dog rescue

狗. 儿童 被 营救
Dog. Children by rescue

营救. 儿童 被 狗
Rescue. Children by dog

The three entries are all clear in meaning and natural to read.

A common usage of the backward-reading connective in English is in dealing with coordinating expressions such as "and" and "or". In Chinese, coordinating expressions have the same sentence structure as in English; therefore, they can be coded the same way in the two languages as in (24)

(24) 情报 科学 ? 与 ⟨运筹 研究 ? 与⟩
info sci ? & ⟨ope res ? &⟩
(information science and
operations research)

8. The Problem of Collocation

Collocation means the placing of similar index strings together and the separation of dissimilar index strings. The assumption is that searchers who have found one item of possible interest will be able to find a second similar item of possible interest more efficiently if it is close to the first. When NEPHIS is used in Chinese, however, collocation may be a problem. To illustrate the problem, let us start with the following string:

(25) Procedures for statistical
analysis of data.

@ 对⟨数据⟩进行⟨统计 分析⟩ 的 步骤
data stat anal of proc.
exert an action

The output strings will be

数据. 对-进行 统计 分析 的 步骤
data. stat anal of proc
exert an action

统计 分析. 对 数据 进行 的 步骤
stat anal data of proc.
exert an action

If we have several similar entries about "data", they will be collocated as follows:

(26) 数据 采集 技术
data collect. techni.
(Techniques of data
collection)

数据. 对-进行 筛选 的 重要性
data. fil. of impor.
exert an action
(The importance of data
filtration)

数据. 对-进行 统计 分析 的 步骤
data. stat ana of pro.
exert an action
(Proceudres for statistical
analysis of data)

数据 分类 技术
data classi. techni.
(Techniques of data
classification)

Under the access term "data" we do not get the desired collocation of "collection" and "classification" but the collocation of the expression meaning "exerting an action", which is a meaningless term from the point of view of information retrieval.

This collocation is a result of the fact that in Chinese, there is no difference in the term form of a verb and a noun. For instance, "evaluate" and "evaluation" have the same characters as in (27)

(27) 评价

Sometimes, an action is expressed by a phrase as in the above string (Procedures for statistical analysis of data), the first, the 4th and the 5th characters combined together mean exerting the action "statistical analysis" on the object "data". This is a very common sentence type, as illustrated in (28)

(28a) Methods of Computer Simulation of the Experimental Process.

对实验过程进行计算机模拟的方法
exp proc comp simu of meth
exert an action

(28b) Policy of Controlling Environmental Pollution.

对环境污染进行控制的政策
envi pollu con of policy
exert an action

All these strings will have the same collocation problem as mentioned above. Because it is not an occasional phenomenon, it may be desirable and worthwhile to adapt NEPHIS to deal with this problem.

The simplest way to approach it is to delete the characters meaning "exerting an action" when they are in the position immediately after the access term. This can be done by adding a further coding symbol, brackets ("(",")"), to these characters. When the program meets characters in brackets, it deletes them if they are in the position immediately following the access term. The collocation can be realized as in (29)

(29) 数据 采集 技术
data colle. tech.

数据 分类 技术
data classi. tech.

数据. 筛选 的 重要性
data. filtra. of import.

数据. 统计 分析 的 步骤
data. stat. anal. of proce.

This change may cause a little extra effort in programming and coding, but it is not very difficult. So NEPHIS will still meet the design criteria, that it be easy to program and easy for the indexer.

9. Conclusions

In this paper, I have investigated the applicability of NEPHIS to several different kinds of Chinese sentence patterns (although my investigation is by no means claimed to be exhaustive), and the result is very satisfactory. The four coding symbols turn out to be applicable to and very useful in Chinese strings. With the proper use of the four coding symbols, the output entry can be both smooth for reading and unambiguous in meaning.

Only one new symbol may be needed in order to improve collocation. (If the requirement for collocation is not very high, this new symbol may be unnecessary.) Although more indexing tests are needed to make this conclusion more reliable, I feel safe in claiming that NEPHIS is applicable to Chinese strings on the basis of my examination thus far.

In summary, KWIC and KWOC string indexing cannot be applied to the Chinese language. Although term list input string indexing can be used for Chinese, it has not been used. The application of PRECIS to the Chinese language requires much effort and the complicated rule system makes the problem worse. The result that NEPHIS can be easily applied to the Chinese language demonstrates that it is a less language dependent system and has wider applicability, because English and Chinese are very different languages. This system therefore merits greater attention and study for multilingual applications.

Acknowledgment:
The author would like to thank Dr.Gillian Michell and Dr.Timothy Craven for their great help in this research and in the writing up of the paper.

References

(1) Craven, T.C.: String Indexing. London, GB: Academic Press 1986.
(2) Ramsden, M.J.: PRECIS Workbook. Trowbridge and Esher: Redwood Burn Ltd 1981.
(3) Creaven, T.C.: NEPHIS: A Nested-Phrase Indexing System. J.ASIS (1977)No.1, p.107-114