TOKENIZATION AS THE INITIAL PHASE IN NLP

Jonathan J. Webster & Chunyu Kit
City Polytechnic of Hong Kong
83 Tat Chee Avenue, Kowloon, Hong Kong
E-mail: ctwebste@cphkvx.bitnet

ABSTRACT

In this paper, the authors address the significance and complexity of tokenization, the beginning step of NLP. Notions of word and token are discussed and defined from the viewpoints of lexicography and pragmatic implementation, respectively. Automatic segmentation of Chinese words is presented as an illustration of tokenization. Practical approaches to identification of compound tokens in English, such as idioms, phrasal verbs and fixed expressions, are developed.

1. Introduction: Tokenization in NLP

In NLP studies, it is conventional to concentrate on pure analysis or generation while taking the basic units, namely words, for granted. It is an obvious truth, however, that without these basic units clearly segregated, it is impossible to carry out any analysis or generation. But too little attention has so far been paid to the process, a kind of preprocessing in a sense, of identifying basic units to be processed. The simplicity of recognizing words in English, resulting from the existence of space marks as explicit delimiters, has most likely misled us into overlooking the complexity of distinguishing other units in English, such as idioms and fixed expressions, not to mention the difficulty in identifying words in other languages, like Chinese, resulting from the absence of delimiters.

In this paper, we define this preprocessing as tokenization. The first step in NLP is to identify tokens, or those basic units which need not be decomposed in a subsequent processing. The entity word is one kind of token for NLP, the most basic one. Our concern, however, is with using the computer to recognize those tokens without distinct delimiters, such as Chinese words, English idioms and fixed expressions.

So far, there exists very little research adopting the notion of tokenization we put forward here. Santos (1990) explored a pragmatic way to transfer English idioms and fixed expressions in the domain of machine translation. Linden et al (1990) focused on determining the idiomatic or non-idiomatic meaning of idioms. It is believed that, by taking idioms and fixed expressions as a kind of basic unit at the same level as words, tokenization should take on a more generalized and realistic significance making NLP and MT systems more robust and practical.

Before we can achieve the identification of such tokens by computational means, many fundamental issues need to be resolved. Among these the most important is clearly the definition of token.

2. Defining the entity word

There are a number of notions of what counts as a token in NLP. Different notions depend on different objectives (e.g. parsing, MT) and often different language backgrounds. To arrive at a definition of token, which is at once linguistically significant and methodologically useful, we propose to first address the issue of what is a word from a lexicographer's perspective.

Speaking as a lexicographer, J. McH. Sinclair proposes to define a lexical item as "a formal item (at least one morpheme long) whose pattern of occurrence can be described in terms of a uniquely ordered series of other lexical items occurring in its environment" (1966:412). For the lexicographer, it is simply a question of finding significant collocations.

Sinclair differentiates between what he calls 'casual' and 'significant' collocation. Casual collocation includes items which have no bearing on the node, and as Sinclair explains "may be accidental, reflecting the place, perhaps, where someone breaks into a committee meeting with the coffee; or they may include the magnificent images of some of our greatest poetry" (1966:418). The larger the corpus, the more casual collocates will be netted, but at the same time their significance will steadily decrease. While, on the other hand, "collocates typical of the item in question will impress their pattern more and more strongly until the pattern is broadly speaking complete and the evidence of further occurrence does not materially alter the pattern" (1966:418).

The lexicographer's approach to identifying words has significance for tokenization. By comparing observed collocation patterns of strings with stored patterns we can proceed to segment the text into words. Finding significant tokens depends on the ability to recognize patterns displaying significant collocation. Rather than simply relying on whether a string is bounded by delimiters on either side, segmentation into significant token relies on a kind of pattern recognition involving collocational patterns.

While suggesting that the search for lexical items begin with those units "which we widely call morphemes", Sinclair acknowledges those problems which are likely to complicate matters for the lexicographer:

(i) homographs; and

(ii) compounds or multi-morpheme items.

Both problems are likely also to affect, perhaps even frustrate attempts at automatic segmentation of strings into meaningful units. Homographs, for instance, possess multiple collocational patterns. It becomes a question of not simply finding a match, but evaluating between patterns to find the one with the best fit. Taking the complex homograph, *hand*, as an example, Sinclair writes, "Grammar is hardly any help at all, and the distinctions gained by a word-class division make little inroads on the complexity of this form. The lexicographer is forced to examine the *internal consistency* of the cluster" (1966:425). What one discovers is that some occurrences of *hand* are collocationally distinct from other occurrences of the same form. Sinclair cites two instances of *hand*. One having collocates like *marriage*, *daughter*, and *engagement*. The other with collocates which include words we associate with card games like *whist*, *rummy*, *ace*, and *flush*. Both groupings are quite distinct. As Sinclair puts it, "the chances of *whist* and *marriage* co-occuring are as poor as those of *archbishop* and *fish*." This suggests we are dealing with different lexical items. On the other hand, "groupings which shade into each other, even though opposite ends do not intercollocate, suggest one item with a wide range."

Polymorphemic items further complicate the situation. Richard Hudson, in his Word Grammar, treats compounds as a single word whose composition consists of a string of two words (1984:50). Citing the example of *furniture shop*, he explains, "the word *shop* provides the link between *furniture* and the rest of the sentence - it is because *shop* is a noun that the whole can occur where nouns can occur; if the whole were plural, the suffix would be added to *shop*, not to *furniture*; and semantically, a *furniture shop* is a kind of shop, and not a kind of furniture"(1984:87).

Hudson goes so far as to float the idea that we treat expressions like *London shop*, *expensive ship*, or even *soon left* (as in *John soon left*) in the same manner as single words consisting of a modifier followed by its head (1984:89). While he admits this reanalysis "may seem perverse", nevertheless he believes there are arguments in its favour. For one thing, the word-order rules for English would be simplified (i.e. a modifier follows its head unless both modifier and head are part of a word, then the modifier comes first). Also, his reanalysis would help to explain why premodifiers cannot themselves have postmodifiers.

Sinclair, on the other hand, would not regard a particular combination of words as a separate polymorphemic item unless its cluster cannot be predicted from the clusters of its components (l966:423). Thus, while some occurrences of *cold* + *feet* are regarded as a separate polymorphemic item, *cold* + *hands* would not be treated as such.

Sinclair fixes no limit on the size of a polymorphemic item. Moreover, contrary to a claim made by Hudson "that the parts of a word cannot be separated by other words which are not part of the same word" (1984:89), Sinclair argues that the components of a polymorphemic item may in fact be discontinuous. Sinclair cites examples like *you must cut your coat, I'm afraid, according to your cloth*, and from a Sunday newspaper, *put all his nuclear eggs in the West German basket*.

The possibility of achieving word recognition through mapping collocations in the text to stored collocational patterns suggests a common-sense, practical approach to tokenization and disambiguation.

3. Automatic word segmentation in Chinese NLP - An example of tokenization

Identification of words is still a perplexing problem in Chinese NLP. As with English words, particularly idioms and compounds, the source of difficulty has to do with the absence of delimiters between tokens.

3.1 Background

As we know, a Chinese character is computationally represented by an internal code. Words, however, each of which may consist of one or more characters, do not have any obvious indicators to mark their boundaries. Tokenization of Chinese words, including idioms and fixed expressions which are, of course, phrases containing words as their constituents but used as words, is generally regarded as another bottleneck following "Chinese character coding". It is known in formal terms as *automatic word segmentation* in China mainland and as *word identification* abroad. In recent years, it has become a very important topic in Chinese NLP. Without coding, it is impossible to input characters into computer. Without word identification, we cannot hope to achieve text processing.

This topic has been approached from two sides. On the theoretical side, researchers have sought an explicit specification of the entity word. The difficulty of word identification has resulted from a confusion of character, word and phrase in Chinese linguistics. Because the construction of words, phrases and sentences are so similar, some scholars even believe they are identical. In an attempt to bring this debate to some conclusion, a standard was introduced by the Chinese State Bureau of Standardization for word segmentation. The term *segmentation unit* was employed to refer to words, idioms, fixed expressions, terminology as long as two or three dozen characters and even any entities which can be treated as an undivided unit in a processing (Kit 1989). This term, as a prototype of token, indicates the appearance of tokenization notion in Chinese computing.

3.2 Basic methods

On the practical side, studies have concentrated on two aspects: 1) the implementation of mechanical segmentation with fundamental supports, such as the construction of a dictionary which permits quick and efficient access; 2) strategies for disambiguation.

At the outset, segmentation methods were invented one after another and seemed inexhaustible. But after systematic study, a structural model was finally built (Kit 1988; Kit et al 1989). In essence, word segmentation involves table-look-up and string matching such that character string of the input text is compared with entities in an existing word list, i.e., the dictionary. Every automatic segmenting method of this kind is proven to be decided by three factors, as shown below in the structural model ASM(d,a,m), in which

ASM stands for Automatic Segmenting Method;

$d \in \{+1,-1\}$, indicates the scanning directions in matching, scanning from left to right is forward and the opposite is backward, respectively;

$a \in \{+1,-1\}$, indicates character addition or omission in each round of string matching that finds a word, respectively;

$m \in \{+1,-1\}$, indicates the usage of maximum or minimum matching, respectively.

It is believed that all elemental methods are included in this model. Furthermore, it can be viewed as the ultimate model for methods of string matching of any elements, including methods for finding English idioms.

The minimum match methods are not appropriate for Chinese word segmentation because almost every Chinese character can be used as a token - a word or a single morpheme. By contrast, however, a maximum match method can obtain an identification rate as high as around 98 per cent, with an adequately large dictionary. The earliest and most influential implementation was the CWDS system (Liang 1984; Liu & Liang 1986), which processed a corpus of 200 million characters in practical use.

A segmenting strategy may integrate more than one basic method to achieve a special task, e.g., forward and backward scanning methods are often employed together to check segmentation ambiguities. Such has been proven an efficient approach, though not perfect.

3.3 Handling ambiguities

Most research today on Chinese word segmentation has shifted to handling ambiguities in order to achieve a higher identification rate. There are two types of ambiguities at the level of word segmentation:

Type I: In a sequence of Chinese characters $S = a_1...a_i b_1...b_j$, if $a_1...a_i$, $b_1...b_j$ and S are each a word, then there is *conjunctive ambiguity* in S. The segment S which is itself a word contains other words. It is also known as *multi-combinational ambiguity*.

Type II: In a sequence of Chinese characters $S = a...a_i b_1...b_j c_1...c_k$, if $a_1...a_i b_1...b_j$ and $b_1...b_j c_1...c_k$ are each a word, then S is an *overlapping ambiguous segment*, or in other words the segment S displays *disjunctive ambiguity*. The segment $b_1...b_j$ is known as an *overlap*, which is usually one character long.

3.3.1 Ambiguity checking

The first step toward resolving segmentation ambiguities is to find them. Bidirectional scanning is one simple and powerful method. Differences in segmentation resulting from the two methods reveal the presence of ambiguities. But there still remain many ambiguities not found using this method. An integral approach to checking segmentation ambiguities was developed as follows:

1. Find all possible words from the beginning of the string and record their end positions;

2. Redo step 1 from those end positions, rather than from the beginning, if there is any new end position equal to or exceeding previous greatest one, a type I or type II ambiguity, respectively, is found.

It is a very simple and efficient strategy for finding any ambiguity and prevent all unnecessary operations on false ambiguities (Kit 1988 & 1992).

3.3.2 Approaches to disambiguation

Normally, the disambiguation stage follows the mechanical segmentation and the ambiguity checking. Two distinct approaches to disambiguation are the knowledge-based and the statistical-based.

The former is to discriminate all ambiguities by means of a built-in knowledge base, including rules, which are applied to a series of similar ambiguities, and special case knowledge for particular cases of ambiguities (Liang 1984 & 1990; Ho et al 1991). A large number of uncertainties are settled in this way, but there is a side-affect: the rules may result in some mistakes that even a mechanical segmenting method can handle properly (Kit 1988). This may be partially due to the complexity of language, but a more sophisticated approach to organizing and applying knowledge is still needed.

As for the latter, deriving from corpus linguistics, general techniques in tagging are employed and some advances have been reported (Lai et al 1991). But the design of a comprehensive and efficient tagging system is still, however, a big problem. Besides, a relaxation approach, which skips the mechanical segmentation and entirely relies on calculation of possibility, is theoretically sound, but practically, its identification rate is just about 95% (Fan & Tsai 1988), lower than that of mechanical methods. An appropriate combination of relaxation and mechanical means is expected to achieve a better result.

4. English compound tokens in NLP

In previous sections, we concentrated on words, in both English and Chinese. In fact, there are still a large number of *compound* tokens that take *simple* tokens, like words, as their constituents. They are critical to many processes in NLP and machine translation so that their identification is of great significance.

4.1 Rationale for the notion of token

Concepts such as *word*, *collocation*, and *multi-morpheme item* are important to lexicographers and linguists; whereas the concept of token is specific to certain processes in NLP and MT. A token will not be broken down into smaller parts. In other words, for the purpose of computational processing, it can be treated as an **atom**.

There are many compound tokens, composed of a number of words, to be trans-ferred as a whole in MT. In syntactic analysis, if it is decided to treat them as indivisible units, with no care as to their inner structure, then they become tokens for syntactic analysis. Token, then is a **terminal node** in processing. This is the essence, and also the importance, of the concept of token.

4.2 Decomposition versus identification

There are mainly two opposing views on how one should deal with English idioms, which have been identified as compound tokens in our framework: one stresses the **decompositionality** of idioms into constituent parts (Wasow, Sag, and Nunberg 1983; Gazdar et al 1985; Stock 1989); another considers idioms as units of language as basic as word and **wholly non-compositional in meaning** (Wood 1986; Linden et al 1990; Santos 1990).

The concept of token may offer a possible solution to this debate. To what degree a linguistic unit requires decomposition will depend on the nature of the task to be performed. In the case of lexical transfer in MT, there is no need to decompose an idiom into its constituent parts. However as noted below in our discussion of idioms and fixed expressions in discontinuous co-occurrence, structural analysis is sometimes necessary. In either case, priority must be given to the recognition of compound tokens. The whole must first be ascertained before one can even consider what are its constituents.

5. Tokenization and lexical information retrieval

There are a number of approaches to recognizing compound tokens. In this section we discuss two in particular. In the first, recognition is achieved by means of accessing lexical knowledge represented as a network of associations. The second adopts an approach combining table-look-up matching and knowledge processing.

5.1 Lexical information retrieval as a basis for token recognition

As noted above, the lexicographer's notion of word corresponds closely to the notion of token we have adopted here. We saw that what the lexicographer takes to be a word is an entity for which there exists some distinctive and significant collocation pattern. This bidirectional association between a word and its companions is itself evidence of that word's integrity and offers insight into its interpretation. The lexicographer's discovery procedure offers a useful model for achieving token identification. We are proposing to train a neural network to recognize tokens on the basis of their companion relations. Once the training process is completed, the neural network will be enabled to perform the tasks of tokenization and disambiguation by matching input with learned patterns of companion relations.

The network might also have to include information about other kinds of relations as well. The basic premise of Richard Hudson's Word Grammar is that the entity word can be realized as part of a system or network of relations. Entities in the lexicon, he explains, include words and their parts, their models, their companions, their referents and their contexts of use. Lexemes are emic units joined systematically to one another along vertical and horizontal dimensions. Every entity in the lexicon is at once a whole and an instance. It is the composite realization of its parts as well as the realization of some model. Along this vertical dimension, information flows from the more general to the more specific. The horizontal dimension, on the other hand, includes the lexical constraints imposed by heads on modifiers as well as vice versa. These Hudson refers to as an entity's companion relations. Such are the relations between collocates. Hudson's network approach accounts for the various realizations of entities as they occur in context in terms of the connections drawn between an entity and its referent(s), utterance-event(s), and companion(s). In a previous implementation of Hudson's network approach, we represented each lexical entry by means of a frame whose slots corresponded to Hudson's five relations (Webster,1987).

5.2 Table-look-up matching

The simplest approach to identification of compound tokens is obviously table-look-up matching. Admittedly, it presumes that a list of sample tokens in sufficient number already exists. The basic steps of this approach are, first, tokenize each single word, then continue matching to find whether there are any compound tokens among these single words. Such an approach is very similar to the basic method of automatic segmentation of Chinese words. This method can recognize English idioms and other compound tokens whose constituents are *continuous*, but has no ability to handle ambiguities and catch variant forms of idioms and fixed expressions in discontinuous co-occurrence.

5.3 Generalized Table-look-up

This is an adjusted table-look-up method designed to deal with idioms and fixed expressions in *discontinuous co-occurrence*, e.g., *keep* [NP] *in mind* in which *keep* plus *in mind* constitutes a fixed expression and *in mind*, a prepositional phrase, is merely part of a bigger token. Between these two parts, there is a noun phrase which is usually not too long. If it is long, we have a variant form of it, i.e., *keep in mind* [NP]. Of course, the [NP] can be substituted with a [Subclause].

In order to identify a compound token like this, we have to determine the NP and Subclause. Operations of this type need to, in part, make use of syntactic analysis. Thus, *partial parsing* needs to be incorporated into the table-look-up. Notice that the parsing may take every word as its token in order to find compound tokens. From this, one can see the importance of structural analysis to the identification of compound tokens.

Besides structural analysis, knowledge about compound tokens, such as where the [NP] and [Subclause] should be put in the discontinuous token *keep ... in mind*, is also required. Discontinuous idioms, phrasal verbs such as *figure out* [NP] and *figure* [it] *out*, and fixed expressions, all have to be processed with the aid of knowledge. By now it is clear that the generalized table-look-up is an approach combining parsing and knowledge processing. With adequate knowledge about discontinuous compound tokens, it may prove effective in their identification.

6. Conclusion

The notion of token must first be defined before computational processing can proceed. Obviously there is more to the issue than simply identifying strings delimited on both sides by spaces or punctuation. We have considered what constitutes a token from two perspectives: one from the lexicographer's experience with identifying words, the second from the experience of researchers in the area of Chinese NLP. From the work on automatic word segmentation in Chinese NLP, we have noted some valuable lessons which can be applied to the recognition of idioms and other fixed-expressions in English. The lexicographer's discovery procedures, informed with the knowledge of lexical relations implemented either as a neural network or in lexical frames, also provide a useful model for the construction of a practical, knowledge-based approach to tokenization and disambiguation.

Main references

[1] Fan, C. K., and Tsai, W. H. 1988. Automatic word identification in Chinese sentences by the relaxation technique, *Computer Processing of Chinese & Oriental Languages*, V.4, No.1.

[2] Gazdar, G., Klein, E., Pullum, G., and Sag, I. 1985. *Generalized Phrase Structure Grammar.* Cambridge, Mass.: Harvard University Press

[3] He, K., Xu, H., and Sun, B. 1991. Expert system for automatic word segmentation of written Chinese. *Proceedings of 1991 International Conference on Computer Processing of Chinese and Oriental Languages*, Taiwan.

[4] Hudson, R.A. 1984. *Word Grammar.* Oxford: Basil Blackwell.

[5] Kit(=Jie), C. 1988. *Methods of Chinese automatic word segmentation and the design and implementation of a practical system*. Master's Thesis, Graduate School, Chinese Academy of Social Sciences, Beijing.

[6] Kit(=Jie), C., Liu, Y., and Liang, N. 1989. On methods of Chinese automatic word segmentation. *Journal of Chinese Information Processing*, Vol.3, No.1.

[8] Kit(=Jie), C. 1989. Some key issues on the Contemporary Chinese Language Word Segmentation Standard Used for Information Processing, *Proceedings of 1989 International Symposium on Standardization of Chinese Information Processing*, Beijing.

[9] Kit, C. 1992. Practical techniques of Chinese automatic word segmentation in the applied system CASS, *Proceedings of PAN-ASIATIC LINGUISTICS-92*, Bangkok.

[10] Kramsky, Jiri. 1969. *The Word as a Linguistic Unit.* The Hague: Mouton.

[11] Lai, T., Lun, S., Sun, C., and Sun, M. 1991. A maximal match Chinese text segmentation algorithm using mainly tags for resolution of ambiguities, *Proceedings of ROCLING IV*, Taiwan.

[12] Liang, N. 1984. Chinese automatic word segmentation system CDWS, *Journal of Beijing University of Aeronautics and Astronautics*, 1984, No.4.

[13] Liang, N. 1990. The knowledge for Chinese word segmentation, *Journal of Chinese Information Processing*, Vol. 4, No. 2.

[14] Linden, E. van der, and Kraaij, W. 1990. Ambiguity resolution and the retrieval of idioms: two approaches. *Proceedings of COLING-90.* Helsinki, Finland.

[15] Liu, Y., and Liang, N. 1986. Basic engineering for Chinese processing - modern Chinese word frequency count, *Journal of Chinese Information Processing*, Vol.1, No.1.

[16] Santos, D. 1990. Lexical gaps and idioms in machine translation, *Proceedings of COLING-90*, Helsinki, Finland.

[17] Sinclair, John McH. 1966. Beginning the study of lexis (1966:410-429), in Bazell, et al. (eds.) In *Memory of J R Firth.* Longmans: London.

[18] Stock, O. 1989. Parsing with flexibility, dynamic strategies and idioms in mind. *Journal of Computational Linguistics*, Vol. 15, No. 1.

[19] Wasow, T., Sag, I., and Nunberg, G. 1983. Idioms: an interim report. In S. Hattori and K. Inoue (eds.) *Proceedings of the 13th international congress of linguistics*, Tokyo, Japan.

[20] Webster, Jonathan J. 1987. A computational model for representing WORD knowledge (1987:432-442) in the *Thirteenth LACUS Forum 1986*, Chicago, Illinois: LINGUISTIC ASSOCIATION OF CANADA AND THE US.

[21] Wood, M. McGee. 1986. *A Definition of Idiom.* Master's Thesis, University of Manchester (1981). Reproduced by the Indiana University Linguistics Club.