Detecting Sequential Delimiters

Jordan Genoff

Abstract : *An approach to identification of sub-sequences with properties of delimiter in a sequence with a priori unknown structure and semantics is presented. The idea is to apply some grammar inferring technique to reveal the structure of the sequence and, since each of the grammar elements represents certain construct, respectively one or more sub-sequences, to calculate a quantitative assessment for the hypothesis that this element is a delimiter. SEQUITUR [1] is used to deduce the sequential structure and a specific kind of probabilistic analysis gives the numerical rating of the hypotheses.*

Key words: *sequence analysis, sequential structure, delimiter.*

INTRODUCTION

An implicitly adopted empirical definition of 'delimiter' in the information processing context should look like this : "Delimiter is a piece of data that separates two contiguous pieces of data or marks the beginning or end of a piece of data".

The motivation to develop a technique for identification of delimiting sub-sequences in a sequence is a direct result from the above definition. First, delimiters (if there are any) are the most reliable places where the sequence may be broken if segmentation is required. Second, delimiters (if identified) are the most semantically reliable parts of a sequence if its content is incomprehensible.

Thus, an attempt to detect delimiters in a sequence can be a very smart first step in its analysis, especially when the goal is to disclose semantics. This paper describes a method for delimiter identification which is based on the idea to apply a grammar inferring technique to reveal the structure of the sequence and, since each of the grammar elements represents certain construct, respectively one or more sub-sequences, to calculate a quantitative assessment for the hypothesis that this element is a delimiter.

As a sequential structure inferring tool is chosen SEQUITUR – an algorithm widely known for its simplicity of operation and its very suitable (for the purpose of the presented method) grammatical representation of sequence's structure. As an assessment measure is developed a specific probabilistic-like analytical procedure giving the degree of partial correlation between the elements of a SEQUITUR generated grammar.

SEQUITUR

Most brilliant innovations are quite simple and this applies to SEQUITUR too. The idea was initially introduced in [2] as a source modeling approach for compression purposes and was finally refined and fully investigated by the same authors in [1] as a general method for deducing sequential structure and representing it as a hierarchical grammar.

SEQUITUR processes an input sequence of discrete symbols and produces a set of rules that represent the sequence as a context-free grammar of a very restricted kind – it is capable of generating one and only one string and it is the original sequence. As with any grammar, the rules are non-terminal symbols. The grammar is hierarchical because each rule contains concatenated terminal symbols and other rules that do not contain directly or indirectly the given rule – there are no recurrent dependencies between rules. The top level rule represents the whole sequence and only it. Every other rule represents any at least twice repeated subsequence and only it.

The algorithm builds the grammar while processing the input sequence from its beginning forwards, symbol by symbol with no back steps. Before the start there is only one empty top level rule S in the grammar. At each step a symbol from the input is appended to the end of S. Then a repetitive procedure is performed in order to test if the following two properties are satisfied for every rule in the grammar :

P1 : Digram uniqueness property : No pair of adjacent symbols (digram) appears more than once everywhere in the grammar – requires that every digram in the grammar be unique. If not satisfied, the digram that violates the property produces a new rule and this rule replaces all occurrences of the digram in the grammar.

P2 : Rule utility property : Every rule is used more than once – ensures that each rule in the grammar is useful. If not satisfied, the rule that violates the property is removed from the grammar and all its occurrences are replaced with its content.

An important fact about SEQUITUR is that in general there could exist more than one grammars built according to these rules for one and the same input sequence.

Table 1 shows the operation of SEQUITUR on the input sequence *abcdbcabcd*.

step	sequence	repeat	grammar	remarks
1	a	0	S => a	
2	ab	0	S => ab	
3	abc	0	S => abc	
4	abcd	0	S => abcd	
5	abcdb	0	S => abcdb	
6	abcdbc	0	S => abcdbc	bc violates P1 , A => bc
		1	S => aAdA A => bc	
7	abcdbca	0	S => aAdAa A => bc	
8	abcdbcab	0	S => aAdAab A => bc	
9	abcdbcabc	0	S => aAdAabc A => bc	bc is the rule A
		0	S => aAdAaA A => bc	aA violates P1 , B => aA
		1	S => BdAB A => bc B => aA	
10	abcdbcabcd	0	S => BdABd A => bc B => aA	Bd violates P1 , C => Bd
		1	S => CAC A => bc B => aA C => Bd	B violates P2
		2	S => CAC A => bc C => aAd	

Table 1. Processing steps and refinement repeats for the sequence *abcdbcabcd*

The final result for the sequence abcdbcabcd is

sequence	grammar	expanded rules
abcdbcabcd	S => CAC	abcdbcabcd
	A => bc	bc
	C => aAd	abcd

Table 2. Final resulting grammar for *abcdbcabcd*

It is easy to see that even for such a short sequence SEQUITOR succeeds to identify the comprising "pieces of data" that tend to show strong correlation between symbols comprising them.

GRAMMAR ANALYSIS

The goal is to decide which of the identified sub-sequences satisfy the requirements to be considered a delimiter. These requirements are of semantic nature and hence they are implied by certain assumptions. A precise description of these assumptions is able to give a precise definition of the requirements.

Here is presented a possible set of assumptions derived from the definition of 'delimiter' (for reasons of clarity 'piece of data' will be denoted as 'word') :

A1 : The inter-symbol correlations inside words are stronger than outside them. This assumption restricts the scope to words with a steady structure and excludes the case when delimiter delimits pieces of data with highly random nature.

A2 : For a relatively long rule the probability that it comprises of a whole word and a delimiter or a part of a word and a delimiter is higher than the probability that it comprises of two whole words or two parts of words and a delimiter between them. A2 follows from A1 and the properties of the grammars created by SEQUITUR.

A3 : A delimiter strives to occupy the marginal positions in the rules. A3 follows from A2.

In order to make the following discussion more clear the term "quasi-expansion" is introduced. When non-terminal grammar symbol (rule) s is quasi-expanded regarding non-terminal grammar symbol d, the expansion is performed fully for all rules comprising s at any level of expansion, untill terminal symbols are reached, except for d whose occurrences are preserved at any level of expansion. Thus the final expanded sequence equivalent to s consists of only terminal symbols and d (if at all). For the example grammar from Table 2, the quasi-expanded regarding A form of S is S => aAdAaAd.

The above assumptions are transformed in the following requirement :

R1 : The grammar symbol d (terminal or non-terminal), being a delimiter and present in the non-terminal grammar symbol s, will occupy the most left or/and the most right position in the quasi-expanded regarding d form of s.

The verification of this requirement for a grammar symbol (terminal or non-terminal) with respect to all non-terminal grammar symbols is algorithmically implementable and allows to compute a normalized numeric value which is proportional to the number of times the requirement is satisfied for the symbol. By this value the symbol may be rated as a possible candidate to be a delimiter.

A suitable formal rating factor R_d for the hypothesis that the grammar symbol d is a delimiter is calculated as

$$R_d = \frac{\sum_{\forall s \in G, s \neq d} (f_{ds}^{(l)} + f_{ds}^{(r)})}{\sum_{\forall s \in G, s \neq d} (f_{ds}^{(l)} + f_{ds}^{(m)} + f_{ds}^{(r)})} \qquad (1)$$

where G is the grammar, and $f_{ds}^{(l)}$, $f_{ds}^{(r)}$, $f_{ds}^{(m)}$ are numerical flags indicating if d is present ($f = 1$) or not present ($f = 0$) at the leftmost ($f_{ds}^{(l)}$), the rightmost ($f_{ds}^{(r)}$) or any middle ($f_{ds}^{(m)}$) position in the quasi-expanded regarding d form of s.

The top-level rule S is not rated because it must not be found in any other rule and thus the sum in its denominator will be zero. It is obvious that for no other grammar symbol the denominator will be zero.

Here is a sample case study :

input sequence :	petarxpletxpletexprezxpetxprataxpreplitav	
grammar symbol	expanded symbol	R_d rating
S => p1ar22e3z415a63pli6v	petarxpletxpletexprezxpetxprataxpreplitav	
1 => et	et	0.50
2 => 4l1	xplet	0.00
3 => 5e	xpre	0.00
4 => xp	xp	0.75
5 => 4r	xpr	0.50
6 => ta	ta	0.00
a	a	0.50
e	e	0.50
i	i	0.00
l	l	0.00
p	p	0.40
r	r	0.33
t	t	0.75
x	x	0.80
v	v	1.00
z	z	0.00

Table 3. A case study on the R_d performance

Table 3 shows unambiguous results according to the rating factor. The most probable candidate is 'v', which (as it seems from the original sequence) is a terminating delimiter. The next probable candidate is 'x', which (as it again seems from the original sequence) is a typical "general purpose" delimiter. Strings 'xp' and 't' are suspicious, too, and it's a matter of consideration what to decide about them.

EXPERIMENTAL INVESTIGATION

Table 3 shows a very nice case of delimiter and that is why the result is so unambiguous. The general case is much more complex. Here are listed some of the influencing factors that lead to many various cases for investigation :

F1 : Whether a delimiter is one symbol long or is more than one symbol long.

F2 : Whether the delimiter is only one or there are more than one distinct.

F3 : If delimiters are more than one, do they have one and the same probabilistic nature or they do not.

F4 : Does a delimiter consist of a subsequence that is present in the words or it does not.

F5 : If a delimiter consists of a subsequence that is present in the words, of what probabilistic nature is the subsequence.

It is clear that a very large number of opportunities to reveal the properties of this technique exist. They are so many, that a kind of heuristic selection among them must be considered. The goal is to postulate such a quantitative rating criteria that shall perform equally well in any case.

CONCLUSIONS AND FUTURE WORK

The method suggested here is not able to take decisions and to choose delimiters by itself alone. It is a tool, which gives the starting piece of information for discussion about which of the most probable candidates are the real delimiters in an unknown sequence.

Though a very exhaustive search was conducted, no existing sources of discussion about delimiter detection were found. From this point of view the paper presents an original problem with an original attempt to its solution.

Future work will be directed to two major goals :

G1 : Discovering the reasons for poor performance in the cases where it is observed. Two groups of reasons are possible : weakness of requirement R1 and/or grammar inadequacy. R1 reconsideration is very possible as a solution to the weakness problem.

G2 : Investigating and exploiting the grammar polymorphism as a solution to the grammar inadequacy problem.

A reliable delimiter detection technique is expected to show interesting results when applied in biological sequence analysis.

REFERENCES

[1] Nevill-Manning, C.G., I.H. Witten, "Identifying Hierarchical Structure in Sequences: A linear-time algorithm", Journal of Artificial Intelligence Research, no. 7, pp. 67-82, 1997.

[2] Nevill-Manning, C.G., Witten, I.H., Maulsby, D.L, "Compression by induction of hierarchical grammars", Proc. Data Compression Conference, Los Alamitos, CA: IEEE Press. 244-253, 1994.

ABOUT THE AUTHOR

Ass.Prof. Jordan Genoff, Department of Computer Systems, Technical University of Sofia at Plovdiv, Phone: +359 32 659 729, E-mail: jgenoff@tu-plovdiv.bg.