Natural Languages Analysis in Machine Translation (MT) based on the STCG (STRING-TREE CORRESPONDENCE GRAMMAR)

Tang Enya Kong, Zaharin Yusoff
Unit Terjemahan Melalui Komputer
Pusat Pengajian Sains Komputer
Universiti Sains Malaysia
11800 Minden, Pulau Pinang, Malaysia.
[e-mail: enyakong@cs.usm.MY and zarin@cs.usm.MY]

0. Abstract

The String-Tree Correspondence Grammar (STCG) [1] is a grammar formalism for defining:

- a set of strings (a language),
- a set of trees (valid representation/interpretation structures),
- the mapping between the two (to be interpreted for analysis & generation).

The formalism is argued to be a totally declarative grammar formalism that can associate, to strings in a language, arbitrary tree structures as desired by the grammar writer to be the linguistic representation structures of the strings. More importantly is the facility to specify the correspondence between the string and the associated tree in a very natural manner. These features are very much desired in grammar writing, in particular for the treatment of certain linguistic phenomena which are 'non-standard', namely featurisation, lexicalisation and crossed dependencies [2,3]. Furthermore, a grammar written in this way naturally inherits the desired property of bi-directionality (in fact non-directionality [4]) such that the same grammar can be interpreted for both analysis and generation.

In this paper, we investigate the properties of the STCG for interpretation towards analysis (as is understood within the context of Machine Translation (MT)). Other than using STCG grammars as specifications for the automatic generation of analysis programs in the Specialised Languages for Linguistic Programming (SLLPs) of MT systems (a study reported in [5,6]), the work also centres around the specification of a general analyser/parser for the STCG. The proposed STCG analyser is capable of mimicking some very useful features in various context-free parsing techniques. One such feature is the use of charts in tabular parsing algorithms, as exemplified in Earley's Algorithm [7], which is very helpful in avoiding redundancies that may otherwise result in a combinatorial explosion. Another is the compact way of representing possible parse trees for ambiguous sentences, such as the one seen in [8]. Though not reported in this paper, we note that the proposed analyser also provide a natural way for handling the kind of awkward phenomena mentioned above (namely lexicalisation, featurisation, and worst of all, crossed dependencies) while at the same time retaining much of the efficiency of standard context-free parsing algorithms (a study reported in [2,3]).

1. The STCG Formalism

The String-Tree Correspondence Grammar is a declarative grammar formalism that can be used to describe the correspondence between strings of terms and trees. In particular, linguistic rules are written with utterances as the string of terms (henceforth STRING) and the corresponding representative linguistic structures as the tree (henceforth TREE). Figure 1 gives an indication of a full STCG rule. The structure of the TREE is totally specified by the linguist and is not constrained by any application of rules (as in the case for the parse tree in the classical context free grammar). In a rule, the main correspondence is first declared: in the example, the STRING #NP1.v.#NP2.part (with #NP1 and #NP2 being *string variables*, ie. variables which are instantiable to strings of terms) is set to correspond to the TREE with root node S (where $A and $B are *forest variables*, ie. variables that can be instantiated to lists of subtrees). The main-corr(espondence) is followed by a declaration of subcorrespondences (on the right hand

side) between substrings of the STRING and subtrees of the TREE, each of which possibly having a list of references (rule names). For example, the sub-corr(espondence) between the substring #NP1 and the subtree rooted at the node NP1 refers to the rules RNP..., the latter being other rules in the grammar. This *reference* is a mechanism by means of which the string and forest variables mentioned earlier are fully instantiated via an operation called *identification* [9,10] resulting in a correspondence between explicit strings of terms and and trees, both without variables. In actual fact, the main-corr as well as the sub-corr specified in the rule are formally recorded in terms of a Structured String Tree Correspondence (SSTC) transparent to the linguist [11] as illustrated in figure 2, where a given correspondence may be non-projective (eg. with discontinuous constituents) as is the case for the node *v(part)* in the example. Note also that the particle is chosen (by the linguist) to be represented as a collection of features in the node *v* - a case of featurisation.

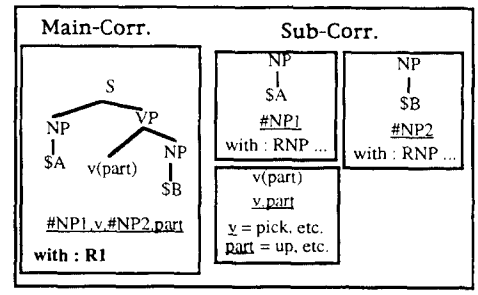


Figure 1.

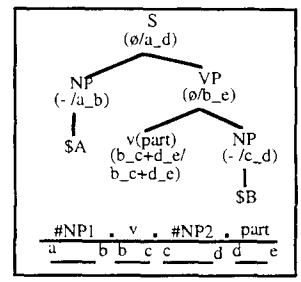


Figure 2.

In very simple terms, a string to tree correspondence in the STCG can be viewed as analogous to the mathematical definition of a relation between integer numbers as in the example given on the right. Here, a relation (in this case a function) f is defined in terms of finer subrelations according to the subdomains.

$$f(x) = \begin{cases} x^2 - 3 & x \leq 3 \\ x^3 + 5 & 3 < x < 5 \\ x & 5 \leq X \end{cases}$$

A set of STCG rules form a grammar, some of which are axiom rules (ie. start rules or rules containing axiom trees, as in the axiom or the start symbol S in the classical context free grammar). With the semantics of the rules being as indicated above, a grammar thus defines a language of strings, a language of representation trees, and the correspondence between elements of the two languages/sets. It is this set of string-tree correspondences that can be interpreted for both analysis and generation.

2. Natural Languages Analysis in MT Based on the STCG

Initially, the STCG was designed to serve as a specification language for writing grammars in MT such that the specifications written in the STCG grammar formalism can then be coded (manually) into the linguistic programs for analysis and generation written in the SLLPs of integrated MT systems. Some substantial work have also been carried out to automate this process, namely towards the automatic generation of analysis programs in the MT systems ARIANE [12] and JEMAH [13] from grammars written in the STCG formalism (see for example [5,6]). However, due to certain limitations in the existing SLLPs for the realisation of a proper implementation of a STCG analyser (as discussed in [2]), we have decided instead to look into the design of an analyser which can directly interpret the STCG grammar.

2.1. The Fundamental Design of the STCG Analyser

As we have seen above, a STCG grammar actually defines a set of SSTCs in a way quite similar to the definition of a mathematical function. In evaluating a mathematical function, if the function is defined in terms of other sub-functions then it can only be completely evaluated after all its sub-functions have been evaluated and return with the appropriate values. We can view the STCG analysis process in the same manner where, by taking the input string/sentence as their STRING, the set of explicit SSTCs defined by the axiom rules of a grammar are constructed based on the resultant sub-SSTCs defined by the reference rules of these axiom rules. Since the

reference rules of the axiom rules may in turn refer to other rules, they may also return the completed SSTCs only after their respective reference rules have been completed. This reference process will terminate when all remaining sub-SSTCs evaluated are defined by subcorrespondences which do not refer to any other rule, namely the 'lexical-SSTCs', which must match with the input words (the non-lexical SSTCs are called 'phrasal-SSTCs'). We illustrate this in the following analysis of the input string "He picks the ball up" with respect to a grammar consisting of rule R1 given in figure 1 and rules RNP1, RNP3 given in figure 3. The rule R1 is given as an axiom rule.

The analysis process begins with the evaluation of the general SSTC defined by the axiom rule R1, which in turn leads to the evaluation of two other sub-SSTCs defined by the reference rules RNP1, RNP3 as illustrated in figure 4.

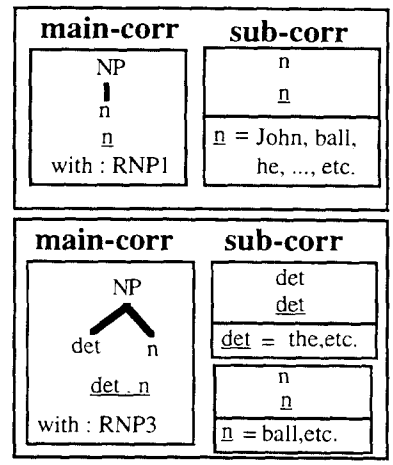


Figure 3.

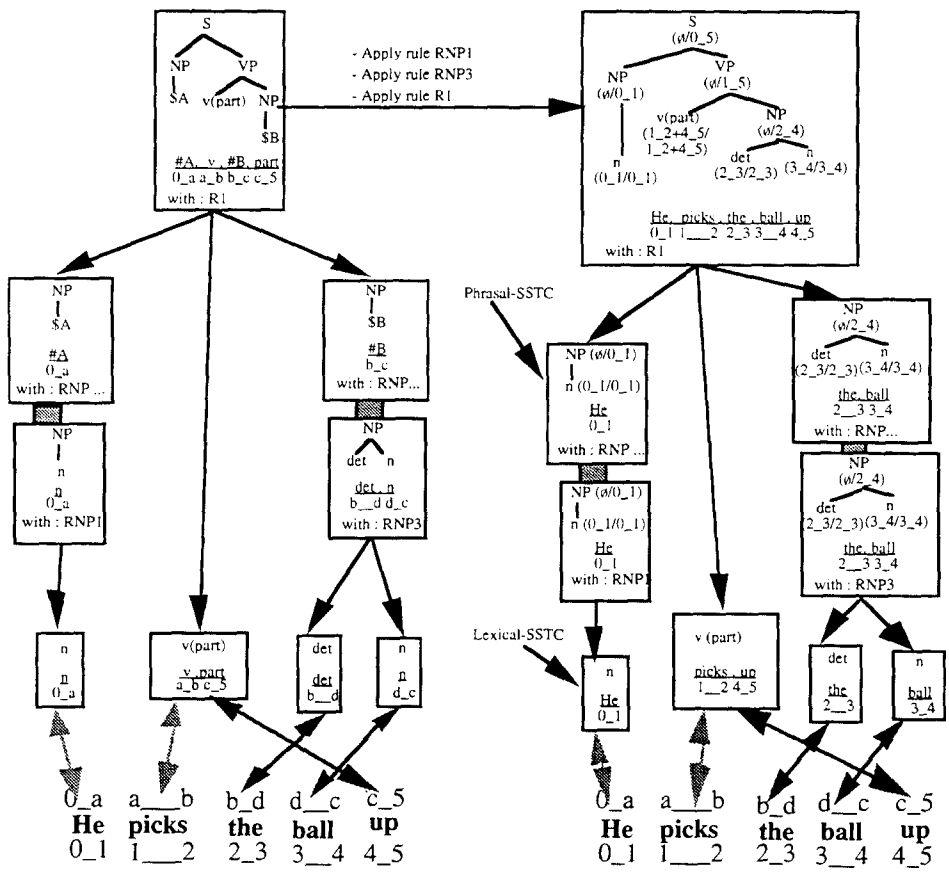


Figure 4.

In the diagram above (on the left), the analysis process expands the SSTC defined by the axiom rule into a string of sub-SSTCs, which is further expanded into another string of sub-SSTCs until it cannot be expanded any further, which is when the string of sub-SSTCs consists only of lexical-SSTCs. The string of lexical-SSTCs is then matched with the words in the input string. Note that the matching need not be in a projective manner, as can be seen in this particular example, where the lexical-SSTCs are matched to the words in the input string in a crossed serial manner - a case of crossed dependencies. In order to keep track of such non-

projective correspondences, we introduce the use of index variables to record the interval corresponding to each symbol appearing in the STRING (as illustrated on the right).

In [2], we proposed a design of the STCG analysis algorithm which is capable of mimicking some very useful features in various context-free parsing techniques. One such feature is the use of charts in tabular parsing algorithms, as exemplified in Earley's Algorithm [7], which is very helpful in avoiding redundancies that may otherwise result in a combinatorial explosion. Another is the representation of shared forest in term of a STCG grammar rules which is in fact following the approach adopted in [8] as illustrated in the next section.

2.2 Multiple Results of analysis for ambiguous input sentence

The example sentence given above is unambiguous, and thus corresponds to only a single representation tree. However, natural language grammars are known to be in the class of highly ambiguous grammars, and as such, there may be numerous representation trees generated for a single sentence in the language described. Instead of storing each representation tree separately in the set of SSTCs defining the correspondences between the given sentence and all its possible representation trees , we should try to represent all these in a space-efficient manner. In the figure given below, we present a compact way of representing a set of SSTCs corresponds to an ambiguous sentence by means of an AND-OR graph of rules - similar to the technique used by [8]. For example, the two SSTCs:

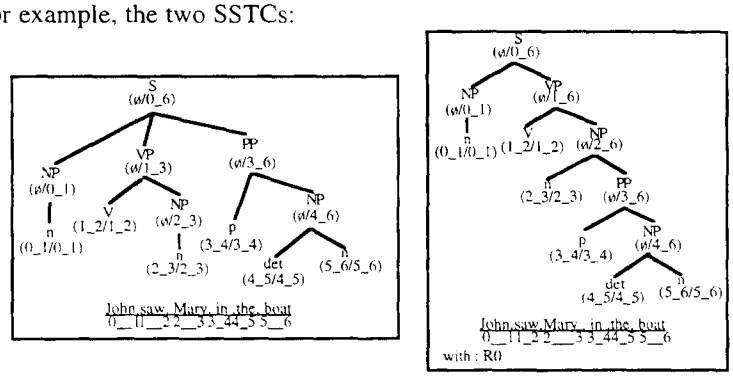


Figure 5:Two linguistic representations of the sentence *John saw Mary in the boat.*

can be factorised into an AND-OR graph of rules R2, R3, RNP5, RPP (given below) and rules RNP1, RNP3 (given in figure 3) in the following manner:

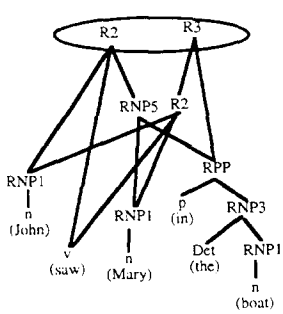
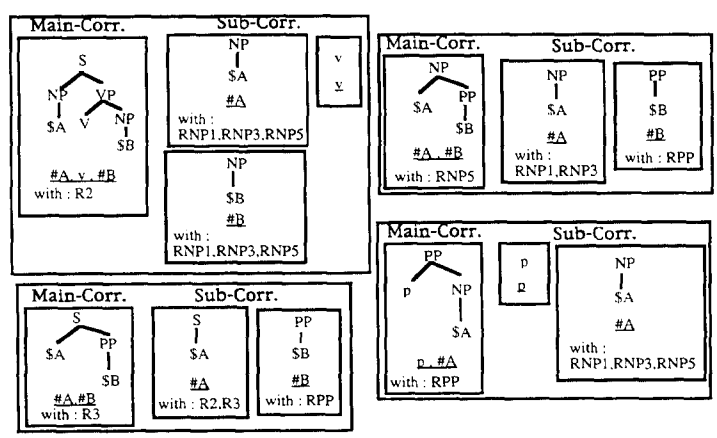


Figure 6 : An AND-OR Graph of STCG grammar rules.



3. Concluding Remarks

Recently, efficient context-free parsing methods such as the LR parser and Earley's Algorithm have been referred to extensively in implementing parsers for most of the formalisms used in the field of NLP. In an effort to retain the efficiency of standard context-free parsing algorithms, most recent declarative formalisms are typically restricted by the constraint of string concatenation in context-free grammars which allows a sentence to be systematically decomposed so that the parsing process can be indexed by the subparts of that decomposition (the substrings). However, it has also been widely recognised that the concatenation restriction of CFG can be problematic in handling phenomena such as lexicalisation, featurisation, and especially crossed dependencies. As an alternative, we propose the STCG formalism which allows for a more 'natural' way of specifiying the strings of the language being described, their corresponding linguistically motivated representation trees, and the correspondence between the two, where the correspondence need not be projective and hence appropriate for the said phenomena. Even though the standard CF parsing methods cannot be adopted directly in the analysis of an input sentence with respect ot a STCG grammar, due to the STRING patterns of the STCG which need not submit to the concatenation restriction of CFG, in this paper we present the general layout (due to the space constraint, however interested readers may get more details in [2]) of an analyser for the STCG which is capable of mimicking some very useful features in various context-free parsing techniques. One such feature is the use of charts in tabular parsing algorithms, as exemplified in Earley's Algorithm [7], which is very helpful in avoiding redundancies that may otherwise result in a combinatorial explosion. Another is the compact way of representing possible parse trees for ambiguous sentences, such as the one seen in [8]. Furthermore, we have also provided a natural way for handling the kind of awkward phenomena such as lexicalisation, featurisation, and worst of all, crossed dependencies, while at the same time retaining much of the efficiency of standard context-free parsing algoritms [2,3].

REFERENCES

[1] Zaharin Y., *String-Tree Correspondence Grammar: a declarative grammar formalism for defining the correspondence between strings of terms and tree structures*, proceedings of the 3rd Conference of the European Chapter of the ACL, Copenhagen, April 1987.

[2] Tang Enya Kong, *Natural languages Analysis in machine translation (MT) based on the STCG*, PhD thesis, Universiti Sains Malaysia, Penang, March 1994 .

[3] Tang Enya Kong, Zaharin Y., *Handling Crossed Dependecies with the STCG*, proceedings of Natural Language Processing Pacific Rim Symposium (NLPRS'95), Sofitel Ambassador Hotel, Seoul, Korea, Dec. 4-6, 1995.

[4] Yves Lepage, Parsing and Generating Context-Sensitive Languages with Correspondence Identification Grammars, proceedings of the Natural Language Processing Pacific Rim Symposium (NLPRS'91), Singapore, 25-26 Nov 1991.

[5] Zaharin Yusoff, Tang Enya Kong, *Generation of analysis programs in ROBRA (ARIANE) From String-Tree Correspondence Grammars (or a Strategy for Analysis in machine tranaslation)*, Proceedings of the 3rd Machine Translation Summit, Washington, D.C., July,1991.

[6] Zaharin Y., Tang Enya Kong, *String-Tree Correspondence Grammars as a base for the automatic generation of analysis programs in machine tranaslation*, proceedings of the International Conference on Current Issues in Computational Linguistics, Penang, June 1991.

[7] J. Earley, *An efficient cantext-free parsing algorithm* , Communications of the ACM, Vol. 13, Num. 2, Feb 1970, pp. 94-102.

[8] Lang, B., *Towards a Uniform Formal Framework for Parsing*, In : Current Issues in Parsing Technology, M. Tomita (ed.), Kluwer Academic Publishers, 1991, pp. 153-171.

[9] Zaharin Y., *Strategies and heuristics in the analysis of natural languages in machine translation*, PhD thesis, Universiti Sains Malaysia, Penang, March 1986.

[10] Y.Lepage, *Un systeme de grammaires correspondancielles d'identification*, these de Docteur, IMAG, Universite Joseph Fourier, Grenoble, June 1989.

[11] Zaharin Yusoff, Christian Boitet, *Representation trees and string-tree correspondences*, proceedings of the 12th International Conference on Computational Linguistics, COLING-88, Budapest, August 1988, pp.59-64.

[12] Ch.Boitet, P.Guillaume, M.Quezel-Ambrunaz, *Le point sur ARIANE-78, debut 1982 (DSE-1), vol.1, part.1 : le logiciel*, GETA, avril 1982.

[13] Tong Loong Cheong, *The JEMAH System : Reference Manual* , UTMK document, USM, Penang, 1988.