TransLite - Development of Lightweight Machine Translation System based on Constraint Synchronous Grammar

Francisco Oliveira¹, Fai Wong¹, Sam Chao¹ and Chi-Wai Tang¹
¹ Faculty of Science and Technology, University of Macau, Macao, China

Received June 22, 2010; Revised March 21, 2011; Accepted 11 June 2011

Abstract: Machine Translation (MT) has stimulated researchers to design different systems for handling the translation task. Typical Rule based MT paradigms involve a long chain of processes, including morphological analysis, part-of-speech tagging, sense disambiguation, parsing, transformation and generation. The development of such systems is not only big but also time consuming, and a careful design have to be specifically related to the languages, since they usually have different properties and grammar. In order to provide a quick way in realizing a translation system for a specific domain or controlled languages between any language pair, a lightweight system, PCT TransLite is proposed. It only relies on parsing Constraint Synchronous Grammar (CSG) in the whole translation process. CSG has the power to describe syntactic relationships between the source and target language simultaneously based on controlled constraints, and to model semantic information in the constituents as features for disambiguation. Since the source syntactic pattern is usually associated with more than one target, the one satisfying all the constraints defined with the rule determines the relationship between them. Moreover, CSG can express non-standard linguistic phenomena easily, including discontinuity and crossing relationships, and words that are vanished or should appear in the translation. The objective in the realization of PCT TransLite is three-fold: in developing simple systems for any language pair rapidly and testing their feasibility when proper rules are defined; in acquiring more rules through an intuitive interface so that they can be parsed by any Context Free Grammar parsing algorithms augmented with constraints and the inference of the target structure; in educating students, for better understanding the MT development and different problems in language analysis through the visualization of syntactic trees. Currently, PCT TransLite has been already used in constructing CSG rules for real MT systems, for education purposes, and got positive feedback.

Keywords: Machine Translation, Constraint Synchronous Grammar

1 Introduction

Nowadays there are many Machine Translation (MT) systems available in the market for helping people in the translation of large amounts of documents. Although these systems are useful, due to the inherent ambiguities and differences between languages, it is currently not possible to have perfect MT systems that always generate translations with good quality. As a result, researchers proposed different MT designs which are either categorized as Corpus-based or Rule-based approaches. Example-based MT (EBMT) [1][2] and Statistic-based MT (SMT) [3][4] systems rely on corpora which comprise of bilingual texts. EBMT systems analyze different pieces of bilingual examples in the extraction and combination of phrases for generating the translation. On the other hand, SMT paradigms rely on probabilities estimated between the translation of words and the
ordering of sentences extracted from the corpora. On both approaches, the accuracy of the translation is highly dependent with the size and information of the corpora. Rule-based MT (RBMT) [5][6] systems handle the translation based on a set of linguistic grammar rules. In typical RBMT paradigms, the translation is based on a pipeline of processes, including morphological analysis for restoring the original format of the source words; part-of-speech tagging for labeling proper constituent information for each word; sense disambiguation in removing improper translations based on the word context; parsing of the source language in an intermediary representation; transfer process in mapping the representation into a target language structure based on a set of conversion rules; and generation for synthesizing the transferred representation into the target language sentence. Since the output generated by each of the processes will be used for subsequent analysis, if any of them generates an error, it is hard to guarantee the quality of the translated result. Moreover, the development time of such systems is big, and different linguistic rules and designs may have to be considered for different language pairs.

In order to accelerate the development time, researchers considered the use of open-source MT systems. Moses [7] and Joshua [8] provide a complete SMT toolkit with the required modules for training the model and handling the translation task. On the other hand, Apertim [9] is a free and open-source MT system based on multi-stage finite-state chunking for structural transfer in RBMT paradigm. All of them provide a detailed guideline in setup the MT system, and related tools for loading the data. However, it is not easy for people who are not familiar with computer to setup such systems.

As a result, this paper presents a lightweight MT system PCT TransLite, which is based on Constraint Synchronous Grammar (CSG) [10] for handling the whole translation task. In other words, the analysis, transformation, and generation from one language into another are done in a one-stage process. PCT TransLite provides a quick and simple way in realizing the MT system for a specific domain or controlled languages, and it is targeted not only for professional users but also people or students who only have basic computer knowledge in the development of a translation system. CSG establishes direct syntactic relationships between the source and target language based on feature constraints. Control conditions are mainly used to disambiguate the correct relationship between the languages since the source syntactic pattern can be associated with more than one target at different situations.

PCT TransLite is developed to meet different objectives. Firstly, in providing an intuitive interface for users to create a simple translation system based on CSG rules for close domain. Since the system is language-pair independent, users are only required to construct grammar rules for the interested language pairs. Secondly, in acquiring more CSG rules in application to real MT systems so that they can be parsed by any Context Free Grammar parsing algorithms, including the Earley [11] and generalized LR algorithms [12] augmented with feature constraints and the inference of the target structure. Finally, in educating students from different study fields. For students of the computer science, PCT TransLite provides an easy way to illustrate the translation process at different levels, especially in the methodologies used in the development of the MT system. For linguistics, it gives a better understanding about the strengths and weaknesses of automatic translation, and how to master the system in the development and translation workflow.

This paper is organized as follows. An introduction of CSG is given in section 2. The design and implementation of PCT TransLite are detailed in section 3. Evaluations are given in section 4, followed by a conclusion.

2 Constraint Synchronous Grammar

Constraint Synchronous Grammar [10] is based on the formalism of Context Free Grammar to the case of synchronous. In this formalism, it consists of a set of production rules which describe the relationship between the source and target language sentential patterns.

By definition, CSG is a tuple $G = (V_N, V_T, C_T, P, S)$, where $V_N$ and $V_T$ are finite, disjoint sets of non-terminal and terminal symbols, respectively; $C_T$ is a finite set of target components, $S$ is denoted as the start symbol and it is an element of $V_N$; $P$ is a finite set of constraint synchronous productions having the form \(A = a\beta\), where $a$ is an element of \((I(V_N) \cup V_T)^*$, and $\beta$ is an element of $C_T$. Non-terminal symbols which occur in both the synchronous formalism are linked by the mapping function $I(V_N)$, where $I(V_N) = \{W_\omega \mid W \in V_N, \omega \in Z\}$, and $Z$ denotes the set of integers.

As an example, suppose that the grammar $G = (\{S, NP, PP, VP, N, V, PREP, ADJ, U\}, \{\} \{\{\} \}$,
CSG is especially effective for modeling non-standard linguistic phenomena for languages which are structurally different. The ordering of the constituents is modeled easily by using the subscripts and the sequence defined in the production rule. As an example, the verb phrase (VP) in the target sentential pattern defined in (2.1) appears in a different order compared to the source syntactic pattern.

The discontinuity between words in different languages is solved by defining non-terminal symbols that appear in the source but not the target pattern or vice-versa. In (2.1), since the words “把” and “借给” should be associated with the verb “to lend”, only one VP is defined for the first target sentential rule associated.

The consideration of constituents that are disappeared or shown in the target syntactical pattern is handled in a similar way. As an example, it often happens that the preposition “的” (of) in the Chinese language is vanished. However, it is necessary to add the word “of” in the target pattern to have a correct translation. As a result, (2.13) can be easily created to model such issue.

According to different context and senses, suitable rules will be selected. As an example, in production rule (2.2), the article “a” should be added to an object in the target sentence but it shouldn’t be associated with a noun if its sense is human. Typically, one word may have different meanings, and the assignment of the best one is controlled by the constraints. Suppose that there is another meaning in the word “黑” as “dark”, with the sense related to environment, besides the one defined in (2.10). The constraint in (2.3) ensures that “black” will be selected instead of “dark”.

CSG rules are parsed by a modified version of generalized LR algorithm, a shift-reduce approach based on an extended LR parsing table. Besides having the actions to be accomplished (shift, reduce, accept), and the state of the parser at different stages of parsing, the table is extended by taking feature’s constraints and target rules into consideration. In other words, as the parser identifies CSG productions through the normal shift actions, it checks the associated constraints to determine if the current reduce action is valid or not.
3 Design of PCT TransLite System

The translation workflow of PCT TransLite, as shown in Figure 3.1, is based on a three-step pipeline: CSG grammar creation, extended LR parsing table creation, and CSG parsing and generation.

In the first stage, users have to decide the languages which will be used in the system. They are required to input grammar rules for translation afterwards. Since PCT TransLite does not have a specific dictionary for any language pair, lexicalized synchronous grammars should be defined, as shown in production rules (2.6)-(2.10). In this case, they represent the English meanings for each element in \( V_T \). On the other hand, since semantic information cannot be retrieved from the bilingual dictionary, features can be encoded explicitly in the pre-terminal rules as shown in (3.1)-(3.5).

\[
\begin{align*}
V & = \text{借給了}[WRD: lend, SUB: human, OBJ: human, TENSE: past] & (3.1) \\
N & = \text{约翰}[WRD: John, SEM: human] & (3.2) \\
N & = \text{钢笔}[WRD: pen, SEM: object, SEM: man-made] & (3.3) \\
N & = \text{佩德羅}[WRD: Peter, SEM: human] & (3.4) \\
ADJ & = \text{黑}[WRD: black, SEM: MOD: NOUN: object] & (3.5)
\end{align*}
\]

A list of features is bracketed and separated by a semi-colon, and for each feature, the names are written in upper case and the values on the right are in the lower case. Multiple features are separated by colon. Information stored in these features can also be used for morphological restorations, gender and number agreements during the generation process. As an example, there are four features associated with the verb “借給” in (3.1). The meaning of the word is “lend”, the sense of the direct and indirect object is human, and the grammatical tense is the present tense.

More information can be encoded in pre-terminal rules to reduce the size of the grammar. As an example, production rules (2.4), (2.6), and (2.12) can be grouped into one as shown in (3.6) with related semantic information encoded.

\[
\begin{align*}
V & = \text{借給了}[WRD: lend, SUB: human, OBJ: human, TENSE: past] & (3.6)
\end{align*}
\]

Finally, the translation is generated by the CSG parsing and generation module relying on the previous constructed table. Whenever the user inputs a sentence for translation, PCT TransLite looks for the parsing table. If the parser meets a non-terminal symbol, it looks for the corresponding task, which can be either a shift, reduce, or accept action. The shift action is used to determine the next symbol in which the parser should follow. The reduce action is performed when the whole sequence of symbols representing a non-terminal symbol is found. As an example, once the source syntactic pattern \( NP_1 \), \( PP \), \( NP_2 \), \( VP \), \( NP_3 \) in (2.1) has been recognized, it can be reduced as \( S \) if the constraints defined are satisfied. Moreover, all the linguistic features defined in \( VP \) are propagated to the reduced non-terminal symbol \( S \).

This process plays an important role in removing different types of ambiguity that might occur during the parsing process. The accept action indicates that the parsing is complete and the input sentence is identified by the system. Since the generation is done at the same time during parsing, once the sentence in question is recognized by the system, the corresponding translation can be constructed easily by looking at the target sentential pattern.

PCT TransLite is executed in Microsoft Windows environment, and the interface, as shown in Figure 3.2, is carefully designed in order to make the user’s interaction with the system as efficient as possible.
In the upper part, all the functions for grammar manipulation and translation are provided. For grammar manipulation, a text area in the center right part is provided for users to input and modify the CSG rules. Once they finish the editing, PCT TransLite converts the grammar rules into the extended LR parsing table. If there are any syntax errors, PCT TransLite prompts a window showing the type of error committed and the line number of the found error. For translation, users just need to input the source sentence in the translation window, and the corresponding translation will be shown in the bottom area.

In order to provide a better understanding on how the sentence and its corresponding translation is generated, PCT TransLite has a function in showing the generated syntactic tree structure of the source sentence. It provides not only an easy way to understand the actions taken by the parser in constructing the tree but also a quick way in getting the translation at each branch of the syntactic tree.

Depending on the type of document and the language to be translated, it is always useful to load suitable terms and grammars to generate better results. In PCT TransLite, users can switch between different grammar rules for handling the translation between different language pairs and domains. Moreover, the parsing table can be reused and shared by other users who work in different machines by the import and export function.

4 Evaluations

In order to investigate the effectiveness of PCT TransLite based on CSG formalism, several experiments are performed for handling the translation task. Firstly, the accuracy of the system is evaluated in a close and open domain. Secondly, the efficiency of the system is measured. The translation from Portuguese to Chinese language is used as an example pair for this evaluation.

The translation quality is evaluated by automatic and human metrics. In automatic assessments, three metrics are considered, including National Institute of Standards and Technology (NIST) [13], Bilingual Evaluation Understudy (BLEU) [14], and Metric for Evaluation of Translation with Explicit Ordering (METEOR) [15]. In human assessments, the average evaluation result of three humans is considered, and the translation quality is classified as Good, Acceptable, and Bad.

The hardware resources and the MT’s knowledge size applied for this evaluation are detailed in Table 4.1.

In the first experiment, a 8k text file is used to evaluate the system in close domain. In order to evaluate the system when the knowledge is out of domain, another 6k test file is applied for open domain. The results of the automatic and human evaluations are shown in Table 4.2.

In terms of automatic assessments, since the knowledge used covers most of the sentences in the close domain, the system gets a BLEU score of 0.8026. However, in the open domain, the score drops drastically due to several reasons. Firstly, since many automatic evaluation metrics rely on n-gram co-occurrence precision in comparing the translated results against the reference answers, the system may generate low scores even the generated sentence is translated well. On the other hand, as all the words and syntactic structures are written in CSG format, if source sentences contain words or grammatical structures which are out of the defined knowledge, an incorrect translation will be generated. In terms of human assessments, the translation quality is 86.1% for Good and Acceptable cases in close domain, and 45.9% in open domain.

In the second experiment, the efficiency of PCT TransLite is measured. Based on the previous test sets, the average translation time is 0.2 seconds per sentence with average length of 7 words.

| Table 4.1: Hardware Specifications and MT’s Knowledge |
|-----------------|-----------------|-----------------|
| **Type**       | **Speed/Size**  |
| Hardware       | Central Processing Unit | 2.7GHz |
|                | Random Access Memory | 2G     |
| Software       | Program          | 3.9MB   |
|                | MT’s Knowledge   | 50KB    |
Table 4.2: Automatic and Human Evaluation Results

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Close</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLEU</td>
<td>0.8026</td>
<td>0.3508</td>
</tr>
<tr>
<td>NIST</td>
<td>9.5181</td>
<td>5.4384</td>
</tr>
<tr>
<td>METEOR</td>
<td>0.8671</td>
<td>0.5430</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Manual</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Good</td>
<td>68.6%</td>
<td>20.6%</td>
</tr>
<tr>
<td>Acceptable</td>
<td>17.5%</td>
<td>25.3%</td>
</tr>
<tr>
<td>Bad</td>
<td>13.9%</td>
<td>54.1%</td>
</tr>
</tbody>
</table>

Since PCT TransLite is targeted for close domain translation and for handling the translation of controlled languages, in most cases, the number of CSG rules stored in the knowledge base is limited. This permits the system to achieve a reasonable accuracy and have good translation efficiency. More CSG syntactic rules and lexicalized information can be added easily in order to further improve the quality and the coverage of the system.

5 Conclusion

In this paper, a lightweight translation system PCT TransLite is proposed for setting up a translation framework efficiently for a close domain. Unlike traditional transfer-based RBMT architectures where the translation process is carried out in sequence based on different modules for structural analysis, transformation, and generation, the system only requires a core analysis and generation module based on CSG formalism. Moreover, different from Statistical and Example based paradigms, PCT TransLite only requires a set of CSG rules for handling the translation task and does not require any information from huge bilingual corpora.

A simple to use interface is developed for better interaction between the users and PCT TransLite. In short, the development of the translation framework is based on three stages. Users first create proper CSG rules for the language pair, and the system will generate the corresponding extended LR parsing table. This is later used in the analysis and generation of the translation given the text to be translated.

Evaluations conducted show the translation quality in close and open domains, and the translation efficiency of PCT TransLite. Currently, the system is successfully being used for two purposes: the creation of CSG rules in application to real translation systems, and for educating linguistic and computer science students in understanding the automatic translation process. They are free to design the translation for any pair of languages according to their preferences. We found that users can easily capture the main concept and techniques in the construction of MT system, and they are stimulated to widen their concerns from the construction of syntactic relationships of bilingual text to the analysis of lexical morphology, part-of-speech assignment, and different types of language ambiguities.

There are still many rooms for improving the current system. Since the whole translation is highly dependent on the CSG rules, we may pre-load some frequent words and meanings for specific languages as lexicalized CSG rules in order to accelerate the development time. Furthermore, semi-automatic extraction methodologies can be considered, like the conversion of skeletal syntactic structures into CSG formalism, and machine learning techniques. In terms of usability, more user-friendly functions are being considered, like CSG syntax highlighting, and error checking on the fly. Currently, PCT TransLite is being implemented in the Web environment to make the system independent of the Operating System and increase the system’s usage.

Acknowledgements

This work was partially supported by the Research Committee of University of Macau under grant UL019B/09-Y3/EEE/LYP01/FST, and also supported by Science and Technology Development Fund of Macau under grant 057/2009/A2.

References


