PARSE DRIVEN TRANSLATION

Petr Horáček
Doctoral Degree Programme (2), FIT BUT
E-mail: xhorac06@stud.fit.vutbr.cz

Supervised by: Alexander Meduna
E-mail: meduna@fit.vutbr.cz

Abstract: This paper presents the idea of translation grammar and syntax driven translation, with focus on natural language translation. It introduces the concept of parse driven translation and contains definitions of formal tools that can be used in this task, such as parse translation grammar and parse translation matrix grammar. A practical example is included, showing a possible application in translating Japanese sentence structure to Czech.

Keywords: language, grammar, syntax driven translation, natural language processing, Japanese-Czech translation

1 INTRODUCTION

Natural language processing (NLP) is an important application area of the formal language theory – attempts at formal analysis and description of the natural language syntax were one of the key factors behind the creation of the theory. However, the basic models of the formal language theory soon proved unsuitable for this task. Since then, new formal models were proposed and studied in both NLP and formal language theory, but the research was largely independent. In our work, we attempt to apply the models of the formal language theory in linguistics, reinforcing the link between the two fields again.

Machine translation is one of the oldest NLP tasks. The first documented idea of using computers in translation is from 1947, by Warren Weaver. In the following years, an extensive research in the area began. However, the practical results were disappointing – only a few working systems were developed, and even their translation quality was unsatisfactory. This led to a decline in interest, and the research was only fully restarted in the seventies. There has been a significant progress since then, with a number of systems applied in practice. In this paper, we propose formal tools that could be used in this task, with focus on direct Japanese-Czech translation.

2 PRELIMINARIES

In this paper, we assume that the reader is familiar with the basic aspects of the modern formal language theory (see [6], [3]) and natural language processing (see [4], [5]). Further information regarding regulated rewriting and matrix grammars can be found in [2].

Definition 2.1 (Context-free grammar)**.** A *context-free grammar (CFG)* G is a quadruple $G = (N, T, P, S)$, where N is a finite set of nonterminals, T is a finite set of terminals, $N \cap T = \emptyset$, P is a finite set of rules, $P \subset N \times (N \cup T)$, $(u, v) \in P$ is written as $u \to v$, and $S \in N$ is the start symbol.

Definition 2.2 (Derivation)**.** Let G be a CFG. Then, a *direct derivation* (denoted as $\Rightarrow$) is defined over strings $x, y \in (N \cup T)^*$, where $x = x_1 u x_2, y = x_1 v x_2$, if and only if there is a rule $u \to v \in P$. We write $x_1 u x_2 \Rightarrow y = x_1 v x_2 [p]$.

We further define $\Rightarrow^+$ as the transitive closure of $\Rightarrow$ and $\Rightarrow^*$ as the transitive and reflexive closure of $\Rightarrow$.

Definition 2.3 (Generated language)**.** Let G be a CFG. The *language generated by* G is defined as $L(G) = \{w : w \in T^*, S \Rightarrow^* w\}$.

Definition 2.4 (Matrix grammar)**.** A *matrix grammar* H is a tuple $H = (G, M)$, where $G = (N, T, P, S)$ is a CFG and M is a finite language over P ($M \subset P^*$), a sentence of this language is called a *matrix*.

Definition 2.5 (Derivation in matrix grammar)**.** Let $H = (G, M)$ be a matrix grammar, $G = (N, T, P, S)$. Let $N = A_1, \ldots, A_m$ for any $m \geq 1$. For any $m_i = p_{i_1} \ldots p_{i_j} \ldots p_{i_{k_i}} \in M$, $p_{i_j} : A_{i_j} \rightarrow x_{i_j}$. Then, for $u, v \in (N \cup T)^*$, $m \in M$ holds that $u \Rightarrow v[m]$ v H, if and only if there are strings $x_0, \ldots, x_n$ such that $u = x_0$, $v = x_n$ and $x_0 \Rightarrow x_1[p_1] \Rightarrow x_2[p_2] \Rightarrow \ldots \Rightarrow x_n[p_n]$ in G, and $m = p_1 \ldots p_n$.

3 SYNTAX AND PARSE DRIVEN TRANSLATION

We propose an approach based on the idea of translation grammar. Informally, a translation grammar is a grammar that generates two corresponding sentences (input and output) in a single derivation. One of the simplest ways to achieve this is to modify the grammar so that every rule has two right-hand sides – the first one generates the input sentence, the second one the output sentence. As there is only one left-hand side, in each derivation step we have to rewrite the same nonterminal in both sentences. Example:

- Rule: 1 : E $\rightarrow$ E + T , E T +
- Corresponding derivation step: (E, E) $\Rightarrow$ (E + T, E T +) [1]

3.1 PARSE TRANSLATION GRAMMAR

For natural language translation, however, we might need a more powerful tool. We propose the parse translation grammar. Informally, it is a system of two grammars with a certain correspondence of their rules. The input and output sentence have the same parse (a sequence of rules applied, denoted by their labels). Example:

Input grammar	Output grammar
1 : E $\rightarrow$ E + T	1 : E $\rightarrow$ E T +

Note that in general, the two corresponding rules do not need to have the same left-hand side, they only need to share the label. Thus, it is for example possible to rewrite a different nonterminal in the input and output sentence in one derivation step.

The translation would in practice proceed as follows:

1. Syntax analysis of the input sentence according to the input grammar – we get a sequence of rules (parse).
$$S_I \Rightarrow^* x_I[\alpha]$$
2. Output sentence generation – we apply the rules of the output grammar according to the sequence from step 1.
$$S_O \Rightarrow^* x_O[\alpha]$$

Definition 3.1 (Parse translation grammar)**.** A *parse translation grammar* is a 5-tuple $H = (G_I, G_O, \Psi, \varphi_I, \varphi_O)$, where

- $G_I = (N_I, T_I, P_I, S_I)$ and $G_O = (N_O, T_O, P_O, S_O)$ are CFG, card P_I = card P_O = card Ψ,
- Ψ is a set of *rule labels*,
- φ_I is a bijection from Ψ to P_I and φ_O a bijection from Ψ to P_O.

We will use the following notation:

$p : A_I \to x_I$ where $p \in \Psi, A_I \to x_i \in P_I$	$\varphi_I(p) = A_I \to x_I$
$x_I \Rightarrow_{G_I} y_I[p]$ where $x_I, y_I \in (N \cup T)^*, p \in \Psi$	derivation step in G_I applying rule $\varphi_I(p)$
$x_I \Rightarrow^n_{G_I} y_I[p_1 \dots p_n]$ where $x_I, y_I \in (N \cup T)^*, p_i \in \Psi$ for $1 \leq i \leq n$	derivation in G_I applying rules $\varphi_I(p_1) \dots \varphi_I(p_n)$

Analogous for output grammar G_O.

Definition 3.2 (Translation)**.** Let $H = (G_I, G_O, \Psi, \varphi_I, \varphi_O)$ be a parse translation grammar. *Translation* $T(H)$ is a set of pairs of sentences, defined as

$$T(H) = \{(w_I, w_O) : w_I \in T_I^*, w_O \in T_O^*, S_I \Rightarrow^*_{G_I} w_I[\alpha], S_O \Rightarrow^*_{G_O} w_O[\alpha], \alpha \in \Psi^*\}.$$

3.2 Parse Translation Matrix Grammar

A CFG by itself does not have enough generative power to describe natural languages. However, we can easily apply the same approach to other grammars as well. We propose a parse translation matrix grammmar, which is analogous to the parse translation grammar above. The main difference is that instead of a sequence of rules, the translation will be based on a sequence of matrices.

Definition 3.3 (Parse translation matrix grammar)**.** A *parse translation matrix grammar* is a 7-tuple $H = (G_I, M_I, G_O, M_O, \Psi, \varphi_I, \varphi_O)$, where

- (G_I, M_I) and (G_O, M_O) are matrix grammars, card M_I = card M_O = card Ψ,
- Ψ is a set of *matrix labels*,
- φ_I is a bijection from Ψ to M_I and φ_O a bijection from Ψ to M_O.

We will use the following notation:

$m : t_I \to x_I$ where $m \in \Psi, t_i \in M_I$	$\varphi_I(m) = t_I$
$x_I \Rightarrow_{G_I, M_I} y_I[m]$ where $x_I, y_I \in (N \cup T)^*, m \in \Psi$	derivation step in G_I, M_I applying matrix $\varphi_I(m)$
$x_I \Rightarrow^n_{G_I, M_I} y_I[m_1 \dots m_n]$ where $x_I, y_I \in (N \cup T)^*$, $m_i \in \Psi$ for $1 \leq i \leq n$	derivation in G_I, M_I applying matrices $\varphi_I(m_1) \dots \varphi_I(m_n)$

Analogous for output grammar (G_O, M_O).

Definition 3.4 (Translation – matrix grammar)**.** Let $H = (G_I, M_I, G_O, M_O, \Psi, \varphi_I, \varphi_O)$ be a parse translation matrix grammar. *Translation* $T(H)$ is a set of pairs of sentences, defined as

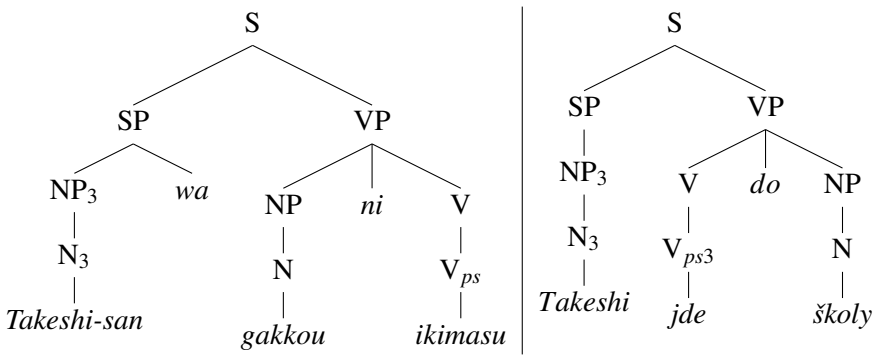
$$T(H) = \{(w_I, w_O) : w_I \in T_I^*, w_O \in T_O^*, S_I \Rightarrow^*_{(G_I, M_I)} w_I[\alpha], S_O \Rightarrow^*_{(G_O, M_O)} w_O[\alpha], \alpha \in \Psi^*\}.$$

3.3 JAPANESE-CZECH TRANSLATION EXAMPLE

In our work, we focus on application possibilites in translating Japanese sentence structure to Czech. Here we will present an example dealing with verb inflection. In the Czech language, the verb form reflects not only the tense, but other grammatical categories such as person, number and gender as well. In Japanese there is no such distinction (out of the grammatical categories mentioned, only the tense affects the inflection). When translating a sentence from Japanese to Czech, we need to be able to select the appropriate word form. For instance, consider the following sentences (Japanese on the left, Czech on the right):

watashi wa gakkou ni ikimasu	*já jdu do školy*
anata wa gakkou ni ikimasu	*ty jdeš do školy*
Takeshi-san wa gakkou ni ikimasu	*Takeshi jde do školy*

The meaning is, respectively, *I go to school*, *you go to school* and *Takeshi goes to school*. As we can see, the verb in the Japanese sentence (*ikimasu*, long form of the verb *iku*, to go) is unaffected by the person. In Czech we need to distinguish between the first-person (*jdu*), the second-person (*jdeš*) and the third-person form (*jde*).



We can tell which form to use by looking at the subject. Consider a parse translation matrix grammar with the following rules (left side is input grammar, right output grammar):

1a:	SP	→	NP_1	1a:	SP	→	NP_1
1b:	SP	→	NP_2	1b:	SP	→	NP_2
1c:	SP	→	NP_3	1c:	SP	→	NP_3
2:	V	→	V_{ps}	2a:	V	→	V_{ps1}
				2b:	V	→	V_{ps2}
				2c:	V	→	V_{ps3}

and matrices:

A:	1a 2	A:	1a 2a
B:	1b 2	B:	1b 2b
C:	1c 2	C:	1c 2c

Regardless of which matrix (A, B or C) we choose to apply in the input gramar (Japanese), the verb form will remain the same, because we always use the same rule to rewrite the nonterminal V (rule 2). But the corresponding matrix in the output grammar (Czech) determines which rule to use (2a, 2b or 2c), and we can generate the correct form of the verb.

4 CONCLUSION

In this paper, we proposed a syntax driven approach to translation, and provided formal tools based on CFG and matrix grammar. Presented practical example demonstrates a possible use of these tools in translation from the Japanese language to the Czech language. For practical experiments and applications in machine translation, we would need to incorporate the parse translation models into a more complex system (for instance, before the syntax analysis itself, a tool to perform morphological analysis would be required). Various systems that use syntactic information in translation have been developed, often relying on a combination of known parsing methods (such as CKY) with statistical approaches and machine learning, as presented for example in [7] or [1]. However, that is beyond the scope of this paper. It should also be noted that the formal tools proposed in this paper are in no way restricted to NLP, and could be applied in other translation or transformation tasks as well (compilers...).

There are several options for future research. We could further study the theoretical properties of the proposed formal models, mainly their generative power and the classes of languages they define. From a more practical point of view, since our approach to translation is based on syntax analysis, we also need to study and develop methods and algorithms for parsing. This is simple in case of CFG (there are well-known methods, as mentioned above), but in case of matrix grammars, there has been relatively little research in the area. Intuitively, considering that matrix grammars are a straightforward extension of CFG, it seems that we should be able to adapt the existing parsing algorithms for CFG.

ACKNOWLEDGEMENT

This work was partially supported by the research plan MSM0021630528 and the grant MSMT FRVS FR97/2011/G1.

REFERENCE

[1] Bojar, O., Čmejrek, M.: *Mathematical Model of Tree Transformations*. EuroMatrix Deliverable 3.2, December, 2007, `http://ufal.mff.cuni.cz/euromatrix/`.

[2] Dassow, J., Păun, Gh.: *Regulated Rewriting in Formal Language Theory*. Berlin: Springer, 1989, ISBN 3-540-51414-7.

[3] Meduna, A.: *Automata and Languages: Theory and Applications* [Springer, 2000]. Springer Verlag, 2005, ISBN 1-85233-074-0, 892 s. Press, UK, WIT Press, 2010, ISBN 978-1-84564-426-0, 224 s.

[4] Mitkov, R.: *The Oxford Handbook of Computational Linguistics*. Oxford University Press, 2004, ISBN 978-0-19-927634-9.

[5] Novotný, M.: *S algebrou od jazyka ke gramatice a zpět*. Academia Praha, 1988.

[6] Rozenberg, G., Salomaa, A.: *Handbook of Formal Languages: Volume I*. Springer Verlag, 1997.

[7] Zollmann, A., Venugopal, A.: Syntax Augmented Machine Translation via Chart Parsing. In *NAACL 2006 - Workshop on statistical machine translation*, New York. June 4-9, 2006.