Syntax Driven Japanese-Czech Translation

Petr Horáček

Department of Information Systems
Faculty of Information Technology
Brno University of Technology

November 4, 2010

Outline

Outline

1. Introduction
   - Motivation
   - Idea
2. Definitions
   - Preliminaries
   - Parse Translation Grammar
   - Parse Translation Matrix Grammar
3. Examples
   - Translating Japanese sentence structure to Czech

Motivation

Automated Translation of Natural Languages

- One of the major NLP tasks
- Practical applications
- Japanese-Czech translation – little research

Syntax Driven Translation

- Well-known concept
- Used in practice (compilers)
- Corresponds to human learning of languages

Syntax Driven Translation

Translation grammar (basic idea)

- A grammar that generates two corresponding sentences (input and translation) in one derivation
- Based on CFG (usually)
- Each rule has two right-hand sides – one generates the input sentence, other the corresponding output sentence
- One left-hand side – always rewriting the same nonterminal

Example

- Rule:

$$1 : \mathsf{E} \rightarrow \mathsf{E} + \mathsf{T} \text{ , } \mathsf{E} \; \mathsf{T} \; +$$

- Derivation step:

$$(\mathsf{E}, \mathsf{E}) \Rightarrow (\mathsf{E} + \mathsf{T}, \mathsf{E} \; \mathsf{T} \; +) \; [1]$$

Parse Driven Translation

Idea

- Based on the the idea of syntax driven translation and translation grammars
- Two grammars (input and output), corresponding rules share labels
- Input sentence and output sentence – same parse (sequence of rules used in derivation, denoted by their labels)
- Example – rules:

Input grammar	Output grammar
$1 : \mathsf{E} \rightarrow \mathsf{E} + \mathsf{T}$	$1 : \mathsf{E} \rightarrow \mathsf{E}\ \mathsf{T}\ +$

- Note: the two corresponding rules do not need to rewrite the same nonterminal

Parse Driven Translation

Translation in practice (idea)

1. Parse the input sentence using input grammar – we get a sequence of rules (parse)

$$S_I \Rightarrow^* x_I[\alpha]$$

2. Generate the translation using output grammar – apply the rules of output grammar according to the sequence from step 1

$$S_O \Rightarrow^* x_O[\alpha]$$

Parse Driven Translation

Dealing with context

- CFG might not have enough generative power to describe natural languages
- We can use grammars with regulated rewriting, such as matrix grammar

Matrix grammar – motivation

- Relatively simple and straightforward extension of CFG
- Easy to describe and translate grammatical rules, structures and relations
- Practical use? (not “too powerful”)

Outline

1. Introduction
   - Motivation
   - Idea
2. Definitions
   - Preliminaries
   - Parse Translation Grammar
   - Parse Translation Matrix Grammar
3. Examples
   - Translating Japanese sentence structure to Czech

Context-Free Grammar

Definition

A context-free grammar (CFG) is a quadruple $G = (N, T, P, S)$, where

- N is a finite set of *nonterminal* symbols
- T is a finite set of *terminal* symbols, $N \cap T = \emptyset$
- P is a finite relation from N to $(N \cup T)^*$, usually represented as a finite set of *rules (productions)* of the form $A \rightarrow x$, where $A \in N$ and $x \in (N \cup T)^*$
- $S \in N$ is the *start symbol*

Derivation step and generated language

Let $u, v \in (N \cup T)^*$ and $p = A \rightarrow x \in P$. Then, *uAv directly derives uxv* according to p in G, written as $uAv \Rightarrow_G uxv\ [p]$ or simply $uAv \Rightarrow uxv$.

$$L(G) = \{w : w \in T^*, S \Rightarrow^* w\}$$

Matrix Grammar

Definition

A matrix grammar is a pair $H = (G, M)$, where

- $G = (N, T, P, S)$ is a context-free grammar
- M is a finite language over P ($M \subseteq P^*$)

Notation

- Let $N = A_1, \ldots, A_m$ for some $m \geq 1$
- For some $m_i = p_{i_1} \ldots p_{i_j} \ldots p_{i_{k_i}} \in M$,

$$p_{i_j} : A_{i_j} \rightarrow x_{i_j}$$

Matrix Grammar

Derivation step

For $x, y \in (N \cup T)^*$, $m \in M$,

$$x \Rightarrow y[m]$$

in H if there are $x_0, \dots, x_n$ such that $x = x_0, x_n = y$, and

1. $x_0 \Rightarrow x_1[p_1] \Rightarrow x_2[p_2] \Rightarrow \cdots \Rightarrow x_n[p_n]$ in G, and
2. $m = p_1 \dots p_n$

Generated language

$$L(H) = \{x : x \in T^*, S \Rightarrow^* x\}$$

Parse Translation Grammar

Definition

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 context-free grammars and card $P_I =$ card P_O
- Ψ is a set of symbols (*rule labels*), φ_I is a bijection from Ψ to P_I and φ_O a bijection from Ψ to P_O

Parse Translation Grammar

Notation

$p : A_I \rightarrow x_I$ — $\varphi_I(p) = A_I \rightarrow x_I$

where $p \in \Psi, A_I \rightarrow x_i \in P_I$

$x_I \Rightarrow_{G_I} y_I[p]$ — one derivation step in G_I, applying rule $\varphi_I(p)$

where $x_I, y_I \in (N \cup T)^*, p \in \Psi$

$x_I \Rightarrow^n_{G_I} y_I[p_1 \dots p_n]$ — derivation in G_I, applying rules $\varphi_I(p_1) \dots \varphi_I(p_n)$

where $x_I, y_I \in (N \cup T)^*, p_i \in \Psi$ for $1 \leq i \leq n$

Analogous for output grammar G_O.

Translation

Translation $T(H)$ is a set of pairs of sentences:

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

where $\alpha \in \Psi^*$

Parse Translation Matrix Grammar

Definition

A parse translation matrix grammar is a 7-tuple

$$H = (G_I, M_I, G_O, M_O, \Psi, \phi_I, \phi_O)$$

where

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

Parse Translation Matrix Grammar

Notation

$m : t_I$ 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 $x_I, y_I \in (N \cup T)^*, m \in \Psi$	one 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 $\varphi_I(m_1) \dots \varphi_I(m_n)$

Translation

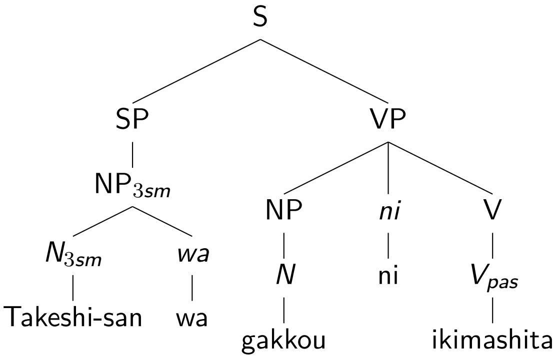
Translation $T(H)$ is a set of pairs of sentences:

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

where $\alpha \in \Psi^*$

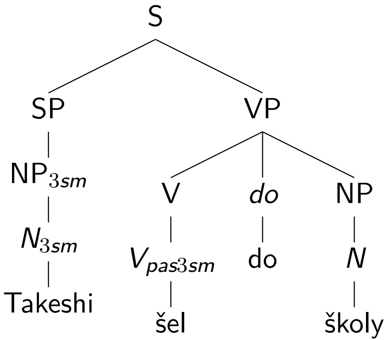
Outline

Example 1: Subject and Verb (1/2)



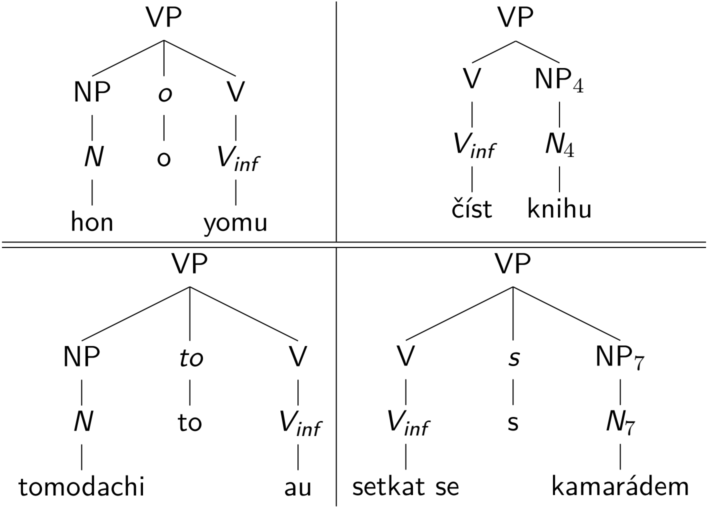
$p1$:	S $\rightarrow$ SP VP
$p2$:	SP $\rightarrow$ NP_{3sm}
$p3$:	NP_{3sm} $\rightarrow$ N_{3sm} wa
$p4$:	VP $\rightarrow$ NP *ni* V
$p5$:	V $\rightarrow$ V_{pas}
$p6$:	NP $\rightarrow$ N
1:	$p1$
2:	$p2$ $p5$
3:	$p3$
4:	$p4$
5:	$p6$
	1 4 2 3 5

Example 1: Subject and Verb (2/2)



$p1$:	S $\rightarrow$ SP VP
$p2$:	SP $\rightarrow$ NP$_{3sm}$
$p3$:	NP$_{3sm}$ $\rightarrow$ N_{3sm}
$p4$:	VP $\rightarrow$ V *do* NP
$p5$:	V $\rightarrow$ V_{pas3sm}
$p6$:	NP $\rightarrow$ N
1:	$p1$
2:	$p2$ $p5$
3:	$p3$
4:	$p4$
5:	$p6$
	1 4 2 3 5

Example 2: Verb Phrase – Object



References

- A. Meduna: *Elements of Compiler Design*, T & F, New York, US, 2008
- J. Dassow, Gh. Păun: *Regulated Rewriting in Formal Language Theory*, Akademie-Verlag, Berlin, 1989.
- E. Banno, Y. Ohno, Y. Sakane, C. Shinagawa: *Genki 1: An Integrated Course in Elementary Japanese*, The Japan Times, 1999

Thank you for attention