CL Seminar 1

Pattern-Based Context-Free Grammars for Machine Translation by Koichi Takeda

Abhinav S Menon

Overview

- Context-Free Grammars
- Pattern-Based CFGs
- The Translation Algorithm
- Features and Agreement
- An Example

Context-Free Grammars

Context-Free Grammars (CFGs) are a common formalism for describing the syntax of natural (and programming) languages. For example,

$\mathbb{S} \rightarrow \mathbb{NP}\ \mathbb{VP}$
$\mathbb{NP} \rightarrow \mathbb{A}\ \mathbb{N}$
$\mathbb{VP} \rightarrow \mathbb{V}\ \mathbb{NP}$
$\mathbb{N} \rightarrow \texttt{cat} \mid \texttt{dog} \mid \texttt{mouse}$
$\mathbb{A} \rightarrow \texttt{the} \mid \texttt{a}$
$\mathbb{V} \rightarrow \texttt{chased} \mid \texttt{hugged}$

Pattern-Based CFGs: Patterns

Pattern-Based CFGs or PCFGs consist of a set of patterns – analogous to the set of rules that make up a CFG.

Each pattern describes how to translate a certain type of phrase from the source to the target language.

An example pattern is

$\mathbb{NP}_1$ [miss]$\mathbb{V}_2$ $\mathbb{NP}_3 \leftarrow \mathbb{S}_2$
$\mathbb{S}_2 \rightarrow \mathbb{NP}_3$ [manquer]$\mathbb{V}_2$ à $\mathbb{NP}_1$

Pattern-Based CFGs: Skeletons

The two rules (one in the source language and one in the target language) that make up the pattern are called its *skeleton*.

For example, the skeleton of the example above

$\mathbb{NP}_1$ [miss]$\mathbb{V}_2$ $\mathbb{NP}_3 \leftarrow \mathbb{S}_2$
$\mathbb{S}_2 \rightarrow \mathbb{NP}_3$ [manquer]$\mathbb{V}_2$ à $\mathbb{NP}_1$

is

$\mathbb{NP}$ $\mathbb{V}$ $\mathbb{NP} \leftarrow \mathbb{S}$
$\mathbb{S} \rightarrow \mathbb{NP}$ $\mathbb{V}$ à $\mathbb{NP}$

Pattern-Based CFGs: Constraints

$\mathbb{NP}_1$ [miss]$\mathbb{V}_2$ $\mathbb{NP}_3 \leftarrow \mathbb{S}_2$
$\mathbb{S}_2 \rightarrow \mathbb{NP}_3$ [manquer]$\mathbb{V}_2$ à $\mathbb{NP}_1$

There are two types of constraints in the above pattern:

- Link constraints (subscripts): specifying the correlation between the constituents of each part.
- Head constraints (enclosed in [ ]): specifying lexeme- or phrase-level conditions on the applicability of the pattern.

The latter makes it possible to translate phrases that are constructed differently in the target language, as in the above example.

The Translation Algorithm: Essentials

The basic algorithm consists of three simple steps:

- Parse the input using the source language CFG skeletons.
- Propagate link and head constraints from the source to the target language and build a target language derivation sequence.
- Generate the output using the target sequence.

The Translation Algorithm: Parsing

Parsing of CFGs is a well-studied area. Many algorithms are known for this task – we will consider the one described in Earley (1970).

This algorithm consists of stepping through the input string one at a time, and trying to parse the upcoming symbols given the incomplete parses until the current point.

It generates all possible derivations for the input string (sometimes called a *parse forest*) in the given CFG.

The Translation Algorithm: Priorities

Under some conditions, it is very easy for the number of target skeletons to increase exponentially for the same number of source skeletons.

Therefore, when two or more patterns share the same source skeleton, we must decide which pattern is more likely to yield a correct translation.

The Translation Algorithm: Priorities

The priority order is as follows:

- Prefer a pattern with more head constraints (a more *specific* pattern).
- Prefer a pattern with more terminal symbols.
- Prefer the shortest derivation sequence.

These preferences can be expressed as numeric *costs* for patterns.

Features and Agreement

The constraints can be extended to lay restrictions on the features of words or phrases as well, to increase its descriptive power.

For example, in French, pronouns as direct objects come *before* the verb. This can be expressed as

$$\mathbb{V}_1\ \mathbb{NP}_2 \leftarrow \mathbb{VP}_1\{+\texttt{OBJ}\}$$
$$\mathbb{VP}_1\{+\texttt{OBJ}\} \rightarrow \mathbb{NP}_2\{+\texttt{PRO}\}\ \mathbb{V}_1$$

Integration of Bilingual Corpora

Given a bilingual corpus, we can translate all the sentences and compare them.

If the translation is correct, nothing needs to be done.

If there was a derivation sequence that resulted in an incorrect translation, use the vocabulary as head constraints.

If there was no paired derivation sequence, add specific patterns for idioms or collocations (if any).

An Example

$$\mathbb{NP}_1\ \mathbb{VP}_1 \leftarrow \mathbb{S}_1$$
$$\mathbb{S}_1 \rightarrow \mathbb{NP}_1\ \mathbb{VP}_1$$

$$\mathbb{VP}_1\ \mathbb{ADVP}_2 \leftarrow \mathbb{VP}_1$$
$$\mathbb{VP}_1 \rightarrow \mathbb{VP}_1\ \mathbb{ADVP}_2$$

$$[\texttt{know}]\mathbb{VP}_1\{+\texttt{OBJ}\}\ \texttt{well} \leftarrow \mathbb{VP}_1$$
$$\mathbb{VP}_1 \rightarrow [\texttt{connaître}]\mathbb{VP}_1\{+\texttt{OBJ}\}\ \texttt{bien}$$

$$\mathbb{V}_1\ \mathbb{NP}_2 \leftarrow \mathbb{VP}_1\{+\texttt{OBJ}\}$$
$$\mathbb{VP}_1\{+\texttt{OBJ}\} \rightarrow \mathbb{V}_1\ \mathbb{NP}_2\{-\texttt{PRO}\}$$

$$\mathbb{V}_1\ \mathbb{NP}_2 \leftarrow \mathbb{VP}_1\{+\texttt{OBJ}\}$$
$$\mathbb{VP}_1\{+\texttt{OBJ}\} \rightarrow \mathbb{NP}_2\{+\texttt{PRO}\}\ \mathbb{V}_1$$

An Example

$\texttt{he} \leftarrow \mathbb{NP}\{+\texttt{PRO} + \texttt{NOMI} + \texttt{3RD} + \texttt{SG}\}$
$\mathbb{NP}\{+\texttt{PRO} + \texttt{NOMI} + \texttt{3RD} + \texttt{SG}\} \rightarrow \texttt{il}$

$\texttt{me} \leftarrow \mathbb{NP}\{+\texttt{PRO} + \texttt{CAUS} - \texttt{3RD} + \texttt{SG}\}$
$\mathbb{NP}\{+\texttt{PRO} + \texttt{CAUS} - \texttt{3RD} + \texttt{SG}\} \rightarrow \texttt{me}$

$\texttt{knows} \leftarrow \mathbb{V}\{+\texttt{FIN} + \texttt{3SG}\}$
$\mathbb{V}\{+\texttt{FIN} + \texttt{3SG}\} \rightarrow \texttt{sait}$

$\texttt{knows} \leftarrow \mathbb{V}\{+\texttt{FIN} + \texttt{3SG}\}$
$\mathbb{V}\{+\texttt{FIN} + \texttt{3SG}\} \rightarrow \texttt{connait}$

$\texttt{well} \leftarrow \mathbb{ADVP}$
$\mathbb{ADVP} \rightarrow \texttt{bien}$

$\texttt{well} \leftarrow \mathbb{ADVP}$
$\mathbb{ADVP} \rightarrow \texttt{beaucoup}$

An Example

According to these rules, the sentence `He knows me well` is parsed.

Fragment	Translation
`He knows me well`	$\mathbb{S}$
`He`	`il`
`knows me well`	$\mathbb{VP}$
`well`	`bien`
`knows me`	$\mathbb{VP}$
`knows`	`connait`
`me`	`me`

We get the correct output: `Il me connait bien`.

References

Takeda, Koichi. “Pattern-based context-free grammars for machine translation.” arXiv preprint cmp-lg/9607011 (1996).