Tree Automata in Parsing and Machine Translation

Andreas Maletti

Institute of Computer Science
University of Leipzig

Leipzig — November 22, 2016
Parsing
determining the syntactic structure of a sentence
subject to a given theory of syntax (encoded in the training data)
▶ constituent syntax
▶ dependency syntax
▶ …
determining the syntactic structure of a sentence
subject to a given theory of syntax (encoded in the training data)
- constituent syntax
- dependency syntax
- ...
Example: We must bear in mind the Community as a whole
Example: *We must bear in mind the Community as a whole*

**POS-tag:** part-of-speech tag, “class” of a word
Constituent Parsing

today

Subcategorization
automatic, e.g. Berkeley (2007)

2000

Statistical approach (cheap, automatically trained)
Penn and WSJ tree bank (1M and 30M words)
automatically obtained weighted CFG

1990

Chomskyan approach (perfect analysis, poor coverage)
hand-crafted CFG, TAG (refined via POS tags)
corrections and selection by human annotators
### Constituent Parsing

<table>
<thead>
<tr>
<th>grammar</th>
<th>$F_1$-score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$</td>
</tr>
<tr>
<td>wCFG</td>
<td>62.7</td>
</tr>
<tr>
<td>wTSG [Post, Gildea, 2009]</td>
<td>82.6</td>
</tr>
<tr>
<td>wTSG [Cohn et al., 2010]</td>
<td>85.4</td>
</tr>
<tr>
<td>wCFG$_{sub}$ [Collins, 1999]</td>
<td>88.6</td>
</tr>
<tr>
<td>wCFG$_{sub}$ [Petrov, Klein, 2007]</td>
<td>90.6</td>
</tr>
<tr>
<td>wCFG$_{sub}$ [Petrov, 2010]</td>
<td>91.8</td>
</tr>
<tr>
<td>wTSG$_{sub}$ [Shindo et al., 2012]</td>
<td>92.9</td>
</tr>
</tbody>
</table>
Subcategorization

Tags:
- official tags often conservative
  - English: \( \approx 50 \) tags
  - German: \( \gg 200 \) tags

ADJA-Sup-Dat-Sg-Fem

```
NP  VP
PRP$ NN  VBZ
  My  dog  sleeps
```
Tags:

- official tags often conservative
  - English: \(\approx 50\) tags
  - German: \(\gg 200\) tags
- all modern parsers use refined tags \(\rightarrow\) subcategorization

```
My dog sleeps
```

ADJA-Sup-Dat-Sg-Fem

Subcategorization
Subcategorization

Tags:
- official tags often conservative
  - English: \(\approx\) 50 tags
  - German: \(\gg\) 200 tags
- all modern parsers use refined tags \(\rightarrow\) subcategorization
- but return parse over official tags \(\rightarrow\) relabeling

```
S
  NP
    PRP$ My
    NN dog
  VP
    VBZ sleeps
```
Definition (Tree automaton)

Tuple \((Q, \Sigma, I, R)\)
- finite set \(Q\) of states
- finite set \(\Sigma\) of terminals
- initial states \(I \subseteq Q\)
- finite set \(R\) of rules of the form \(q \rightarrow \sigma(q_1, \ldots, q_k)\)

\((\sigma \in \Sigma, k \geq 0, q, q_1, \ldots, q_k \in Q)\)

Example rules

\[
q_4 \rightarrow q_5 \quad q_0 \rightarrow S \quad q_0 \rightarrow S
\]

\[
q_2 \quad q_3 \quad q_1 \quad q_4 \quad q_6 \quad q_2
\]
Definition (Derivation semantics and recognized tree language)

Let \((Q, \Sigma, I, R)\) tree automaton

- for each leaf position labeled \(q\) and rule \(q \rightarrow r \in R\)

- recognized tree language

\[\{ t | \exists q \in I: q \Rightarrow^* t\}\]
### Constituent Parsing

<table>
<thead>
<tr>
<th>Grammar</th>
<th>$F_1$-score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$</td>
</tr>
<tr>
<td>wCFG</td>
<td></td>
</tr>
<tr>
<td>wTSG [Post, Gildea, 2009]</td>
<td>82.6</td>
</tr>
<tr>
<td>wTSG [Cohn et al., 2010]</td>
<td>85.4</td>
</tr>
<tr>
<td>wCFG sub [Collins, 1999]</td>
<td>88.6</td>
</tr>
<tr>
<td>wCFG sub [Petrov, Klein, 2007]</td>
<td>90.6</td>
</tr>
<tr>
<td>wCFG sub [Petrov, 2010]</td>
<td></td>
</tr>
<tr>
<td>wTSG sub [Shindo et al., 2012]</td>
<td>92.9</td>
</tr>
</tbody>
</table>

Hence: subcategorization = /finite-state

All modern models equivalent to tree automata in expressive power.
Constituent Parsing

| Grammar      | $|w| \leq 40$ | full |
|--------------|-------------|------|
| wCFG         | 62.7        |      |
| wTSG [Post, Gildea, 2009] | 82.6        |      |
| wTSG [Cohn et al., 2010]   | 85.4        | 84.7 |
| wCFG$_{sub}$ [Collins, 1999] | 88.6        | 88.2 |
| wCFG$_{sub}$ [Petrov, Klein, 2007] | 90.6        | 90.1 |
| wCFG$_{sub}$ [Petrov, 2010] | 91.8        |      |
| wTSG$_{sub}$ [Shindo et al., 2012] | 92.9        | 92.4 |

Hence: subcategorization = finite-state all modern models equivalent to tree automata in expressive power.
Constituent Parsing

| Grammar       | $F_1$-Score | $|w| \leq 40$ | Full |
|---------------|-------------|----------------|------|
| wCFG          | 62.7        |                |      |
| wTSG [Post, Gildea, 2009] | 82.6       |                |      |
| wTSG [Cohn et al., 2010] | 85.4, 84.7 |                |      |
| wCFG$_{sub}$ [Collins, 1999] | 88.6, 88.2 |                |      |
| wCFG$_{sub}$ [Petrov, Klein, 2007] | 90.6, 90.1 |                |      |
| wCFG$_{sub}$ [Petrov, 2010] | 91.8        |                |      |
| wTSG$_{sub}$ [Shindo et al., 2012] | 92.9, 92.4 |                |      |

Hence:

- subcategorization = finite-state
- all modern models equivalent to tree automata in expressive power
Constituent Parsing

Comparison:
- rule of subcategorized CFG vs. corresponding rule of tree automaton

\[
S-1 \rightarrow \text{ADJP-2} \quad S-1 \quad S-1 \rightarrow S(\text{ADJP-2}, \ S-1)
\]

Advances in NLP
- best learning algorithms from positive data (state splitting & EM)
- fastest evaluators of weighted tree automata (coarse-to-fine parsing)
- fastest $n$-best derivation extraction
- …
Parsing

determining the syntactic structure of a sentence
subject to a given theory of syntax (encoded in the training data)
  ▶ constituent syntax
  ▶ dependency syntax
  ▶ …

T John saw a dog yesterday which was a Yorkshire Terrier
John saw a dog yesterday which was a Yorkshire Terrier.

Practical results:
- Linear-time statistical parsers
- Google’s “Parsey McParseface” [Andor et al., 2016]
  - $94\%$ F1-score; linguists achieve $96–97\%$
John saw a dog yesterday which was a Yorkshire Terrier.
John saw a dog yesterday which was a Yorkshire Terrier.

Practical results:
- linear-time statistical parsers
- Google’s “Parsey McParseface” [Andor et al., 2016]
  94% $F_1$-score; linguists achieve 96–97%
Dependency Parsing

Theoretical problems

Given edge-weighted directed graph, extract “best” edge cover

- (general) [Edmonds, 1965]
- that is a tree [Chu-Liu & Edmonds, 1965–1967]
- that is projective tree [Eisner, 1996]
- that is acyclic NP-hard [Guruswami et al., 2011]
- that is a tree with page-number 2 [Gómez-Rodríguez & Nivre, 2013]
- that has page-number $k \geq 2$ NP-hard [Kuhlmann & Jonsson, 2015]
- that is a tree with page-number $k \geq 3$ open
The lexicon generates string language $\mathcal{L}$ with $\mathcal{L} \cap c^+d^+e^+ = \{c^i d^i e^i \mid i \geq 1\}$ for goal item $D$

- $L(c) = \{C\}$
- $L(d) = \{D/E\backslash C, \ D/E/D\backslash C\}$
- $L(e) = \{E\}$
Under a suitable relabeling, characterize the set of valid proof trees
- for just applications \(\rightarrow\) sub-regular tree languages
- for compositions of order 1 \(\rightarrow\) open (probably still regular)
- for compositions of order \(k \geq 2\) \(\rightarrow\) open
- for arbitrary compositions \(\rightarrow\) context-free tree language

ongoing work with Marco Kuhlmann
Lexicalization

**Definition (lexicalized)**

A grammar is **lexicalized** if each rule contains a lexical item.

**Existing results**

- CFG weakly lexicalize themselves
- TAG weakly lexicalize themselves
- TAG strongly lexicalize CFG and TSG
- CFTG strongly lexicalize TAG and themselves
- \((d + 1)\)-TAG strongly lexicalize \(d\)-TAG

- Greibach normal form
- [Schabes, 1990]
- [Schabes, 1990]
- [M, Engelfriet, 2012]
- [De Santo et al., 2016]
Lexicalization

Multiple context-free tree grammar:

\[
A \rightarrow T_1 \sigma T_3 \quad B \rightarrow B \beta T_2 x_1 \quad B' \rightarrow B \beta x_1 \quad T_1 \rightarrow B \beta
\]

\[
S \rightarrow \alpha A \quad B \rightarrow x_1 \quad B' \rightarrow x_1 \quad T_1 \rightarrow T_2 T_3 x_1 \tau \nu
\]
Lexicalization

Derivation tree and evaluation:

\[
\rho_4 \quad B' \quad A \quad \rho_3 \quad \rho_2 \quad B \quad \rho_5 \quad \rho_6
\]

\[
\begin{array}{c}
\rho_4 \quad B' \quad A \quad \rho_3 \quad \rho_2 \quad B \quad \rho_5 \quad \rho_6
\end{array}
\]

MCFTG strongly lexicalize themselves and inv. of their expressive power ongoing work with Joost Engelfriet and Sebastian Maneth
MCFTG strongly lexicalize themselves and inv. of their expressive power ongoing work with Joost Engelfriet and Sebastian Maneth
<table>
<thead>
<tr>
<th></th>
<th>Review translation [by Google Translate]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The room it is not narrowly was a simple, bathtub was also attached.</td>
</tr>
<tr>
<td>2</td>
<td>Wi-fi, TV and I was available.</td>
</tr>
<tr>
<td>3</td>
<td>Church looked When morning awake open the curtain.</td>
</tr>
<tr>
<td>4</td>
<td>When looking at often, wives, went out and is invited to try to go […].</td>
</tr>
<tr>
<td>5</td>
<td>But was a little cold, morning walks was good.</td>
</tr>
</tbody>
</table>
The room it is not narrowly was a simple, bathtub was also attached.

Wi-fi, TV and I was available.

Church looked When morning awake open the curtain.

When looking at often, wives, went out and is invited to try to go [...].

But was a little cold, morning walks was good.
Machine Translation

Short History:

today

Reformation
phrase-based and syntax-based systems
statistical approach (cheap, automatically trained)

1991

Dark age
rule-based systems (e.g., SYSTRAN)
Chomskyan approach (perfect translation, poor coverage)

1960
Machine Translation

Vauquois triangle:

Translation model: string-to-string
Machine Translation

Vauquois triangle:

Translation model: **string-to-tree**
Machine Translation

Vauquois triangle:

Translation model: tree-to-tree
parallel corpus, word alignments, parse tree
parallel corpus, word alignments, parse tree

I would like your advice about Rule 143 concerning inadmissibility.
Synchronous tree substitution grammar: productions $N \rightarrow (r, r_1)$

- nonterminal $N$
- right-hand side $r$ of context-free grammar production
- right-hand side $r_1$ of tree substitution grammar production

Weighted Synchronous Grammars

Synchronous tree substitution grammar: productions $N \rightarrow (r, r_1)$

- nonterminal $N$
- right-hand side $r$ of context-free grammar production
- right-hand side $r_1$ of tree substitution grammar production
- (bijective) synchronization of nonterminals

Production application:

- Selection of synchronous nonterminals
Synchronous Grammars

Production application:

- Selection of synchronous nonterminals
Synchronous Grammars

Production application:

1. Selection of synchronous nonterminals
2. Selection of suitable production
Synchronous Grammars

Production application:
1. Selection of synchronous nonterminals
2. Selection of suitable production
3. Replacement on both sides
Synchronous Grammars

Production application:

1 synchronous nonterminals
Synchronous Grammars

Production application:

1 synchronous nonterminals
Synchronous Grammars

Production application:
1. synchronous nonterminals
2. suitable production
Synchronous Grammars

Production application:
1. synchronous nonterminals
2. suitable production
3. replacement
I would like your advice about Rule 143 concerning inadmissibility following [Galley, Hopkins, Knight, Marcu: What’s in a translation rule? Proc. NAACL, 2004]
I would like your advice about Rule 143 concerning inadmissibility.

would like your advice about Rule 143 concerning inadmissibility

would like your advice about Rule 143 concerning inadmissibility

I would like your advice about Rule 143 concerning inadmissibility.

I would like your advice about Rule 143 concerning inadmissibility.
PPER would like your advice about Rule 143.
Könnten Sie eine Auskunft zu Artikel 143 geben?
PPER would like your advice about Rule 143.
PPER would like your advice about Rule 143.
Könnten Sie eine Auskunft zu Artikel 143 geben?

*Production Extraction*

**Repeated production extraction:**

(extractable productions marked in red)
KÖNNTEN Sie eine Auskunft zu Artikel 143 geben.
PER would like your advice about Rule 143.

Könnten Sie eine Auskunft zu Artikel 143 geben?
Synchronous Tree Substitution Grammars

Advantages:

- very simple
- implemented in framework ‘Moses’
- “context-free”
Synchronous Tree Substitution Grammars

Advantages:

- very simple
- implemented in framework ‘Moses’
- “context-free”

Disadvantages:

- problems with discontinuities
- composition and binarization not possible
- “context-free”
## Evaluation

### English → German translation task:

(higher BLEU is better)

<table>
<thead>
<tr>
<th>Type</th>
<th>System</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>vanilla WMT 2013</td>
<td>WMT 2015</td>
</tr>
<tr>
<td>string-to-tree</td>
<td>STSG</td>
<td>15.2</td>
</tr>
<tr>
<td>tree-to-tree</td>
<td>STSG</td>
<td>14.5</td>
</tr>
</tbody>
</table>

STSG = synchronous tree substitution grammar

# Evaluation

## English → German translation task:

(higher BLEU is better)

<table>
<thead>
<tr>
<th>Type</th>
<th>System</th>
<th>BLEU</th>
<th>WMT 2013</th>
<th>WMT 2015</th>
</tr>
</thead>
<tbody>
<tr>
<td>string-to-tree</td>
<td>STSG</td>
<td>15.2</td>
<td>19.4</td>
<td>24.5</td>
</tr>
<tr>
<td>tree-to-tree</td>
<td>STSG</td>
<td>14.5</td>
<td>—</td>
<td>15.3</td>
</tr>
</tbody>
</table>

STSG = synchronous tree substitution grammar

**Observations:**

- syntax-based systems competitive with manual adjustments
- much less so for vanilla systems
- very unfortunate situation (more supervision yields lower scores)

---


very specific production

every production for ‘advice’ contains sentence structure

(syntax “in the way”)
Synchronous Grammars

Synchronous multi tree substitution grammar: \( N \rightarrow (r, \langle r_1, \ldots, r_n \rangle) \)


- nonterminal \( N \)
- right-hand side \( r \) of context-free grammar production
- right-hand sides \( r_1, \ldots, r_n \) of regular tree grammar production
Synchronous multi tree substitution grammar: $N \rightarrow (r, \langle r_1, \ldots, r_n \rangle)$

- nonterminal $N$
- right-hand side $r$ of context-free grammar production
- right-hand sides $r_1, \ldots, r_n$ of regular tree grammar production

ART-NN-VV $\rightarrow$

- eine
- Auskunft
- geben

ART

NN

VV
**Synchronous Grammars**

**Synchronous multi tree substitution grammar:** \( N \rightarrow (r, \langle r_1, \ldots, r_n \rangle) \)

- nonterminal \( N \)
- right-hand side \( r \) of context-free grammar production
- right-hand sides \( r_1, \ldots, r_n \) of regular tree grammar production

---

\[
\text{ART-NN-VV} \rightarrow \\
\quad \text{eine} \quad \text{Auskunft} \quad \text{geben} \\
\quad \text{ART} \quad \text{NN} \quad \text{VV}
\]
Synchronous Grammars

Synchronous multi tree substitution grammar: \( N \rightarrow (r, \langle r_1, \ldots, r_n \rangle) \)

- nonterminal \( N \)
- right-hand side \( r \) of context-free grammar production
- right-hand sides \( r_1, \ldots, r_n \) of regular tree grammar production

Synchronous Grammars

Synchronous multi tree substitution grammar: \( N \rightarrow (r, \langle r_1, \ldots, r_n \rangle) \)


- nonterminal \( N \)
- right-hand side \( r \) of context-free grammar production
- right-hand sides \( r_1, \ldots, r_n \) of regular tree grammar production
- synchronization via map \( NT \ r_1, \ldots, r_n \) to \( NT \ r \)
Synchronous Grammars

Production application:

- synchronous nonterminals
Production application:
- synchronous nonterminals
Synchronous Grammars

Production application:

1. Synchronous nonterminals
2. Suitable production
Production application:

1. synchronous nonterminals
2. suitable production
3. replacement
PPER would like your advice about Rule 143

variant of [M.: How to train your multi bottom-up tree transducer. Proc. ACL, 2011]
PPER would like your advice about Rule 143.

variant of [M.: How to train your multi bottom-up tree transducer. Proc. ACL, 2011]
Könnten Sie eine Auskunft zu Artikel 143 geben?

variant of [M.: How to train your multi bottom-up tree transducer. Proc. ACL, 2011]
PPER would like your advice about Rule 143

variant of [M.: How to train your multi bottom-up tree transducer. Proc. ACL, 2011]
Advantages:

- complicated discontinuities
- implemented in framework ‘Moses’
- binarizable, composable
Synchronous Multi Tree Substitution Grammars

Advantages:

- complicated discontinuities
- implemented in framework ‘Moses’
- binarizable, composable

Disadvantages:

- output non-regular (tree-level) or non-context-free (string-level)
  (in fact output is captured by MRTG = MCFTG without variables)
- not symmetric (input context-free; output not)
## Evaluation

<table>
<thead>
<tr>
<th>Task</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>STSG</td>
</tr>
<tr>
<td>English → German</td>
<td>15.0</td>
</tr>
<tr>
<td>English → Arabic</td>
<td>48.2</td>
</tr>
<tr>
<td>English → Chinese</td>
<td>17.7</td>
</tr>
<tr>
<td>English → Polish</td>
<td>21.3</td>
</tr>
<tr>
<td>English → Russian</td>
<td>24.7</td>
</tr>
</tbody>
</table>

STSG = synchronous tree substitution grammar
SMTSG = synchronous multi tree substitution grammar

### Evaluation

<table>
<thead>
<tr>
<th>Task</th>
<th>BLEU</th>
<th>Productions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>STSG</td>
<td>SMTSG</td>
</tr>
<tr>
<td>English → German</td>
<td>15.0</td>
<td>*15.5</td>
</tr>
<tr>
<td>English → Arabic</td>
<td>48.2</td>
<td>*49.1</td>
</tr>
<tr>
<td>English → Chinese</td>
<td>17.7</td>
<td>*18.4</td>
</tr>
<tr>
<td>English → Polish</td>
<td>21.3</td>
<td>*23.4</td>
</tr>
<tr>
<td>English → Russian</td>
<td>24.7</td>
<td>*26.1</td>
</tr>
</tbody>
</table>

STSG = synchronous tree substitution grammar
SMTSG = synchronous multi tree substitution grammar

Observations:
- Consistent improvements
- 1 magnitude more productions
- SMTSG alleviate some of the problems of syntax-based systems

### Evaluation

<table>
<thead>
<tr>
<th>Task</th>
<th>BLEU</th>
<th>Productions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>STSG</td>
<td>SMTSG</td>
</tr>
<tr>
<td>English → German</td>
<td>15.0</td>
<td>*15.5</td>
</tr>
<tr>
<td>English → Arabic</td>
<td>48.2</td>
<td>*49.1</td>
</tr>
<tr>
<td>English → Chinese</td>
<td>17.7</td>
<td>*18.4</td>
</tr>
<tr>
<td>English → Polish</td>
<td>21.3</td>
<td>*23.4</td>
</tr>
<tr>
<td>English → Russian</td>
<td>24.7</td>
<td>*26.1</td>
</tr>
<tr>
<td></td>
<td>14M</td>
<td>144M</td>
</tr>
<tr>
<td></td>
<td>55M</td>
<td>491M</td>
</tr>
<tr>
<td></td>
<td>17M</td>
<td>162M</td>
</tr>
<tr>
<td></td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

**STSG** = synchronous tree substitution grammar  
**SMTSG** = synchronous multi tree substitution grammar

**Observations:**

- consistent improvements
- 1 magnitude more productions
- SMTSG alleviate some of the problems of syntax-based systems

Synchronous Grammars

Notes:
- tree-to-tree models easier for theoretical investigation
- strongly related to tree transducers
- we disallow trivial input sides of just a nonterminal ($\varepsilon$-free)

Synchronous grammar:

Tree transducer:
Synchronous Grammars

Major linear tree transducers:

<table>
<thead>
<tr>
<th>input sides</th>
<th>synchronization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>bijective</td>
</tr>
<tr>
<td>shallow</td>
<td>nondeleting top-down</td>
</tr>
<tr>
<td>general</td>
<td>nondeleting extended</td>
</tr>
</tbody>
</table>

Further distinction:

- allow productions on disconnected input nonterminals
  → regular look-ahead
- allow arbitrary trees for disconnected input nonterminals
  → no look-ahead
Synchronous Grammars

Evaluation properties:

- rotations implementable?
- symmetric?
- domain regular?
- range regular?
- closed under composition?

(for arbitrary $t_1, t_2, t_3$)


Icons by interactivemania (http://www.interactivemania.com/) and UN Office for the Coordination of Humanitarian Affairs
Synchronous Grammars

Illustration of rotation:

S
  NP
    NNP Alice
    VBD carries
  VP
    NNP Bob

S
  NP
    NNP Bob
  VP
    VBZ is
      VBN carried
    NP
      IN by
      NNP Alice

→
Top-down Tree Transducer

Hasse diagram:

Model |
-------|
ns-TOP | X | X | ✓ | ✓ | ✓ | ✓ |
n-TOP  | X | X | ✓ | ✓ | ✓ | ✓ |
s-TOP  | X | X | ✓ | ✓ | ✓ | X₂ |
s-TOP^R| X | X | ✓ | ✓ | ✓ | ✓ |
TOP    | X | X | ✓ | ✓ | ✓ | X₂ |
TOP^R  | X | X | ✓ | ✓ | ✓ | ✓ |

(composition closure in subscript)
Synchronous Tree Substitution Grammars

Hasse diagram:

<table>
<thead>
<tr>
<th>Model</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>n-TOP</td>
<td>✗ ✗ ✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>TOP</td>
<td>✗ ✗ ✓ ✓ ✗ 2</td>
</tr>
<tr>
<td>TOP R</td>
<td>✗ ✗ ✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>ns-STSG</td>
<td>✓ ✓ ✓ ✓ ✗ 2</td>
</tr>
<tr>
<td>n-STSG</td>
<td>✓ ✗ ✓ ✓ ✗ ∞</td>
</tr>
<tr>
<td>s-STSG R</td>
<td>✓ ✗ ✓ ✓ ✗ 2</td>
</tr>
<tr>
<td>STSG</td>
<td>✓ ✗ ✓ ✓ ✗ 4</td>
</tr>
<tr>
<td>STSG R</td>
<td>✓ ✗ ✓ ✓ ✗ 3</td>
</tr>
</tbody>
</table>

(composition closure in subscript)

composition closures by

Advantages of SMTSG

- always have regular look-ahead
- can always be made nondeleting & shallow
- closed under composition

Advantages of SMTSG

- always have regular look-ahead
- can always be made nondeleting & shallow
- closed under composition

Disadvantages of SMTSG:

- non-regular range

(technically interesting?)

Synchronous Multi Tree Substitution Grammars

Hasse diagram:

<table>
<thead>
<tr>
<th>Model</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>n-TOP</td>
<td>✗</td>
</tr>
<tr>
<td>TOP</td>
<td>✗</td>
</tr>
<tr>
<td>TOP(^R)</td>
<td>✗</td>
</tr>
<tr>
<td>ns-STSG</td>
<td>✓</td>
</tr>
<tr>
<td>n-STSG</td>
<td>✗</td>
</tr>
<tr>
<td>s-STSG(^R)</td>
<td>✗</td>
</tr>
<tr>
<td>STSG</td>
<td>✗</td>
</tr>
<tr>
<td>STSG(^R)</td>
<td>✗</td>
</tr>
<tr>
<td>SMTSG</td>
<td>✗</td>
</tr>
<tr>
<td>reg. range</td>
<td>✗</td>
</tr>
<tr>
<td>symmetric</td>
<td>✓</td>
</tr>
</tbody>
</table>

(string-level) range characterization by
Synchronous Multi Tree Substitution Grammars

Theorem

$$(\mathit{STSG}^R)^3 \subsetneq \text{reg.-range SMTSG}$$
Summary

Parsing:

- tree automata = CFG with subcategorization (which are the state-of-the-art models for many languages)
- wealth of open problems for non-constituent parsing (alternative theories seem to be on the rise; “Parsey McParseface”)
Summary

Parsing:

- tree automata = CFG with subcategorization (which are the state-of-the-art models for many languages)
- wealth of open problems for non-constituent parsing (alternative theories seem to be on the rise; "Parsey McParseface")

Machine translation:

- all major translation models in use are grammar-based (and their expressive power is often ill-understood)
- combination of parser and translation model challenging (although that is typically just a regular domain restriction)
- evaluation of theoretically well-behaved models (in practice)

Thank you for the attention.
Summary

Parsing:

- tree automata = CFG with subcategorization (which are the state-of-the-art models for many languages)
- wealth of open problems for non-constituent parsing (alternative theories seem to be on the rise; “Parsey McParseface”)

Machine translation:

- all major translation models in use are grammar-based (and their expressive power is often ill-understood)
- combination of parser and translation model challenging (although that is typically just a regular domain restriction)
- evaluation of theoretically well-behaved models (in practice)

Thank you for the attention.