Tree Automata in Parsing and Machine Translation

Andreas Maletti

Institute of Computer Science
Universität Leipzig, Germany

Lille — November 15, 2018
Parsing

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

determining the syntactic structure of a sentence

subject to a given theory of syntax (encoded in the training data)

- constituent syntax
- dependency syntax
- ...

SNP-SBJNPNNPMr.NNPHahn,, NPNPDTthe ADJPNPCD62HYPH- NNyear
HHYPH- JJold
NMLNMLNNchairman
and NMLJJchiefJJexecutiveNNofficer
PPINof NPNNPGeorgiaHYPH- NNPPacificNNPCorp.

VPVBZis VPVBGleading NPNPNPDTthe NMLNNforestHYPH- NNPPacificCorporation.

The unsolicited,$CD3.19$-NONE-*U*-NNbid for NPNNPGreatNNPNorthernNNPNekoosaNNPCorp.
Example: We must bear in mind the Community as a whole

S
  NP
    PRP  MD
      We  must
  VP
    VB  PP
      bear  in
        NP
          NP  DT
            mind  the  Community
  NP
    NP  IN  NP  PP
      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

Berkeley parser:

```
S
  NP
    PRP  MD  VB
      We  must  bear
    IN  NP
      in  NN
    NP
      DT  NN
        the  Community
    PP
      IN  NP
        as  DT  NN
          a  whole
```

BLLIP parser:

```
S
  NP
    PRP  MD  VB
      We  must  bear
    IN  NP
      in  NN
    NP
      DT  NNP
        the  Community
    PP
      IN  NP
        as  DT  NN
          a  whole
```
Constituent Parsing

Today

Linear-time dependency models; optimized by neural networks

2016

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

| grammar | $F_1$-score | \( |w| \leq 40 \) | full |
|---------|-------------|----------------|------|
| CFG     |             | 62.7           |      |
| TSG [Post, Gildea, 2009] | | 82.6           |      |
| TSG [Cohn et al., 2010] | | 85.4           | 84.7 |
| CFG\textsubscript{sub} [Collins, 1999] | | 88.6           | 88.2 |
| CFG\textsubscript{sub} [Petrov, Klein, 2007] | | 90.6           | 90.1 |
| CFG\textsubscript{sub} [Petrov, 2010] | |               | 91.8 |
| TSG\textsubscript{sub} [Shindo et al., 2012] | | 92.9           | 92.4 |
All models use weights for disambiguation:
Subcategorization

Tags:

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

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

```
S-1
  NP-4
    PRP$-3
      My
    NN-2
      dog
  VP-5
    VBZ-7
      sleeps
```
Subcategorization

Tags:

- official tags often conservative
  - **English**: ~ 50 tags
  - **German**: ≫ 200 tags
- all modern parsers use refined tags → subcategorization
- but return parse over official tags → relabeling

```
My dog sleeps
```
Subcategorization

These CFG derivations

also admit
Constituent Parsing

Read off CFG productions:

\[
\begin{align*}
S & \rightarrow NP \ VP \\
PRP$ & \rightarrow My \\
VP & \rightarrow VBZ \\
NP & \rightarrow PRP \\
VP & \rightarrow VBD \ ADVP \\
ADVP & \rightarrow RB
\end{align*}
\]

\[
\begin{align*}
NP & \rightarrow PRP$ \ NN \\
NN & \rightarrow dog \\
VBZ & \rightarrow sleeps \\
PRP & \rightarrow I \\
VBD & \rightarrow scored \\
RB & \rightarrow well
\end{align*}
\]
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

\[
\begin{align*}
q_4 & \rightarrow q_5 \quad \text{VP} \quad q_3 \\
q_0 & \rightarrow q_1 \quad S \quad q_4 \\
q_0 & \rightarrow q_6 \quad S \quad q_2
\end{align*}
\]
Let \((Q, \Sigma, I, R)\) tree automaton

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

- recognized tree language

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

| grammar          | $|w| \leq 40$ | full     |
|------------------|-------------|----------|
| CFG              |             | 62.7     |
| TSG [Post, Gildea, 2009] | 82.6       |          |
| TSG [Cohn et al., 2010]  | 85.4       | 84.7     |
| CFG sub [Collins, 1999] | 88.6       | 88.2     |
| CFG sub [Petrov, Klein, 2007] | 90.6       | 90.1     |
| CFG sub [Petrov, 2010]     |             | 91.8     |
| TSG sub [Shindo et al., 2012] | 92.9       | 92.4     |
Constituent Parsing

| grammar               | $|w| \leq 40$ | full   |
|-----------------------|--------------|--------|
| CFG                   | 62.7         |        |
| TSG [Post, Gildea, 2009] | 82.6         |        |
| TSG [Cohn et al., 2010] | 85.4 84.7    |        |
| CFG_{sub} [Collins, 1999] | 88.6 88.2    |        |
| CFG_{sub} [Petrov, Klein, 2007] | 90.6 90.1    |        |
| CFG_{sub} [Petrov, 2010] | 91.8         |        |
| TSG_{sub} [Shindo et al., 2012] | 92.9 92.4    |        |
### Constituent Parsing

| Grammar        | $|w| \leq 40$ | Full      |
|----------------|-------------|-----------|
| CFG            | 62.7        |           |
| TSG [Post, Gildea, 2009] | 82.6        |           |
| TSG [Cohn et al., 2010] | 85.4        | 84.7      |
| CFG sub [Collins, 1999] | 88.6        | 88.2      |
| CFG sub [Petrov, Klein, 2007] | 90.6        | 90.1      |
| CFG sub [Petrov, 2010] | 91.8        |           |
| TSG 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

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

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
  - ...

John saw a dog yesterday which was a Yorkshire Terrier
Dependency Parsing

Illustration page-number:

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.

Practical results:
- Linear-time statistical parsers
- Google's "Parsey McParseface" [Andor et al., 2016]
  - 94% F1-score; linguists achieve 96–97%
Dependency Parsing

Illustration page-number:

Practical results:

- linear-time statistical parsers
- Google’s “Parsey McParseface”
  94% $F_1$-score; linguists achieve 96–97%

[Andor et al., 2016]
Given edge-weighted directed graph, extract “best” edge cover

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

\[
\begin{align*}
L(c) &= \{C\} \\
L(d) &= \{D/E\backslash C, \ D/E/D\backslash C\} \\
L(e) &= \{E\}
\end{align*}
\]
Theoretical problems

Under a suitable relabeling, characterize the set of valid proof trees:

- for just applications → sub-regular tree languages
- for compositions of order 1 → open (probably still regular)
- for compositions of order $k \geq 2$ → open
- for arbitrary compositions → 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]
Multiple context-free tree grammar:
Lexicalization

Derivation tree and evaluation:

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
1. The room it is not narrowly was a simple, bathtub was also attached.
2. Wi-fi, TV and I was available.
3. Church looked When morning awake open the curtain.
4. But was a little cold, morning walks was good.
1. The room it is not narrowly was a simple, bathtub was also attached.
2. Wi-fi, TV and I was available.
3. Church looked When morning awake open the curtain.
4. But was a little cold, morning walks was good.

Original [Japanese — © TripAdvisor]}

1. 部屋もシンプルでしたが狭くなく、バスタブもついていました。
2. Wi-fi、テレビも利用出来ました。
3. 朝起きてカーテンを開けると教会が見えました。
4. ちょっと寒かったけれど、朝の散策はグッドでしたよ。
The room was simple, but it was not small, and the bathtub was also attached.

Wi-fi, TV was also available.

When I woke up in the morning and opened the curtain, I saw the church.

It was a bit cold, but walking in the morning was good.
Machine Translation

Short History:

Today
- Neural networks

2016
- 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
I would like your advice about Rule 143 concerning inadmissibility.
parallel corpus, word alignments, parse tree

via GIZA++ [Och, Ney: A systematic comparison of various statistical alignment models. Computational Linguistics 29(1), 2003]
I would like your advice about Rule 143 concerning inadmissibility.

Könnten Sie mir eine Auskunft zu Artikel 143 im Zusammenhang mit der Unzulässigkeit geben?

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

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
Production application:

- Selection of synchronous nonterminals
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:

- synchronous nonterminals
Production application:

1. synchronous nonterminals
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 following [Galley, Hopkins, Knight, Marcu: What’s in a translation rule? Proc. NAACL, 2004]
I would like your advice about Rule 143 concerning inadmissibility of
Könnten Sie mir eine Auskunft zu Artikel 143 im Zusammenhang mit der Unzulässigkeit geben?

I would like your advice about Rule 143 concerning inadmissibility of

would like your advice about Rule 143 concerning inadmissibility following [Galley, Hopkins, Knight, Marcu: What’s in a translation rule? Proc. NAACL, 2004]
Removal of extractable production:

I would like your advice about Rule 143 concerning inadmissibility.
Removal of extractable production:

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

Would you like your advice about Rule 143?
Would you like your advice about Rule 143?
PPER would like your advice about Rule 143

Könnten Sie eine Auskunft zu Artikel 143 geben
PPER would like your advice about Rule 143

Könnten Sie eine Auskunft zu Artikel 143 geben

PPER PPER PPER would like your advice about Rule 143

PPER PPER PPER Eine Auskunft zu Artikel 143 geben

KOUS PPER ART NN APPR NN CD geben

PP S

NP

PP

PP

PP

PP

VV
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 vanilla</th>
<th>BLEU WMT 2013</th>
<th>BLEU WMT 2015</th>
</tr>
</thead>
<tbody>
<tr>
<td>string-to-string</td>
<td>FST</td>
<td>16.8</td>
<td>20.3</td>
<td>25.2</td>
</tr>
<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

---

**Evaluation**

English → German translation task: (higher BLEU is better)

<table>
<thead>
<tr>
<th>Type</th>
<th>System</th>
<th>BLEU vanilla</th>
<th>BLEU WMT 2013</th>
<th>BLEU WMT 2015</th>
</tr>
</thead>
<tbody>
<tr>
<td>string-to-string</td>
<td>FST</td>
<td>16.8</td>
<td>20.3</td>
<td>25.2</td>
</tr>
<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
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


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

\[
\text{ART} \quad \text{NN} \quad \text{VV}
\]
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

ART-NN-VV →

NP-VV →

NP  ART  APPR  PP  VV

ART  NN  zu  Artikel  143  PP

ART-NN-VV →

NP  ART  NN  CD  PP  VV

eine  Auskunft  geben

about  Rule  143  PP

ART  NN  VV  geben

zu  Artikel  143  PP
Synchronous Grammars

Production application:

1. synchronous nonterminals
2. suitable production
3. replacement
PPER would like your advice about Rule 143.
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?
Synchronous Multi Tree Substitution Grammars

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

**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/](http://www.interactivemania.com/)) and UN Office for the Coordination of Humanitarian Affairs
Illustration of rotation:
Hasse diagram:

```
    ┌───┐
   ┤ TOP1 ┤
   └───┘
     │    │
     │    │
     s-TOP1

    ┌───┐
   ┤ TOP2 ┤
   └───┘
     │    │
     │    │
     s-TOP2

      ┌───┐
     ┤ ns-TOP1(R) ┤
     └───┘
       ┌───┐
      ┤ n-TOP1(R) ┤
      └───┘
        ┌───┐
        s-TOP2(R)
        └───┘
```

(composition closure in subscript)

<table>
<thead>
<tr>
<th>Model</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>ns-TOP</td>
<td>X X ✓ ✓ ✓</td>
</tr>
<tr>
<td>n-TOP</td>
<td>X X ✓ ✓ ✓</td>
</tr>
<tr>
<td>s-TOP</td>
<td>X X ✓ ✓ X2</td>
</tr>
<tr>
<td>s-TOPR</td>
<td>X X ✓ ✓ ✓</td>
</tr>
<tr>
<td>TOP</td>
<td>X X ✓ ✓ X2</td>
</tr>
<tr>
<td>TOPR</td>
<td>X X ✓ ✓ ✓</td>
</tr>
</tbody>
</table>
Synchronous Tree Substitution Grammars

Hasse diagram:

<table>
<thead>
<tr>
<th>Model</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>n-TOP</td>
<td>X X ✓ ✓ ✓ ✓</td>
</tr>
<tr>
<td>TOP</td>
<td>X X ✓ ✓ X₂</td>
</tr>
<tr>
<td>TOPR</td>
<td>X X ✓ ✓ ✓</td>
</tr>
<tr>
<td>ns-STSG</td>
<td>✓ ✓ ✓ ✓ X₂</td>
</tr>
<tr>
<td>n-STSG</td>
<td>✓ X ✓ ✓ X₄</td>
</tr>
<tr>
<td>s-STSG</td>
<td>R</td>
</tr>
<tr>
<td>STSG</td>
<td>✓ X ✓ ✓ X₃</td>
</tr>
<tr>
<td>STSGR</td>
<td>✓ X ✓ ✓ X₃</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

**Synchronous Multi Tree Substitution Grammars**

**Advantages of SMTSG**
- always have regular look-ahead
- can always be made nondeleting & shallow
- closed under composition

**Disadvantages of SMTSG:**
- non-regular range (theoretically interesting?)

Synchronous Multi Tree Substitution Grammars

Hasse diagram:

Model                  Property

n-TOP                  ✓   ✓   ✓   ✓   ✓   ✓
TOP                    ✓   ✓   ✓   ✓   ✓   X₂
TOP⁰                   ✓   ✓   ✓   ✓   ✓   ✓
s-STSG                 ✓   ✓   ✓   ✓   ✓   X₂
ns-STSG               ✓   X   ✓   ✓   ✓   X∞
STSG                   ✓   X   ✓   ✓   ✓   X₂
STSG⁰                  ✓   X   ✓   ✓   ✓   X₄
SMTSG                  ✓   X   ✓   ✓   ✓   ✓
reg. range             ✓   X   ✓   ✓   ✓   ✓
symmetric             ✓   ✓   ✓   ✓   ✓   ✓

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

Theorem

$$\left(\text{STSG}^\mathbb{R}\right)^3 \subsetneq \text{reg.-range SMTSG}$$

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”)

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)
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.