Tree Automata in Parsing and Machine Translation

Andreas Maletti

Institute of Computer Science
Universität Leipzig, Germany

Leipzig — April 20, 2017
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.Hahn,, NPNPDTthe ADJPNPCD62HYPH- NNyear HYPH- JJold NMLNMLNNchairman and NMLJJchiefJJexecutiveNNofficer PPINof NPNNPGeorgiaHYPH- NNPPacificNNPCorp. VPVBZis VPVBGleading NPNPNPDTthe NMLNNforestHYPH- NNPPin product concern POS's JJunsolicitedNMLQP$$CD3.19CDbillion -NONE-*U* PPINfor NPNNPGreatNNPNorthernNNPNekoosaNNPCorp.

Parsing
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*

**Constituent Parsing**

**POS-tag:** part-of-speech tag, “class” of a word
Berkeley parser:

BLLIP parser:
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

<table>
<thead>
<tr>
<th>grammar</th>
<th>$F_1$-score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$</td>
</tr>
<tr>
<td>CFG</td>
<td></td>
</tr>
<tr>
<td>TSG [Post, Gildea, 2009]</td>
<td>82.6</td>
</tr>
<tr>
<td>TSG [Cohn et al., 2010]</td>
<td>85.4</td>
</tr>
<tr>
<td>CFG$_{sub}$ [Collins, 1999]</td>
<td>88.6</td>
</tr>
<tr>
<td>CFG$_{sub}$ [Petrov, Klein, 2007]</td>
<td>90.6</td>
</tr>
<tr>
<td>CFG$_{sub}$ [Petrov, 2010]</td>
<td></td>
</tr>
<tr>
<td>TSG$_{sub}$ [Shindo et al., 2012]</td>
<td>92.9</td>
</tr>
</tbody>
</table>
All models use weights for disambiguation:
Subcategorization

Tags:
- official tags often conservative
  - English: ≈ 50 tags
  - German: ≫ 200 tags

ADJA-Sup-Dat-Sg-Fem
Tags:

- official tags often conservative
  - English: ≈ 50 tags
  - German: ≫ 200 tags
- all modern parsers use refined tags → subcategorization
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

```
S
  /   |
 NP   VP
  / |
PRP$ NN VBZ
  |
 My dog sleeps
```
These CFG derivations

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

also admit

```
NP
PRP   PRP
   I      I
VP
VBZ
sleeps
VBZ
sleep
```
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 (subcategorizations)
- 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**

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

\[
q_0 \rightarrow S
\]
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

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

| Grammar                     | $F_1$-score $|w| \leq 40$ | $F_1$-score 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 sub = finite-state

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

| Grammar          | $F_1$-Score $|w| \leq 40$ | $F_1$-Score 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:

\[
\text{subcategorization} = \frac{\text{finite-state}}{\text{all modern models}} \approx \text{equivalent to tree automata in expressive power}
\]
Constituent Parsing

<table>
<thead>
<tr>
<th>Grammar</th>
<th>$F_1$-score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$</td>
</tr>
<tr>
<td>CFG</td>
<td>62.7</td>
</tr>
<tr>
<td>TSG [Post, Gildea, 2009]</td>
<td>82.6</td>
</tr>
<tr>
<td>TSG [Cohn et al., 2010]</td>
<td>85.4</td>
</tr>
<tr>
<td>CFG$_{sub}$ [Collins, 1999]</td>
<td>88.6</td>
</tr>
<tr>
<td>CFG$_{sub}$ [Petrov, Klein, 2007]</td>
<td>90.6</td>
</tr>
<tr>
<td>CFG$_{sub}$ [Petrov, 2010]</td>
<td></td>
</tr>
<tr>
<td>TSG$_{sub}$ [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

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

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

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
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”
  
  94% $F_1$-score; linguists achieve 96–97%

[Andor et al., 2016]
## 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 $D$

\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 \[\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:
MCFTG strongly lexicalize themselves and inv. of their expressive power
ongoing work with Joost Engelfriet and Sebastian Maneth
Lexicalization

Derivation tree and evaluation:

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>But was a little cold, morning walks was good.</td>
</tr>
</tbody>
</table>
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も、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.

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。

Wi-fi、テレビも利用出来ました。

朝起きてカーテンを開けると教会が見えました。

ちょっと寒かったけれど、朝の散策はグッドでしたよ。
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
parallel corpus, word alignments, parse tree

I would like your advice about Rule 143 concerning inadmissibility.

Könnten Sie mir eine Auskunft zu Artikel 143 im Zusammenhang mit der Unzulässigkeit geben.
I would like your advice about Rule 143 concerning inadmissibility.
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 \( N \to (r, r_1) \)

- nonterminal \( N \)
- right-hand side \( r \) of context-free grammar production
- right-hand side \( r_1 \) of tree substitution grammar production

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

Synchronous Grammars

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

KOUS → would like
Könnten
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
Production application:

- synchronous nonterminals
Synchronous Grammars

Production application:
1. synchronous nonterminals
2. suitable production

S →

Könnten

PPER

would like

PPER

advice

PP

geben

VV

S

KOUS

PPER

ART

NN

NP

PP

→
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]
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.

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

"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)
PPER would 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?
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>WMT 2013</th>
<th>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

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

<table>
<thead>
<tr>
<th>Type</th>
<th>System</th>
<th>BLEU 2013</th>
<th>BLEU 2015</th>
</tr>
</thead>
<tbody>
<tr>
<td>string-to-string</td>
<td>FST</td>
<td>16.8</td>
<td>20.3</td>
</tr>
<tr>
<td>string-to-tree</td>
<td>STSG</td>
<td>15.2</td>
<td>19.4</td>
</tr>
<tr>
<td>tree-to-tree</td>
<td>STSG</td>
<td>14.5</td>
<td>—</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 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

\[
\text{ART-NN-VV} \quad \text{about} \quad \text{Rule 143} \quad \text{PP}
\]

\[
\text{NP-VV} \rightarrow
\]

\[
\text{ART} \quad \text{NN} \quad \text{APPR} \quad \text{NN} \quad \text{CD} \quad \text{PP} \quad \text{VV}
\]

\[
\text{PP} \quad \text{143} \quad \text{zu} \quad \text{Artikel}
\]
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 \( NT r \)
Production application:

- synchronous nonterminals
Synchronous Grammars

Production application:

- synchronous nonterminals
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]
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]
PPER would like your advice about Rule 143

variant of [M.: How to train your multi bottom-up tree transducer. Proc. ACL, 2011]
Synchronous Multi Tree Substitution Grammars

**Advantages:**
- complicated discontinuities
- implemented in framework ‘Moses’
  
  [Braune, Seemann, Quernheim, M.: Shallow local multi bottom-up tree transducers in SMT. *Proc. ACL*, 2013]
- 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></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>STSG</td>
<td>SMTSG</td>
<td></td>
</tr>
<tr>
<td>English → German</td>
<td>15.0</td>
<td>*15.5</td>
<td></td>
</tr>
<tr>
<td>English → Arabic</td>
<td>48.2</td>
<td>*49.1</td>
<td></td>
</tr>
<tr>
<td>English → Chinese</td>
<td>17.7</td>
<td>*18.4</td>
<td></td>
</tr>
<tr>
<td>English → Polish</td>
<td>21.3</td>
<td>*23.4</td>
<td></td>
</tr>
<tr>
<td>English → Russian</td>
<td>24.7</td>
<td>*26.1</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

## 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

### 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

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
Illustration of rotation:
Top-down Tree Transducer

Hasse diagram:

```
            TOP
           /    \
          /     \
         /      \
        /        \n       TOP_2     s-TOP_1
                  /    \    \
                 /      \
                /       \
               /        \
              s-TOP_2   n-TOP_1^R
                        /    \    \
                       /      \
                      /       \
                     /        \
                    ns-TOP_1^R
```

Model Property

<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-TOP^R</td>
<td>X X ✓ ✓ ✓</td>
</tr>
<tr>
<td>TOP</td>
<td>X X ✓ ✓ X2</td>
</tr>
<tr>
<td>TOP^R</td>
<td>X X ✓ ✓ ✓</td>
</tr>
</tbody>
</table>

(composition closure in subscript)
Synchronous Tree Substitution Grammars

Hasse diagram:

Model       Property

<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 ✓ ✓ Xₘ</td>
</tr>
<tr>
<td>s-STSGᴿ</td>
<td>✓ X ✓ ✓ X₂</td>
</tr>
<tr>
<td>STSG</td>
<td>✓ X ✓ ✓ X₄</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

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
TOP⁵ | ✓ ✓ ✓ ✓ ✓ ✓
ns-STSG | ✓ ✓ ✓ ✓ X₂
n-STSG | ✓ ✓ ✓ ✓ X₁
s-STSG⁵ | ✓ ✓ ✓ ✓ X₂
STSG | ✓ ✓ ✓ ✓ X₄
STSG⁵ | ✓ ✓ ✓ ✓ X₃
SMTSG | ✓ ✓ ✓ X ✓ ✓
reg. range | ✓ ✓ ✓ ✓ ✓ ✓
symmetric | ✓ ✓ ✓ ✓ ✓ ✓

(composition closure in subscript)

(string-level) range characterization by

Theorem

\[(\text{STG}^R)^3 \subsetneq \text{reg.-range SMTSG}\]

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”)

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)
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.