Andreas Maletti

Institute of Computer Science
University of Leipzig

Leipzig — December 2, 2016
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

Berkeley parser:

BLLIP parser:
**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

| 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 |      |
All models use weights for disambiguation:
Subcategorization

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

```
NP
PRP$
My

NN
dog

VP

VBZ
sleeps
```

ADJA-Sup-Dat-Sg-Fem
Subcategorization

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

NP-4
   PRP$-3  NN-2
  My  dog

VBZ-7
sleeps

S-1
 NP-4  VP-5

ADJA-Sup-Dat-Sg-Fem
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

also admit
Constituent Parsing

Read off CFG productions:

S → NP VP
PRP$ → My
VP → VBZ
NP → PRP
VP → VBD ADVP
ADVP → RB

NP → PRP$ NN
NN → dog
VBZ → sleeps
PRP → I
VBD → scored
RB → well
Tree Automaton

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

1. $q_4 \rightarrow q_5 \quad q_0 \rightarrow q_1 \quad q_0 \rightarrow q_6$
2. $q_2 \quad q_3 \quad q_4$
3. $q_2$
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

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

<table>
<thead>
<tr>
<th>Grammar</th>
<th>$F_1$-score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$</td>
</tr>
<tr>
<td>CFG</td>
<td>62.7</td>
</tr>
<tr>
<td>TSG [Post, Gildea, 2009]</td>
<td>82.6</td>
</tr>
<tr>
<td>TSG [Cohn et al., 2010]</td>
<td>85.4 84.7</td>
</tr>
<tr>
<td>CFG$_{sub}$ [Collins, 1999]</td>
<td>88.6 88.2</td>
</tr>
<tr>
<td>CFG$_{sub}$ [Petrov, Klein, 2007]</td>
<td>90.6 90.1</td>
</tr>
<tr>
<td>CFG$_{sub}$ [Petrov, 2010]</td>
<td>91.8</td>
</tr>
<tr>
<td>TSG$_{sub}$ [Shindo et al., 2012]</td>
<td>92.9 92.4</td>
</tr>
</tbody>
</table>

Hence:

$$\text{subcategorization} = \frac{1}{f}\text{finite-state}$$

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

<table>
<thead>
<tr>
<th>Grammar</th>
<th>$w \leq 40$</th>
<th>Full</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFG</td>
<td>62.7</td>
<td></td>
</tr>
<tr>
<td>TSG [Post, Gildea, 2009]</td>
<td>82.6</td>
<td></td>
</tr>
<tr>
<td>TSG [Cohn et al., 2010]</td>
<td>85.4</td>
<td>84.7</td>
</tr>
<tr>
<td>CFG$_{sub}$ [Collins, 1999]</td>
<td>88.6</td>
<td>88.2</td>
</tr>
<tr>
<td>CFG$_{sub}$ [Petrov, Klein, 2007]</td>
<td>90.6</td>
<td>90.1</td>
</tr>
<tr>
<td>CFG$_{sub}$ [Petrov, 2010]</td>
<td></td>
<td>91.8</td>
</tr>
<tr>
<td>TSG$_{sub}$ [Shindo et al., 2012]</td>
<td>92.9</td>
<td>92.4</td>
</tr>
</tbody>
</table>

Hence: subcategorization = $/f_{finite-state}$ all modern models 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 84.7</td>
</tr>
</tbody>
</table>
| 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
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” [Andor et al., 2016]

94% $F_1$-score; linguists achieve 96–97%
### Theoretical problems

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 \(k \geq 3\) \[\text{open}\]
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 $D$

$L(c) = \{C\}$
$L(d) = \{D/E\setminus C, D/E/D\setminus C\}$
$L(e) = \{E\}$
### 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:
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
<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>
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. When looking at often, wives, went out and is invited to try to go [...].
5. But was a little cold, morning walks was good.
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**
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 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.
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.

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]
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.
Removal of extractable production:

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 would like your advice about Rule 143.
PPER would like your advice about Rule 143.
PPER would like your advice about Rule 143.

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

Könnten Sie eine Auskunft zu Artikel 143 geben?
Repeat production extraction: (extractable productions marked in red)
**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>
<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

---

### 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></td>
<td>vanilla</td>
</tr>
<tr>
<td>string-to-string</td>
<td>FST</td>
<td>16.8</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

#### 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 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
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
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]
PPER would like your advice about Rule 143

Könnten Sie eine Auskunft zu Artikel 143 geben

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’
  
- 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

---

<table>
<thead>
<tr>
<th>Task</th>
<th>BLEU</th>
<th>Productions</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>STSG</td>
<td>SMTSG</td>
<td>STSG</td>
<td>SMTSG</td>
</tr>
<tr>
<td>English → German</td>
<td>15.0</td>
<td>*15.5</td>
<td>14M</td>
<td>144M</td>
</tr>
<tr>
<td>English → Arabic</td>
<td>48.2</td>
<td>*49.1</td>
<td>55M</td>
<td>491M</td>
</tr>
<tr>
<td>English → Chinese</td>
<td>17.7</td>
<td>*18.4</td>
<td>17M</td>
<td>162M</td>
</tr>
<tr>
<td>English → Polish</td>
<td>21.3</td>
<td>*23.4</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>English → Russian</td>
<td>24.7</td>
<td>*26.1</td>
<td>—</td>
<td>—</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>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>STSG</td>
<td>SMTSG</td>
<td>STSG</td>
<td>SMTSG</td>
</tr>
<tr>
<td>English → German</td>
<td>15.0</td>
<td>*15.5</td>
<td>14M</td>
<td>144M</td>
</tr>
<tr>
<td>English → Arabic</td>
<td>48.2</td>
<td>*49.1</td>
<td>55M</td>
<td>491M</td>
</tr>
<tr>
<td>English → Chinese</td>
<td>17.7</td>
<td>*18.4</td>
<td>17M</td>
<td>162M</td>
</tr>
<tr>
<td>English → Polish</td>
<td>21.3</td>
<td>*23.4</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>English → Russian</td>
<td>24.7</td>
<td>*26.1</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/) and UN Office for the Coordination of Humanitarian Affairs
Synchronous Grammars

Illustration of rotation:

\[
\begin{align*}
S &\rightarrow NP \\
\text{VBD} &\rightarrow carries \\
NP &\rightarrow NNP \\
\text{VBN} &\rightarrow carried \\
\text{IN} &\rightarrow by \\
NP &\rightarrow NNP \\
\text{VBP} &\rightarrow is \\
\text{IN} &\rightarrow by \\
\text{Alice} &\rightarrow NP \\
Bob &\rightarrow NNP \\
\end{align*}
\]
Top-down Tree Transducer

Hasse diagram:

(composition closure in subscript)
Synchronous Tree Substitution Grammars

Hasse diagram:

Model | Property
--- | ---
n-TOP | ✓ ✓ ✓ ✓ ✓ ✓
TOP | ✓ ✓ ✓ ✓ X₂
TOP^R | ✓ ✓ ✓ ✓ ✓
ns-STSG | ✓ ✓ ✓ ✓ X₂
n-STSG | ✓ X ✓ ✓ X∞
s-STSG^R | ✓ X ✓ ✓ X₂
STSG | ✓ X ✓ ✓ X₄
STSG^R | ✓ X ✓ ✓ X₃

(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

(theretically interesting?)

Synchronous Multi Tree Substitution Grammars

Hasse diagram:

Model | Property
--- | ---
n-TOP | x x v v v
TOP | x x v v x 2
TOP R | x x v v v
ns-STSG | v v v v x 2
n-STSG | v x v v x ∞
s-STSG R | v x v v x 2
STSG | v x v v x 4
STSG R | v x v v x 3
SMTSG | v x v v x
reg. range | v x v v v
symmetric | v v v v v

Synchronous Multi Tree Substitution Grammars

Theorem

\[(\text{STSG}^R)^3 \not\subseteq \text{reg.-range SMTSG}\]

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.