Joshua:

Open Source Toolkit for Parsing-based Machine Translation

Zhifei Li, Chris Callison-Burch, Chris Dyer,
Juri Ganitkevitch, Sanjeev Khudanpur, Lane Schwartz,
Wren Thornton, Jonathan Weese, and Omar Zaidan

Highlights

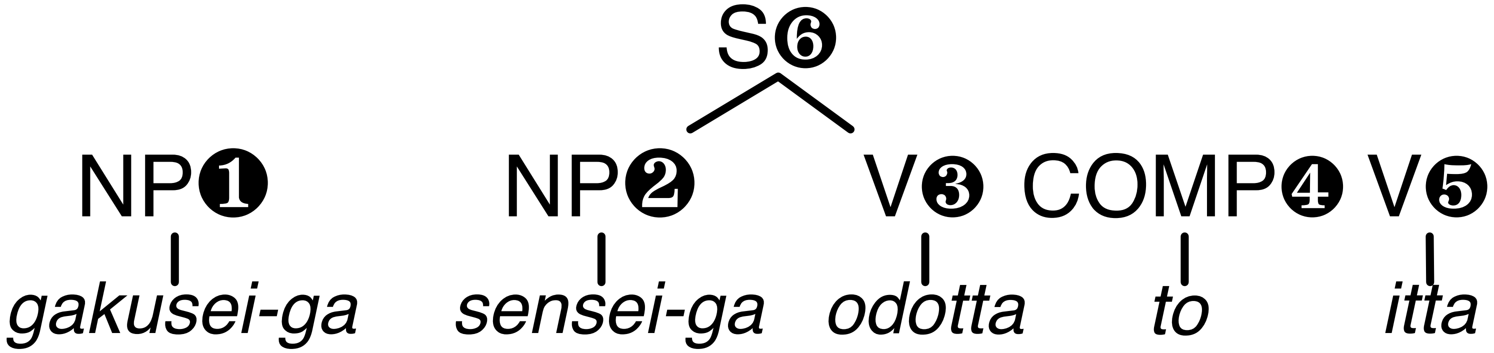
- Fully-featured decoder
  - SCFG decoder in the style of Hiero Chiang (2007)
  - n-gram language model integration
- Attempts to minimize external dependencies
  - Implemented our own MERT and grammar extraction
  - Currently only requires Giza++ and SRILM
- Written in Java
- Goals are to be scalable, easy to extend

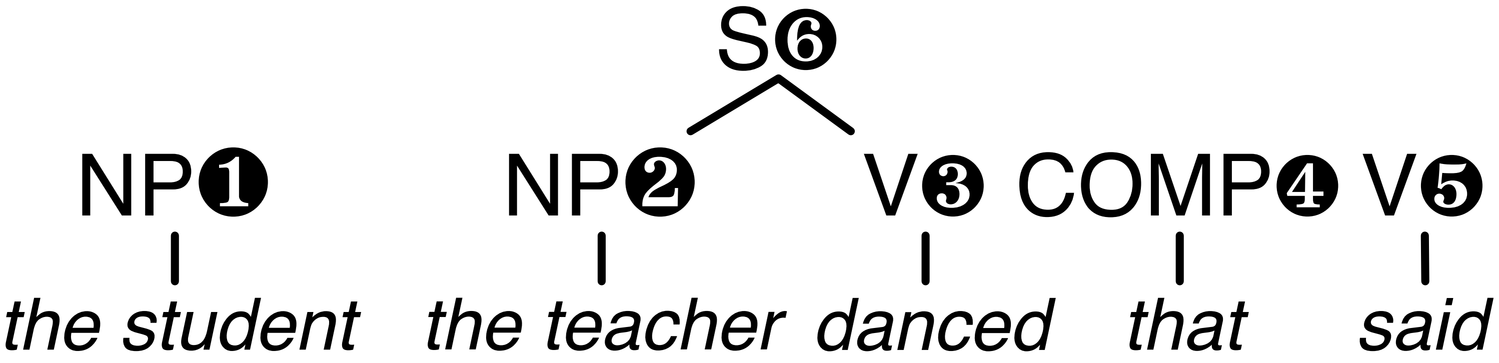
Synchronous CFGs

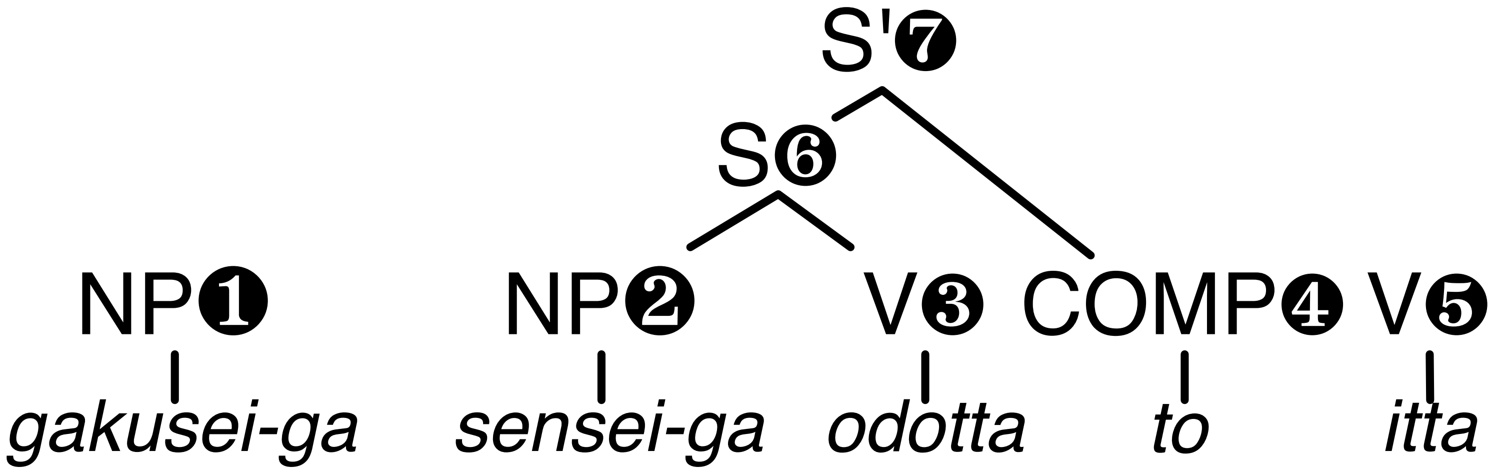
- Generalize context free grammars so they generate pairs of related strings
- Useful for specifying relationship between languages
- Formal definition:
  - T_s: a set of source-language terminal symbols
  - T_t: a set of target-language terminal symbols
  - N: a shared set of nonterminal symbols
  - A set of rules of the form $X \rightarrow \langle \alpha, \beta, \sim, w \rangle$
    - $X \in N$
    - α is a sequence source terminals and non-terminals
    - β is a sequence of target terminals and non-terminals
    - ~ is a one-to-one correspondence between the non-terminals
    - w is a weight or probability assigned to the rule

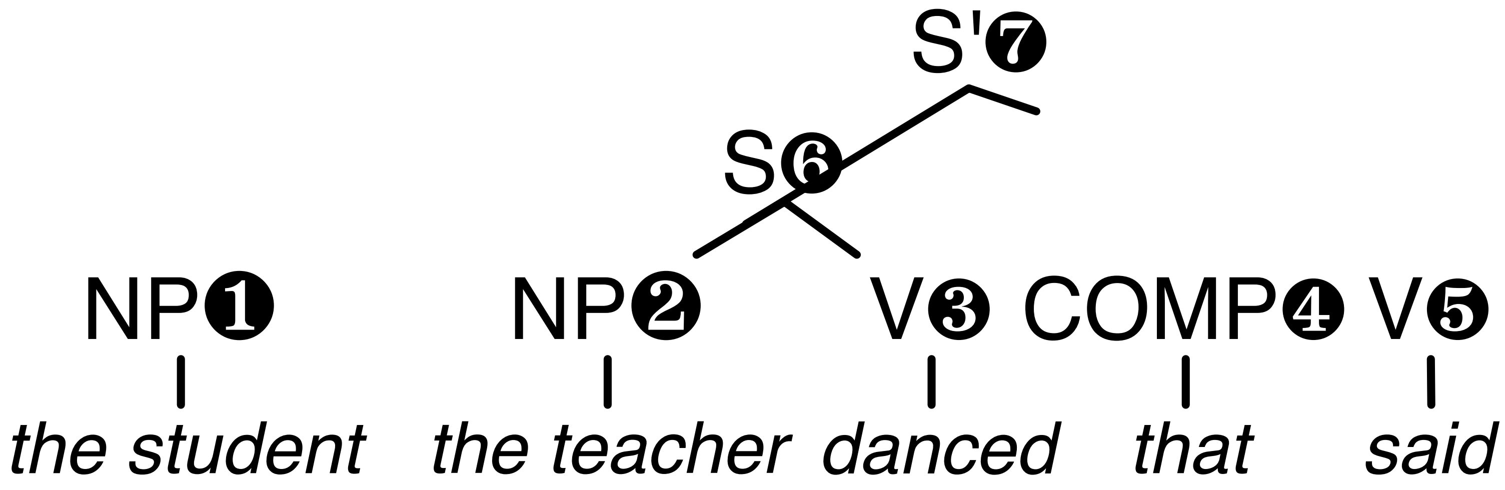
Example SCFG

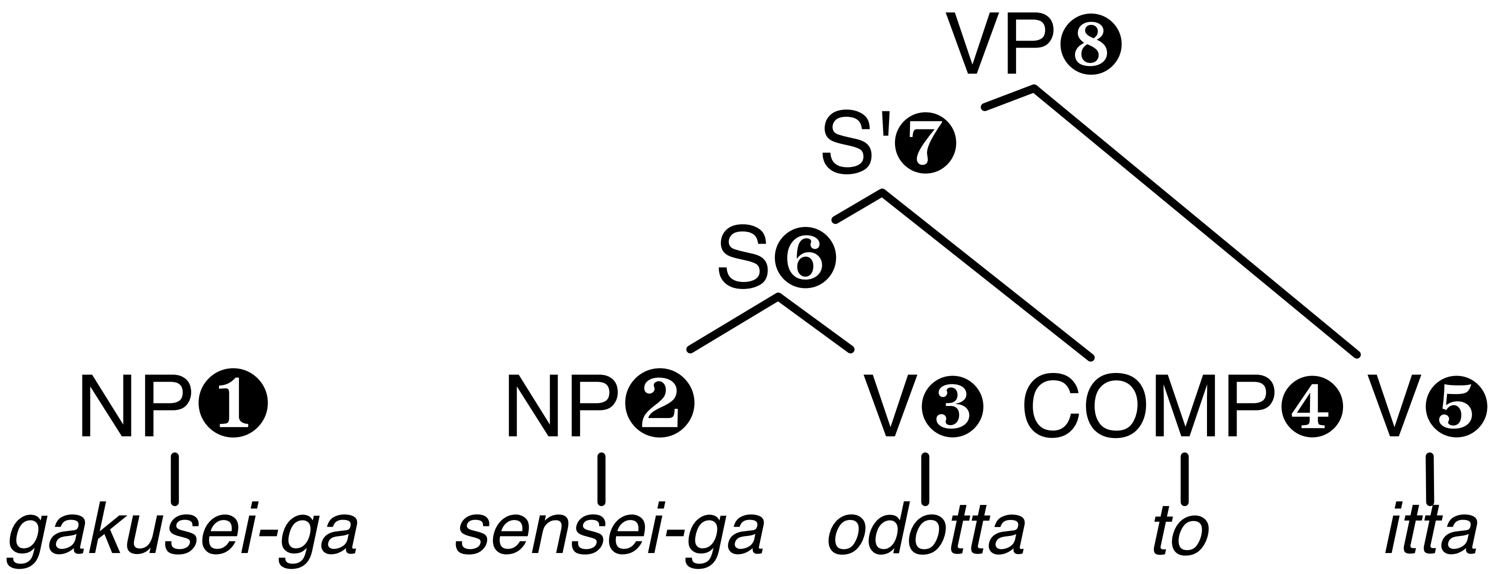
	Japanese	English
S →	NP① VP②	NP① VP②
S'→	S① COMP②	COMP② S①
VP →	NP① V②	V② NP①
NP →	*gakusei-ga*	*student*
NP →	*sensei-ga*	*teacher*
V →	*odotta*	*danced*
V →	*itta*	*said*
COMP →	*to*	*that*

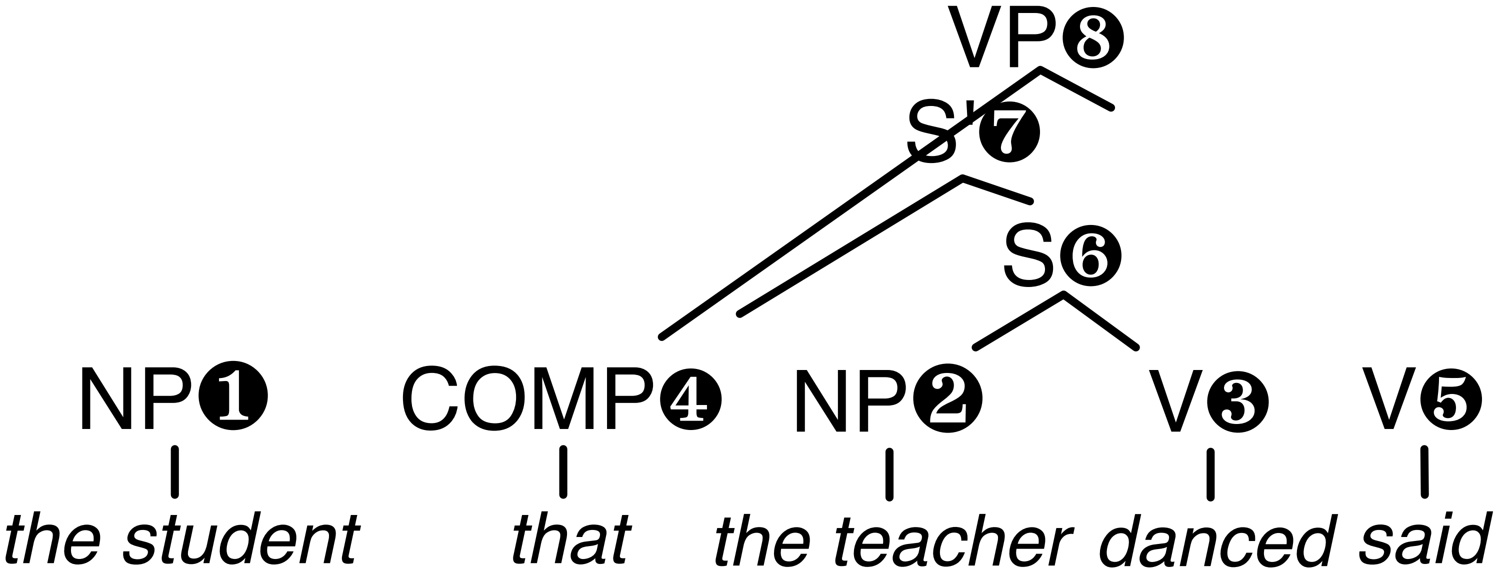

S❻
NP❶
NP❷
V❸
COMP❹
V❺
gakusei-ga
sensei-ga
odotta
to
itta

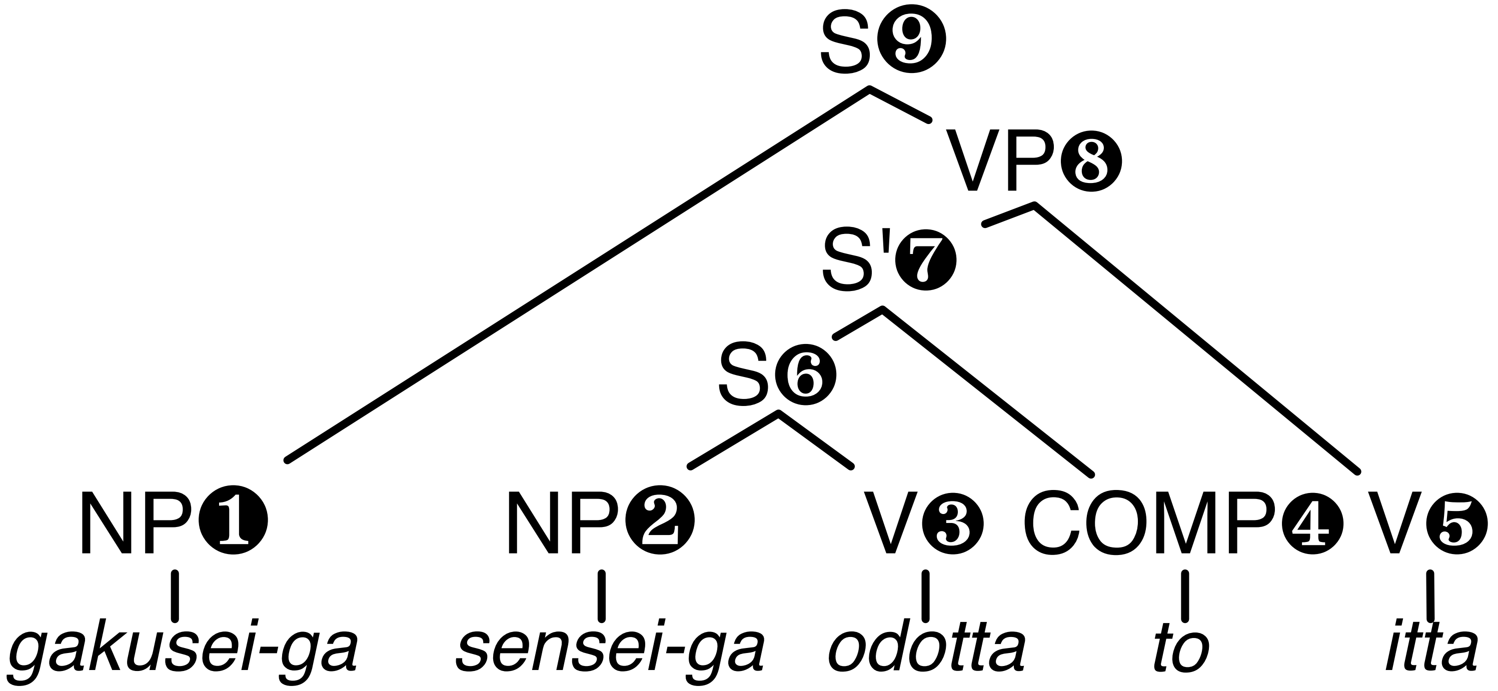

S❻
NP❶
NP❷
V❸
COMP❹
V❺
the student
the teacher
danced
that
said

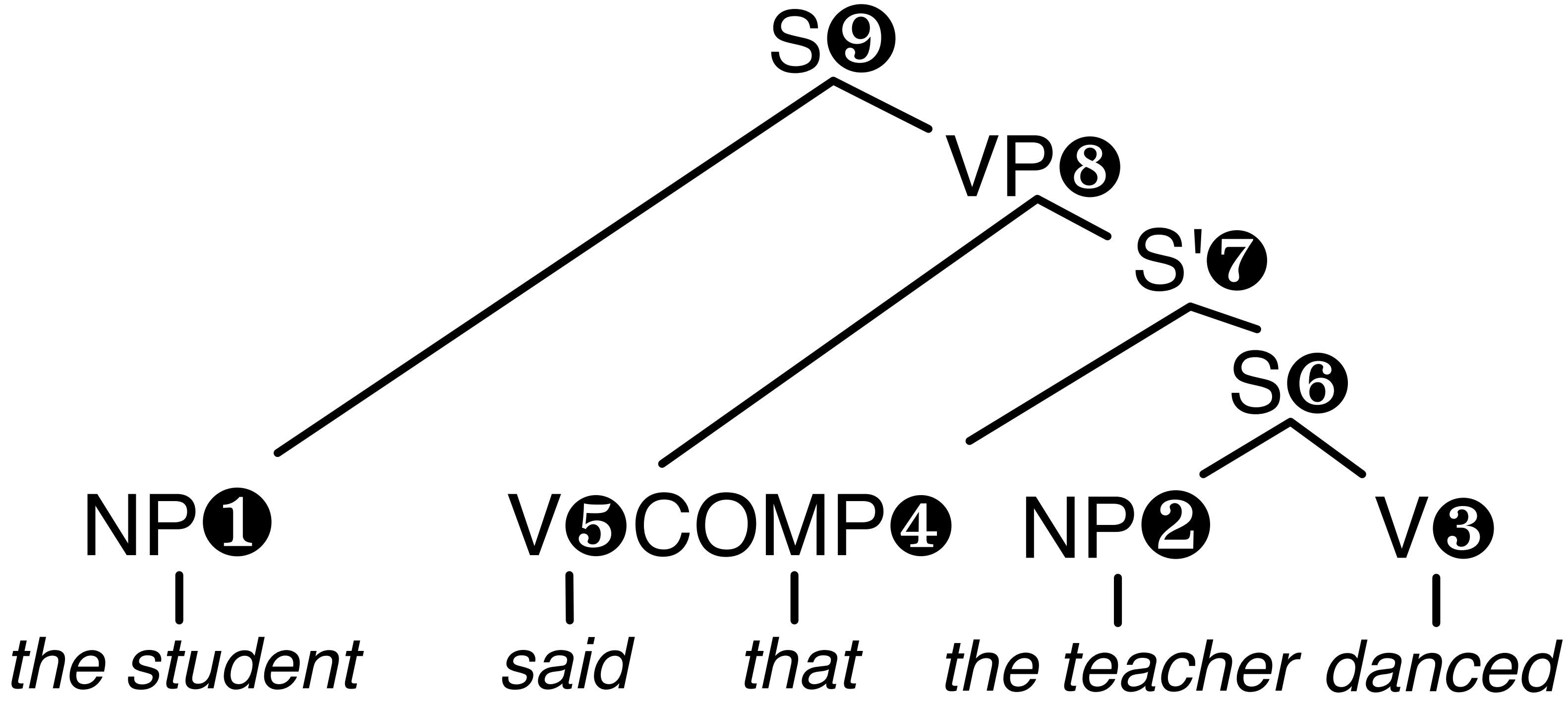

S'❼
S❻
NP❶
NP❷
V❸
COMP❹
V❺
gakusei-ga
sensei-ga
odotta
to
itta


S'❼
S❻
NP❶
NP❷
V❸
COMP❹
V❺
the student
the teacher
danced
that
said


VP❽
S'❼
S❻
NP❶
NP❷
V❸
COMP❹
V❺
gakusei-ga
sensei-ga
odotta
to
itta

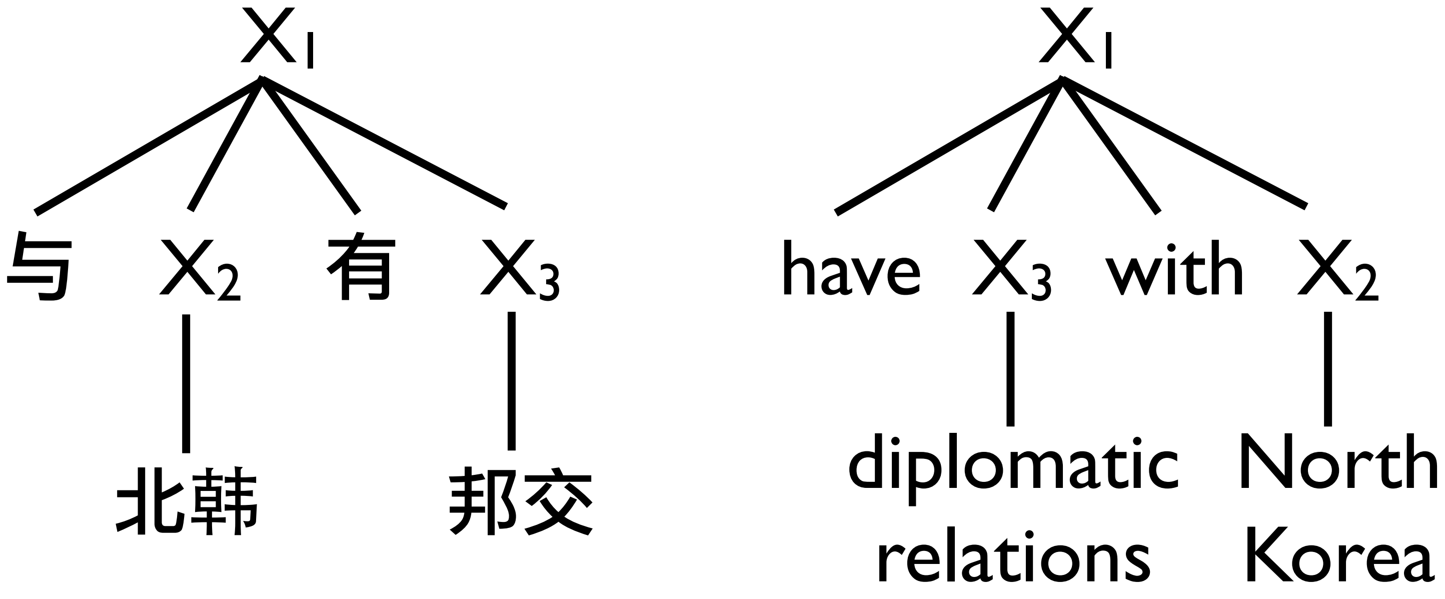

VP❽
S'❼
S❻
NP❶
COMP❹
NP❷
V❸
V❺
the student
that
the teacher
danced
said


S❾
VP❽
S'❼
S❻
NP❶
NP❷
V❸
COMP❹
V❺
gakusei-ga
sensei-ga
odotta
to
itta


S❾
VP❽
S'❼
S❻
NP❶
V❺
COMP❹
NP❷
V❸
the student
said
that
the teacher
danced

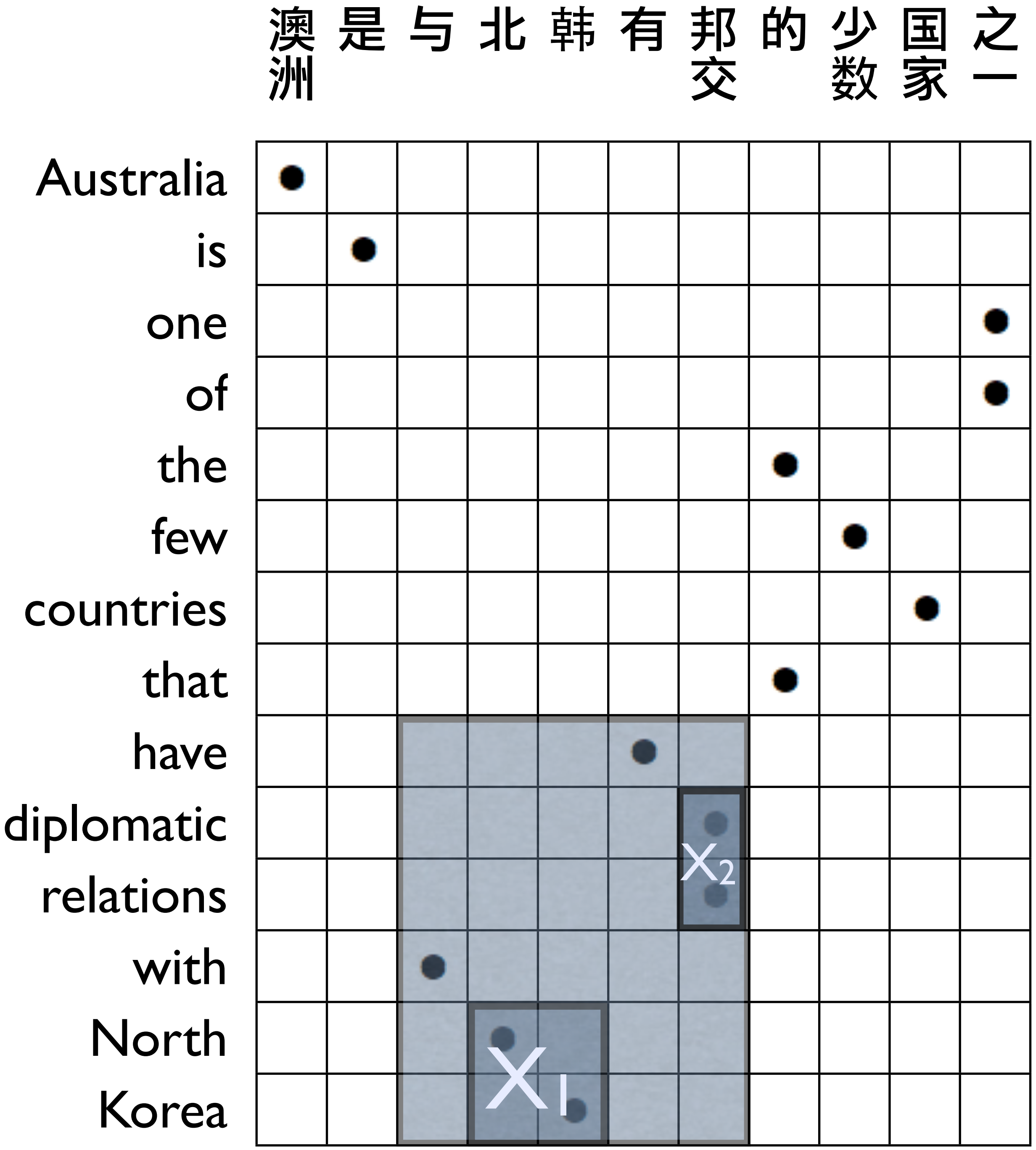
Hiero-style rules

- Currently, only support Hiero-style rules with single non-terminal
- Not as nice as linguistically motivated rules, but useful for things like reordering



**Thanks to David Chiang for Hiero slides*

Extracting Hiero rules



	澳洲	是	与	北	韩	有	邦交	的	少数	国家	之一
Australia	●										
is		●									
one											●
of											●
the								●			
few									●		
countries										●	
that								●			
have						●					
diplomatic							●				
relations							●				
with			●								
North				●							
Korea					●						

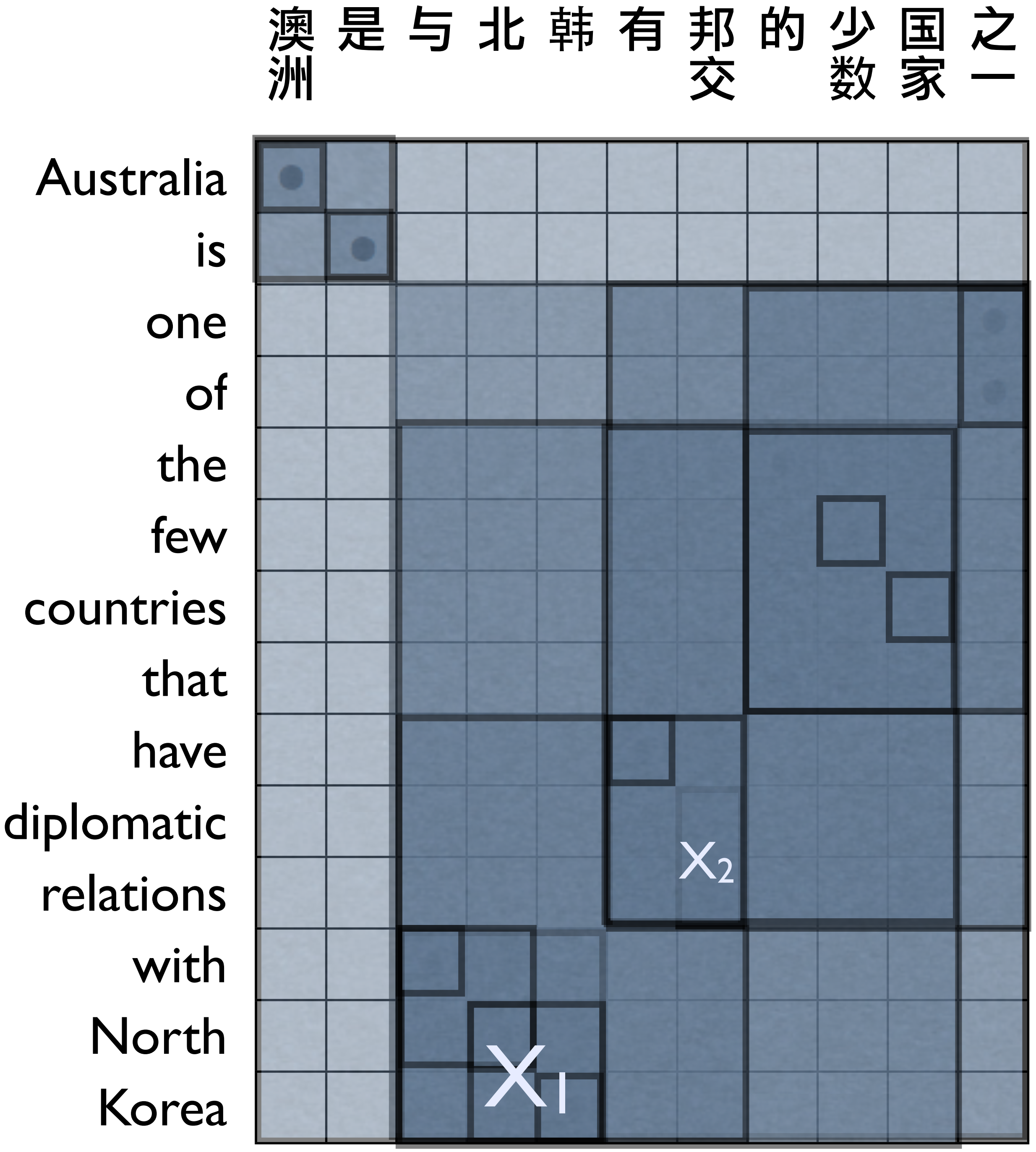
(与 北 韩 有 邦交, have diplomatic relations with North Korea)

(邦交, diplomatic relations)

(北 韩, North Korea)

X → 与 X_1 有 X_2,
have X_2 with X_1

Extracting Hiero rules



(与 北 韩 有 邦交, have diplomatic relations with North Korea)

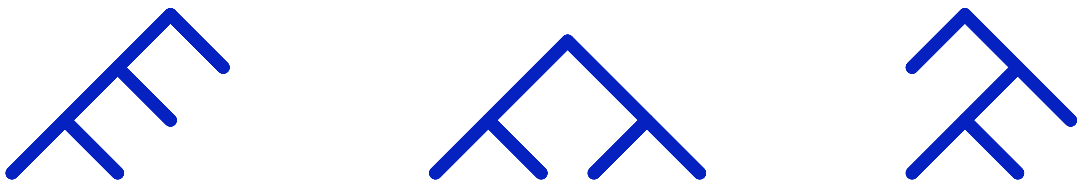
(邦交, diplomatic relations)

(北 韩, North Korea)

$X \rightarrow$ 与 X_1 有 X_2, have X_2 with X_1

Some challenges with Hiero

- Large number of rules
  - Decreases time/space efficiency
- Spurious ambiguity
  - Decreases time efficiency
  - Pollutes *n*-best lists

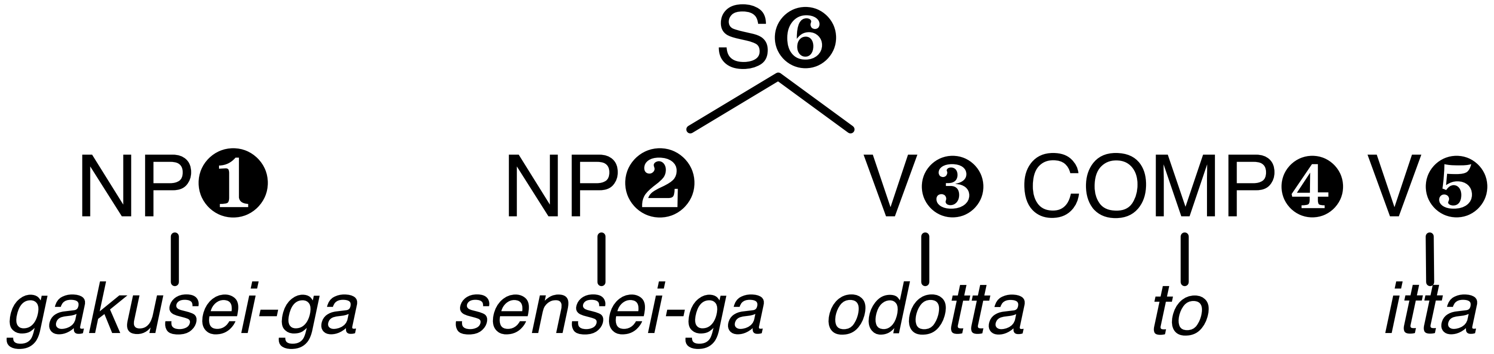


- Ad hoc constraints:
  - Initial phrases ≤10 words, rules ≤6 symbols
  - Require an aligned terminal
  - Limit to two nonterminals, nonadjacent

Some challenges with SCFGs

- Integration of an n-gram language model is difficult under SCFGs
  - Using an n-gram LM generally makes translation quality much better
  - We do not construct a translation in a left-to-right fashion as in phrase-based SMT

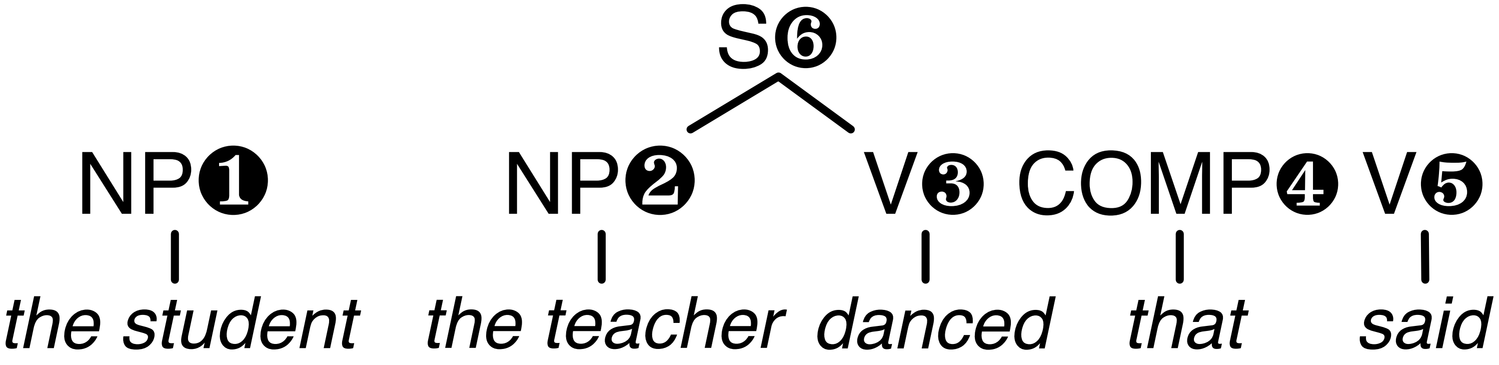
n-gram language model



S❻

NP❶ NP❷ V❸ COMP❹ V❺

gakusei-ga sensei-ga odotta to itta

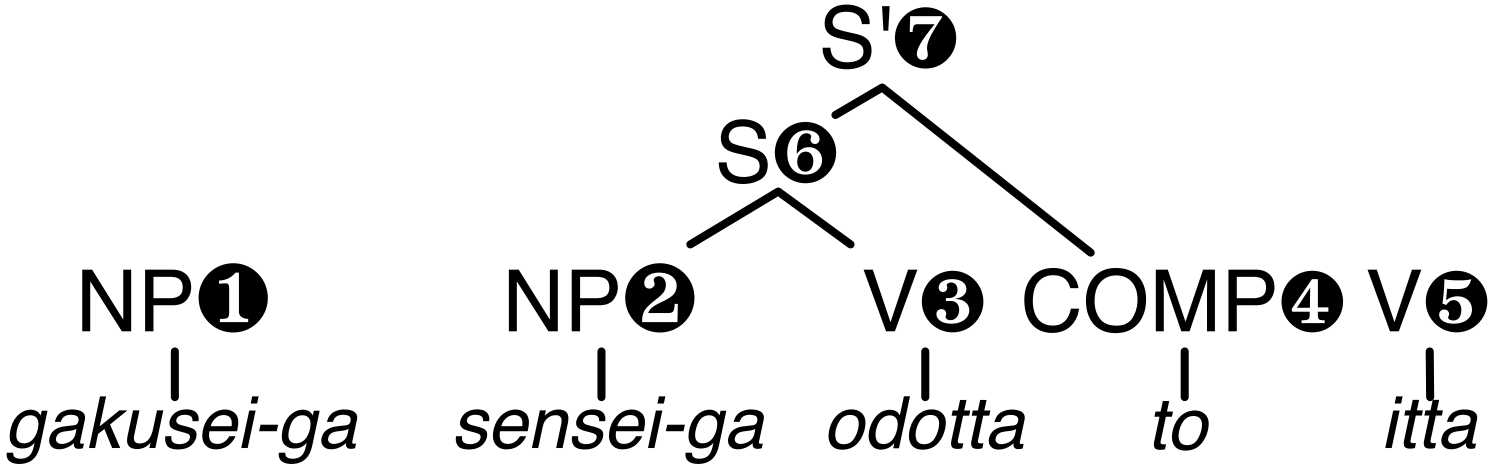


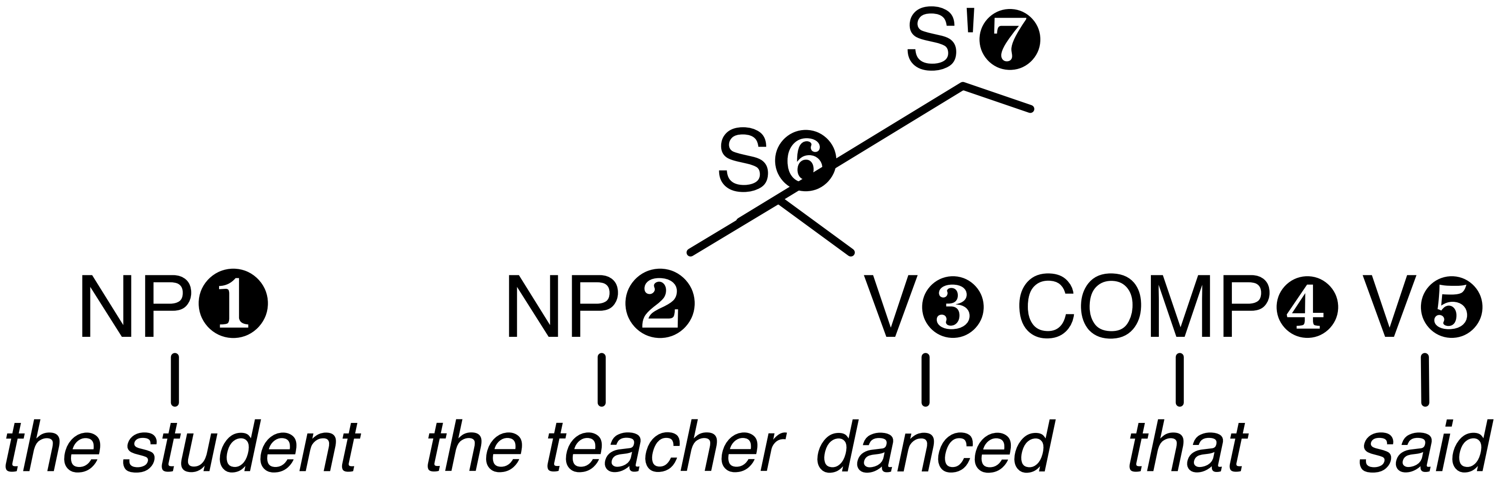
S❻

NP❶ NP❷ V❸ COMP❹ V❺

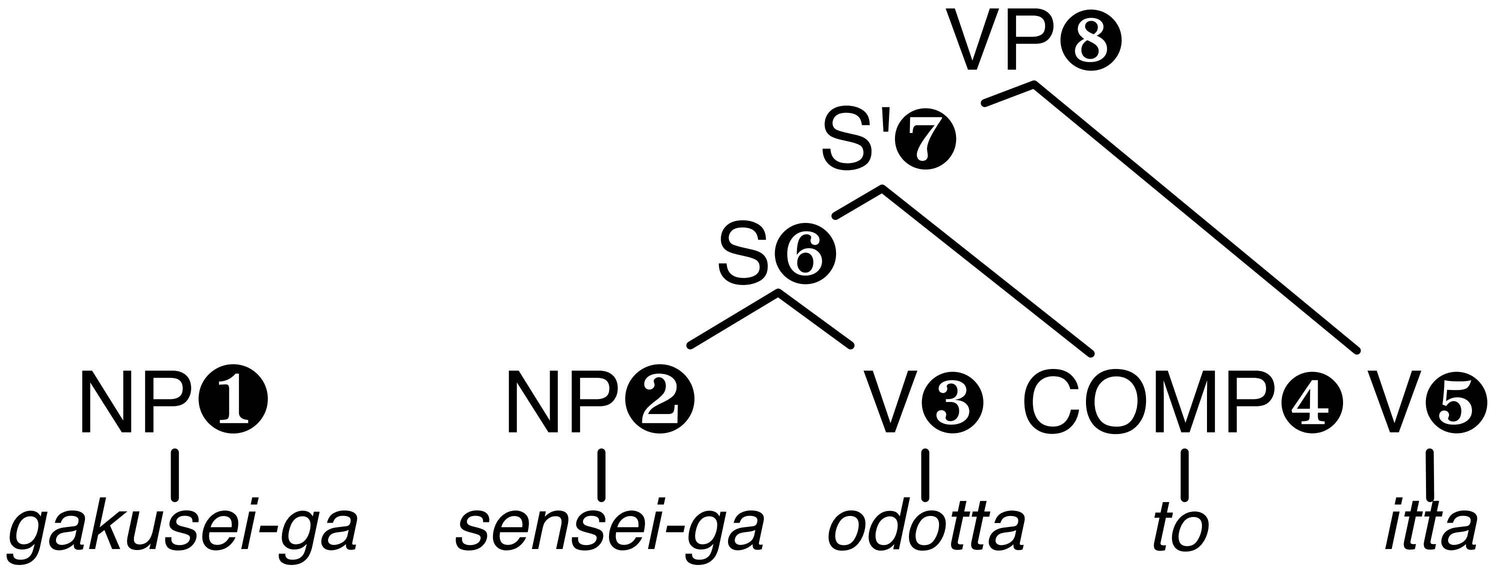
the student the teacher danced that said

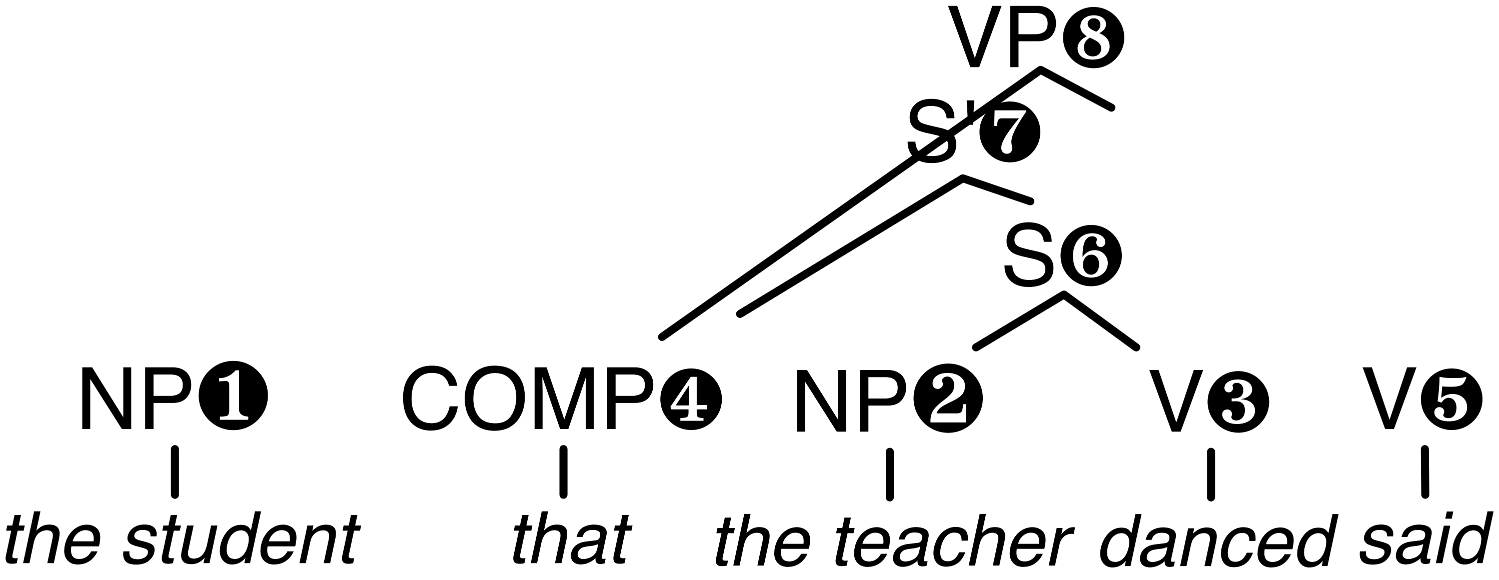
n-gram language model





n-gram language model

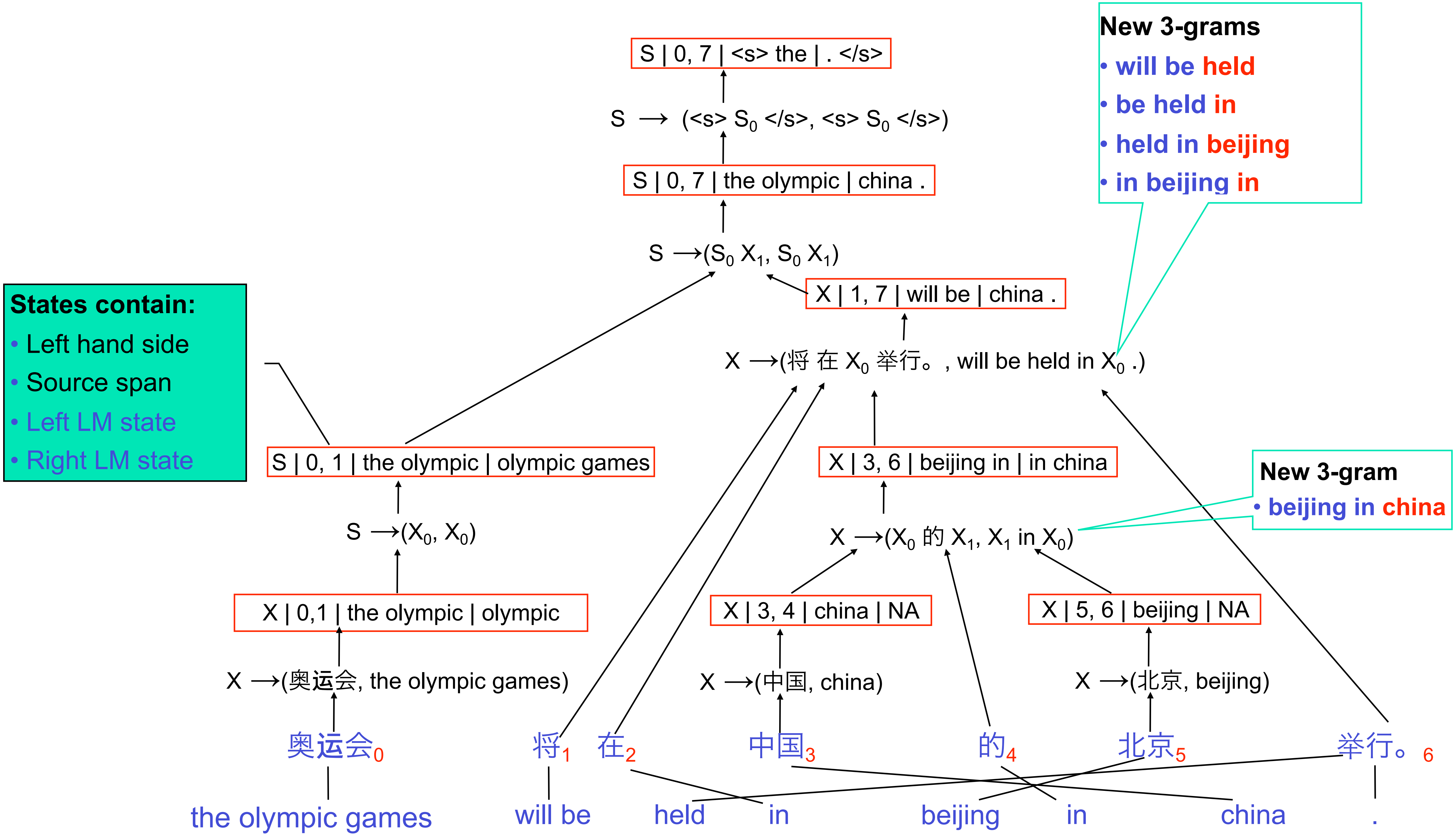




LM state in chart parsing

- Decoding takes place via chart parsing
- Chart parsing
  - Decoder maintains a *chart*, which contains an array of *cells*
  - A cell maintains a list of *items*
  - Derivations are stored in a structure called a *hypergraph*.
- State of an *Item*
  - Source span
  - Left hand side nonterminal symbol
  - Left/right LM state

Example Derivation



States contain:

- Left hand side
- Source span
- Left LM state
- Right LM state

New 3-grams

- **will be held**
- **be held in**
- **held in beijing**
- **in beijing in**

New 3-gram

- **beijing in china**

S | 0, 7 | <s> the | . </s>

S → (<s> S_0 </s>, <s> S_0 </s>)

S | 0, 7 | the olympic | china .

S →(S_0 X_1, S_0 X_1)

X | 1, 7 | will be | china .

X →(将 在 X_0 举行。, will be held in X_0 .)

S | 0, 1 | the olympic | olympic games

X | 3, 6 | beijing in | in china

S →(X_0, X_0)

X →(X_0 的 X_1, X_1 in X_0)

X | 0,1 | the olympic | olympic

X | 3, 4 | china | NA

X | 5, 6 | beijing | NA

X →(奥运会, the olympic games)

X →(中国, china)

X →(北京, beijing)

奥运会$_0$ 将$_1$ 在$_2$ 中国$_3$ 的$_4$ 北京$_5$ 举行。$_6$

the olympic games will be held in beijing in china .

Other Bells and Whistles

- Beam and cube pruning Huang and Chiang (2007)
- Built in minimum error rate training Och and Ney (2003)
  - Modular, so easily allows optimization to objective functions other than Bleu Zaidan (2009)
- Suffix array indexing of the parallel corpus Lopez (2007)
  - Allows on-the-fly look up of translation rules
- *n*-best extraction from hypergraphs Chiang (2007)
- Equivalent LM state maintenance Li and Khudanpur (2008)
- Support for parallel decoding

Decoding Speed

- Training data
  - Task: Chinese to English translation
  - Sub-sampled from parallel corpus containing approx 3M sentence pairs
    - obtained 570k sentence pairs
    - Number of translation rules: 3M
  - LM data: Gigaword and English side of the parallel
    - Number of n-grams in LM: 49M
- Speed and translation quality comparison:

38 times faster than the baseline!

Decoder	**Speed (sec/sent)**	**BLEU-4**	
		MT03	**MT03**
Python	26.5	34.4%	32.7%
Java	1.2	34.5%	32.9%
Java (parallel)	0.7		

Current directions

- Recreating Syntax-Augmented Machine Translation for more linguistically motivated translation rules Zollmann and Venugopal (2006)
- Implementing Bloom Filter Language Models to allow much larger LMs to be used with less memory Talbot and Osborne (2007)
- Integrating Minimum Bayes Risk Re-ranking of n-best translations extracted from hypergraphs Kumar and Byrne (2004)
- Scaling to a 1,000,000,000 word parallel corpus Callison-Burch (2009)

Where to get the software

- Subversion repository at
  - http://sourceforge.net/projects/joshua
- Quick installation instructions are in
  - joshua/trunk/INSTALL.txt
- Instructions on running with sample grammar are in
  - joshua/trunk/README.txt
- For support write to
  - Joshua_support@googlegroups.com

Thanks!

- Happy hacking!