Aspects of Tree-Based Statistical Machine Translation

Marcello Federico
(based on slides by Gabriele Musillo)

Human Language Technology
FBK-irst

2011

Outline

Tree-based translation models:

- Synchronous context free grammars
- BTG alignment model
- Hierarchical phrase-based model

Decoding with SCFGs:

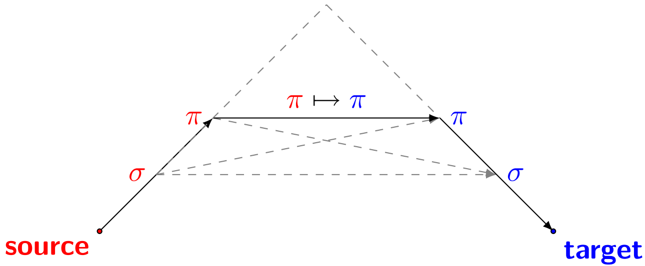
- Translation as Parsing
- DP-based chart decoding
- Integration of language model scores

Learning SCFGs

- Rule extraction from phrase-tables

Tree-Based Translation Models

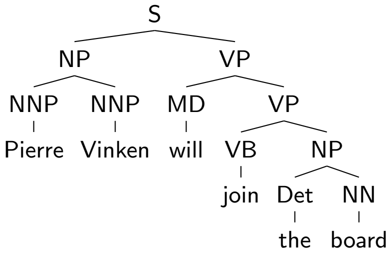
Levels of Representation in Machine Translation:



- $\pi \mapsto \sigma$: tree-to-string
- $\sigma \mapsto \pi$: string-to-tree
- $\pi \mapsto \pi$: **tree-to-tree**

? Appropriate Levels of Representation ?

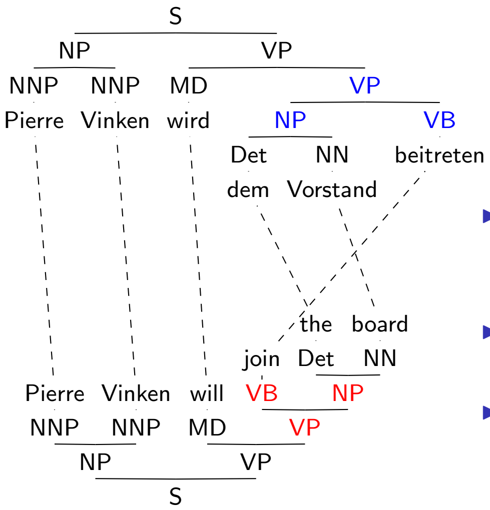
Tree Structures



Syntactic Structures:

- **rooted ordered** trees
- internal nodes labeled with **syntactic categories**
- leaf nodes labeled with words
- **linear** and **hierarchical** relations between nodes

Tree-to-Tree Translation Models



- syntactic **generalizations** over pairs of languages: **isomorphic** trees
- syntactically informed **unbounded reordering**
- formalized as derivations in **synchronous grammars**

? Adequacy of Isomorphism Assumption ?

Context-Free Grammars

CFG (Chomsky, 1956):

- formal model of languages
- more expressive than FSAs and REs
- first used in linguistics to describe **embedded** and **recursive** structures

CFG Rules:

- **left-hand side nonterminal symbol**
- **right-hand side string of nonterminal or terminal symbols**
- distinguished **start** nonterminal symbol

$$\begin{cases} S \rightarrow 0S1 & S \text{ rewrites as } 0S1 \\ S \rightarrow \epsilon & S \text{ rewrites as } \epsilon \end{cases}$$

CFG Derivations

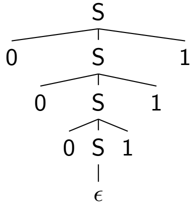
Generative Process:

1. Write down the **start** nonterminal symbol.
2. **Choose** a **rule** whose left-hand side is the left-most written down nonterminal symbol.
3. **Replace** the symbol with the right-hand side of that rule.
4. Repeat step 2 **while there are nonterminal symbols written down**.

Derivation

$$
\begin{aligned}
S &\Rightarrow_{S\to 0S1} & 0S1 \\
&\Rightarrow_{S\to 0S1} & 00S11 \\
&\Rightarrow_{S\to 0S1} & 000S111 \\
&\Rightarrow_{S\to \epsilon} & 000111
\end{aligned}
$$

Parse tree



CFG Formal Definitions

CFG $G = \langle V, \Sigma, R, S \rangle$:

- V: finite set of nonterminal symbols
- Σ: finite set of terminals, disjoint from V
- R: finite set of rule $\alpha \rightarrow \beta$, with α a nonterminal and β a string of terminals and nonterminals
- S: the start nonterminal symbol

Let u, v be strings of $V \cup \Sigma$, and $\alpha \rightarrow \beta \in R$, then we say:

- $u\alpha v$ **yields** $u\beta v$, written as $u\alpha v \Rightarrow u\beta v$
- u **derives** v, written as $u \Rightarrow^{\star} v$, if $u = v$ or a sequence $u_1, u_2, \ldots, u_k$ exists for $k \geq 0$ and $u \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \ldots \Rightarrow u_k \Rightarrow v$
- $L(G) = \{w \in \Sigma^{\star} | S \Rightarrow^{\star} w\}$

CFG Examples

G$_1$:

$$R = \{S \rightarrow NP\ VP,$$
$$NP \rightarrow N | DET\ N | N\ PP,$$
$$VP \rightarrow V\ NP | V\ NP\ PP,$$
$$PP \rightarrow P\ NP,$$
$$DET \rightarrow the | a,$$
$$N \rightarrow Alice | Bob | trumpet,$$
$$V \rightarrow chased,$$
$$P \rightarrow with\}$$

? **derivations** of
Alice chased Bob with the trumpet

G$_3$:

$$R = \{NP \rightarrow NP\ CONJ\ NP | NP\ PP | DET\ N,$$
$$PP \rightarrow P\ NP, P \rightarrow of,$$
$$DET \rightarrow the | two | three,$$
$$N \rightarrow mother | pianists | singers,$$
$$CONJ \rightarrow and\}$$

? **derivations** of
the mother of three pianists and two singers

- same parse tree can be derived in different ways ($\neq$ order of rules)
- same sentence can have different parse trees ($\neq$ choice of rules)

Transduction Grammars aka Synchronous Grammars

TG (Lewis and Stearns, 1968; Aho and Ullman, 1969):

- **two or more strings derived simultaneously**
- more powerful than FSTs
- used in NLP to model **alignments**, unbounded **reordering**, and mappings from surface forms to logical forms

Synchronous Rules:

- left-hand side nonterminal symbol associated with **source** and **target** right-hand sides
- **bijection** $_{[]}$ mapping nonterminals in source and target of right-hand sides

$$\begin{cases} E \rightarrow E_{[1]} + E_{[3]} \;/\; + \; E_{[1]} \; E_{[3]} & \text{infix to Polish notation} \\ E \rightarrow E_{[1]} * E_{[2]} \;/\; * \; E_{[1]} \; E_{[2]} & \\ E \rightarrow n \;/\; n & n \in N \end{cases}$$

Synchronous CFG

NP $\longrightarrow$ DT$_{\boxed{1}}$NPB$_{\boxed{2}}$ / DT$_{\boxed{1}}$NPB$_{\boxed{2}}$

NPB $\longrightarrow$ JJ$_{\boxed{1}}$NN$_{\boxed{2}}$ / JJ$_{\boxed{1}}$NN$_{\boxed{2}}$

NPB $\longrightarrow$ NPB$_{\boxed{1}}$JJ$_{\boxed{2}}$ / JJ$_{\boxed{2}}$NPB$_{\boxed{1}}$

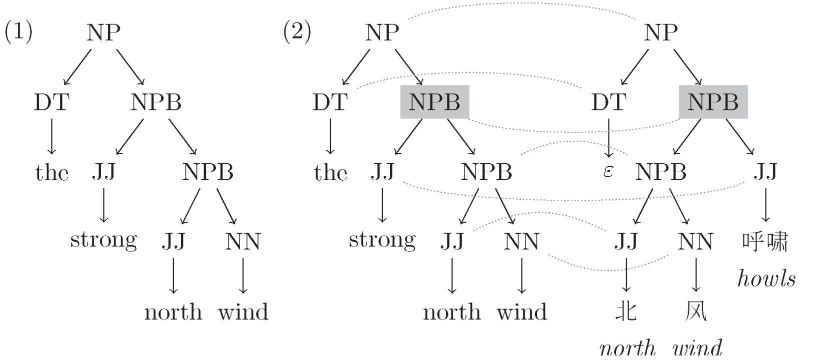
DT $\longrightarrow$ the / ε

JJ $\longrightarrow$ strong / 呼啸

JJ $\longrightarrow$ north / 北

NN $\longrightarrow$ wind / 风

- **1-to-1 correspondence** between nodes
- **isomorphic** derivation trees
- uniquely determined **word alignment**



Bracketing Transduction Grammars

BTG (Wu, 1997):

- special form of SCFG
- only one nonterminal X
- nonterminal rules:

$$\begin{cases} X \rightarrow X_{[1]}\ X_{[2]}\ /\ X_{[1]}\ X_{[2]} & \textbf{monotone} \text{ rule} \\ X \rightarrow X_{[1]}\ X_{[2]}\ /\ X_{[2]}\ X_{[1]} & \textbf{inversion} \text{ rule} \end{cases}$$

- preterminal rules where $e \in V_t \cup \{\epsilon\}$ and $f \in V_s \cup \{\epsilon\}$:

$$\begin{cases} X \rightarrow f\ /\ e & \text{lexical translation rules} \end{cases}$$

SCFG Derivations

Generative Process:

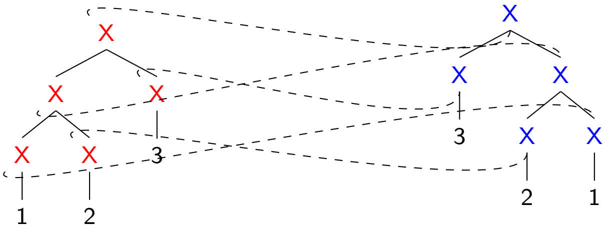
1. Write down the **source** and **target** start symbols.
2. **Choose** a **synchronous rule** whose left-hand side is the left-most written down **source** nonterminal symbol.
3. **Simultaneously rewrite** the **source** symbol and its **corresponding target** symbol with the **source** and the **target** side of the rule, respectively.
4. Repeat step 2 **while there are written down source and target nonterminal symbols**.

$$\begin{cases} X \rightarrow X_{[1]}\ X_{[2]}\ /\ X_{[1]}\ X_{[2]} & \\ X \rightarrow X_{[1]}\ X_{[2]}\ /\ X_{[2]}\ X_{[1]} & \\ X \rightarrow k\ /\ k & k \in \{1, 2, 3\} \end{cases}$$

BTG Alignments

$$
\begin{aligned}
\langle X, X \rangle &\Rightarrow_{X_1 X_2 / X_2 X_1} & \langle X_1 X_2, X_2 X_1 \rangle \\
&\Rightarrow_{X_1 X_2 / X_2 X_1} & \langle X_3 X_4 X_2, X_2 X_4 X_3 \rangle \\
&\Rightarrow_{1/1} & \langle 1 X_4 X_2, X_2 X_4 1 \rangle \\
&\Rightarrow_{2/2} & \langle 12 X_2, X_2 21 \rangle \\
&\Rightarrow_{3/3} & \langle 123, 321 \rangle
\end{aligned}
$$

re-indexed symbols



Phrase-Based Models and SCFGs

SCFG Formalization of Phrase-Based Translation Models:

- $\forall$ phrase pair $\langle \tilde{f}, \tilde{e} \rangle$, make rule

$$X \rightarrow \tilde{f} \ / \ \tilde{e}$$

- make monotone rules

$$S \rightarrow S_{[1]} \ X_{[2]} \ / \ S_{[1]} \ X_{[2]}$$

$$S \rightarrow X_{[1]} \ / \ X_{[1]}$$

- make reordering rules

$$X \rightarrow X_{[1]} \ X_{[2]} \ / \ X_{[2]} \ X_{[1]}$$

? Completeness ? Correctness ?

Hierarchical Phrase-Based Models

HPBM (Chiang, 2007):

- first tree-to-tree approach to perform better than phrase-based systems in large-scale evaluations
- **discontinuous phrases**
- **long-range reordering rules**
- formalized as synchronous context-free grammars

澳洲	是	与	北韩	有	邦交	的	少数	国家	之一	。
Aozhou	shi	yu	Beihan	you	bangjiao	de	shaoshu	guojia	zhiyi	.
Australia	is	with	North Korea	have	dipl. rels.	that	few	countries	one of	.

Australia is one of the few countries that have diplomatic relations with North Korea.

HPBM: Motivations

Aozhou	shi	yu	Beihan	you	bangjiao	de	shaoshu	guojia	zhiyi .
Australia	is	with	North Korea	have	dipl. rels.	that	few	countries	one of .

Typical Phrase-Based Chinese-English Translation:

[Aozhou] $[\text{shi}]_1$ $[\text{yu Beihan}]_2$ [you] [bangjiao] [de shaoshu guojia zhiyi] [.]

[Australia] [has] [dipl. rels.] $[\text{with North Korea}]_2$ $[\text{is}]_1$ [one of the few countries] [.]

- Chinese VPs follow PPs / English VPs precede PPs

  yu X_1 *you* X_2 / *have* X_2 *with* X_1

- Chinese NPs follow RCs / English NPs precede RCs

  X_1 *de* X_2 / *the* X_2 *that* X_1

- translation of *zhiyi* construct in English word order

  X_1 *zhiyi* / *one of* X_1

HPBM: Example Rules

$$S \rightarrow X_1 \ / \ X_1 \quad (1)$$
$$S \rightarrow S_1 \ X_2 \ / \ S_1 \ X_2 \quad (2)$$
$$X \rightarrow yu \ X_1 \ you \ X_2 \ / \ have \ X_2 \ with \ X_1 \quad (3)$$
$$X \rightarrow X_1 \ de \ X_2 \ / \ the \ X_2 \ that \ X_1 \quad (4)$$
$$X \rightarrow X_1 \ zhiyi \ / \ one \ of \ X_1 \quad (5)$$
$$X \rightarrow Aozhou \ / \ Australia \quad (6)$$
$$X \rightarrow Beihan \ / \ N. \ Korea \quad (7)$$
$$X \rightarrow she \ / \ is \quad (8)$$
$$X \rightarrow bangjiao \ / \ dipl.rels. \quad (9)$$
$$X \rightarrow shaoshu \ guojia \ / \ few \ countries \quad (10)$$

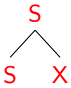
HPBM: Example Translation Step 1

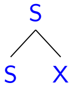
S

S

$$S \rightarrow S_1\ X_2\ /\ S_1\ X_2$$

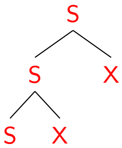
HPBM: Example Translation Step 2

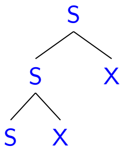




$$S \rightarrow S_1 \; X_2 \; / \; S_1 \; X_2$$

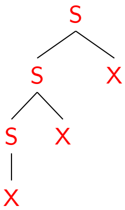
HPBM: Example Translation Step 3

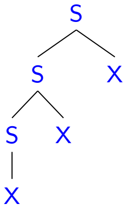




$$S \rightarrow X_1 \;/\; X_1$$

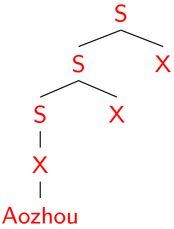
HPBM: Example Translation Step 4

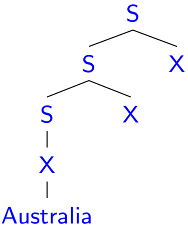




$X \rightarrow$ *Aozhou* / *Australia*

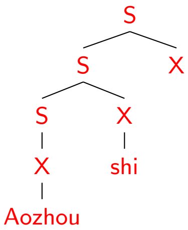
HPBM: Example Translation Step 5

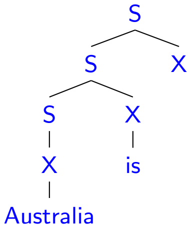




$X \rightarrow$ *she* / *is*

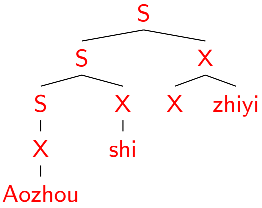
HPBM: Example Translation Step 6

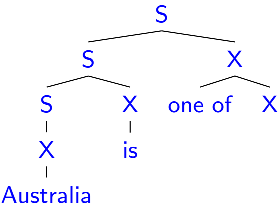




$$X \rightarrow X_1 \text{ } zhiyi \text{ } / \text{ } one \text{ } of \text{ } X_1$$

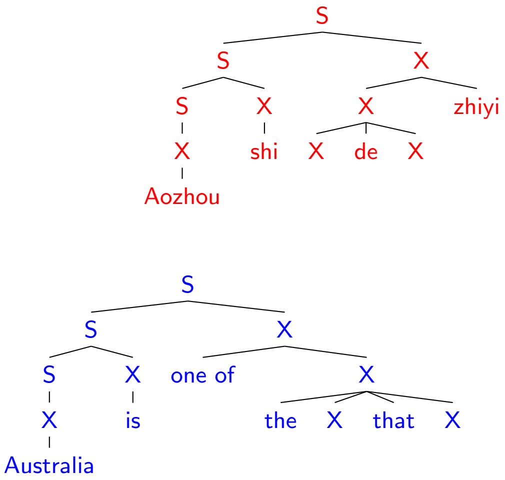
HPBM: Example Translation Step 7





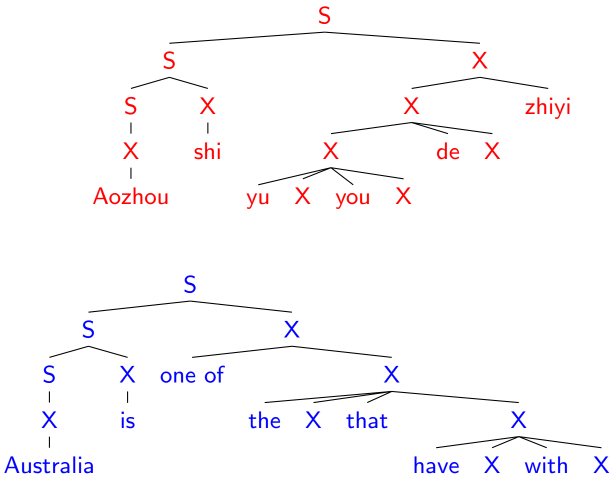
$$X \rightarrow X_1 \; de \; X_2 \; / \; the \; X_2 \; that \; X_1$$

HPBM: Example Translation Step 8



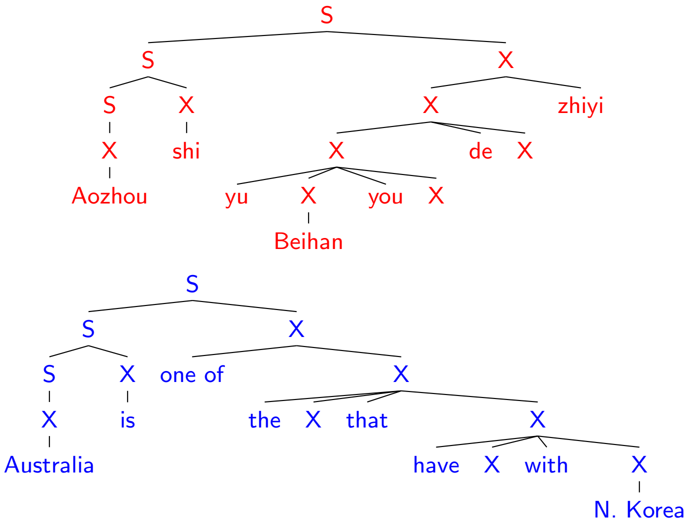
$$X \rightarrow \textit{yu } X_1 \textit{ you } X_2 \ / \ \textit{have } X_2 \textit{ with } X_1$$

HPBM: Example Translation Step 9



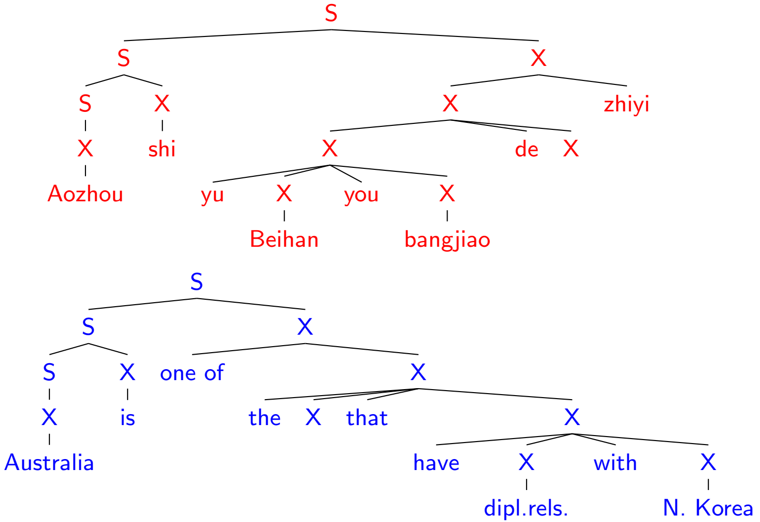
$X \rightarrow$ *Beihan* / *N. Korea*

HPBM: Example Translation Step 10



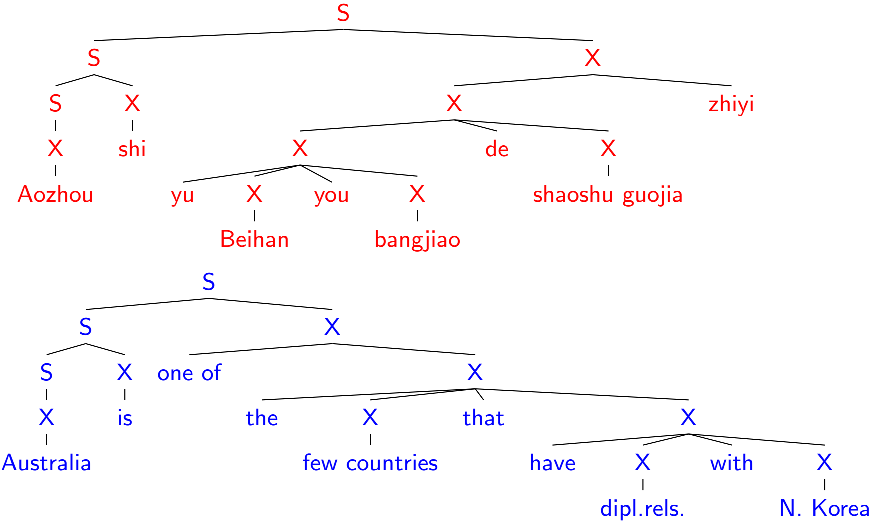
$X \rightarrow$ *bangjiao* / *dipl.rels.*

HPBM: Example Translation Step 11



$X \rightarrow$ *shaoshu guojia* / *few countries*

HPBM: Example Translation



Summary

Synchronous Context-Free Grammars:

- formal model to synchronize source and target derivation processes
- BTG alignment model
- HPB recursive reordering model

Additional topics (optional):

- Decoding SCFGs: Translation as Parsing
- Learning SCFGs from phrase tables

Synchronous Context-Free Grammars

SCFGs:

- CFGs in **two dimensions**
- **synchronous** derivation of **isomorphic**[a] **trees**
- **unbounded reordering** preserving **hierarchy**

[a] excluding leafs

$\dots$

$VB \rightarrow PRP_1\ VB1_2\ VB2_3\ /\ PRP_1\ VB2_3\ VB1_2$

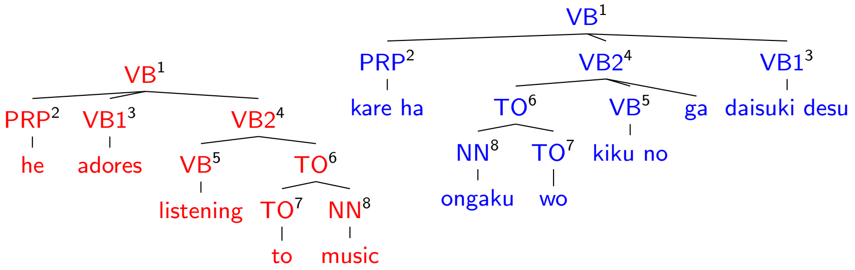
$VB2 \rightarrow VB_1\ TO_2\ /\ TO_2\ VB_1\ ga$

$TO \rightarrow TO_1\ NN_2\ /\ NN_2\ TO_1$

$PRP \rightarrow he\ /\ kare\ ha$

$VB \rightarrow listening\ /\ daisuki\ desu$

$\dots$



Weighted SCFGs

- rules $A \rightarrow \alpha \ / \ \beta$ associated with positive weights $\mathbf{w}_{A \rightarrow \alpha / \beta}$
- derivation trees $\pi = \langle \pi_1, \pi_2 \rangle$ weighted as

$$\mathbf{W}(\pi) = \prod_{A \rightarrow \alpha/\beta \in G} \mathbf{w}_{A \rightarrow \alpha/\beta}^{f_{A \rightarrow \alpha/\beta}(\pi)}$$

- probabilistic SCFGs if the following conditions hold

$$\mathbf{w}_{A \rightarrow \alpha/\beta} \in [0,1] \text{ and } \sum_{\alpha,\beta} \mathbf{W}_{A \rightarrow \alpha/\beta} = 1$$

- notice: SCFGs might well include rules of type

$$A \rightarrow \alpha/\beta_1 \ldots A \rightarrow \alpha/\beta_k$$

MAP Translation Problem

Maximum A Posterior Translation:

$$e^\star = \operatorname*{argmax}_{e} p(e|f)$$
$$= \operatorname*{argmax}_{e} \sum_{\pi \in \Pi(f,e)} p(e, \pi | f)$$

$\Pi(f, e)$ *is the set of synchronous derivation trees yielding* $\langle f, e \rangle$

- Exact MAP decoding is NP-hard (Simaan, 1996; Satta and Peserico, 2005)

Viterbi Approximation

Tractable Approximate Decoding:

$$
\begin{aligned}
e^{\star} &= \operatorname*{argmax}_{e} \sum_{\pi \in \Pi(f,e)} p(e, \pi | f) \\
&\simeq \operatorname*{argmax}_{e} \max_{\pi \in \Pi(f,e)} p(e, \pi | f) \\
&= E(\operatorname*{argmax}_{\pi \in \Pi(f)} p(\pi))
\end{aligned}
$$

$\Pi(f)$ *is the set of synchronous derivations yielding* f

$E(\pi)$ *is the target string resulting from the synchronous derivation* π

Translation as Parsing

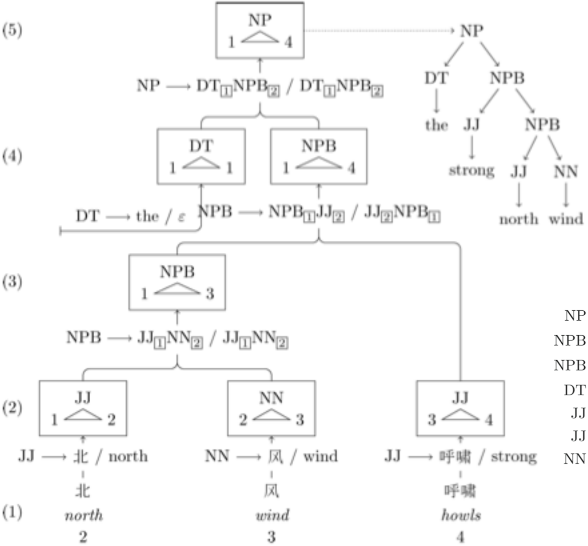
$$\pi^\star = \underset{\pi \in \Pi(f)}{\operatorname{argmax}}\, p(\pi)$$

Parsing Solution:

1. compute the **most probable derivation tree** that generates f using the **source dimension** of the WSCFG
2. build the **translation string** e by applying the **target dimension** of the rules used in the most probable derivation

- most probable derivation computed in $O(n^3)$ using **dynamic programming** algorithms for parsing **weighted CFGs**
- transfer of algorithms and optimizations developed for CFG to SMT

Translation as Parsing: Illustration



$NP \longrightarrow DT_{\boxed{1}} NPB_{\boxed{2}}$ / $DT_{\boxed{1}} NPB_{\boxed{2}}$

$NPB \longrightarrow JJ_{\boxed{1}} NN_{\boxed{2}}$ / $JJ_{\boxed{1}} NN_{\boxed{2}}$

$NPB \longrightarrow NPB_{\boxed{1}} JJ_{\boxed{2}}$ / $JJ_{\boxed{2}} NPB_{\boxed{1}}$

DT ⟶ the / ε

JJ ⟶ strong / 呼啸

JJ ⟶ north / 北

NN ⟶ wind / 风

Weighted CFGs in Chomsky Normal Form

WCFGs:

- rules $A \rightarrow \alpha$ associated with positive weights $\mathbf{w}_{A\rightarrow\alpha}$
- derivation trees π weighted as

$$\mathbf{W}(\pi) = \prod_{A\rightarrow\alpha\in G} \mathbf{w}_{A\rightarrow\alpha}^{f_{A\rightarrow\alpha}(\pi)}$$

- probabilistic CFGs if the following conditions hold

$$\mathbf{w}_{A\rightarrow\alpha} \in [0,1] \text{ and } \sum_{\alpha} \mathbf{w}_{A\rightarrow\alpha} = 1$$

WCFGs in CNF:

- rules in CFGs in Chomsky Normal Form: $\mathbf{A} \rightarrow \mathbf{BC}$ or $\mathbf{A} \rightarrow \mathbf{a}$
- **equivalence** between WCFGs and WCFGs in CNF
- no analogous equivalence holds for weighted SCFGs

Translation as Weighted CKY Parsing

Given a WSCFG G and a source string f:

1. project **G** into its source WCFG **G**

$$A \xrightarrow{w} \alpha \in G \text{ if } A \xrightarrow{w} \alpha/\beta \in G \text{ and } \forall A \xrightarrow{w'} \alpha/\beta' \in G \;\; w \geq w'$$

2. transform **G** into its CNF **G'**
3. solve $\pi'^{\star} = \text{argmax}_{\pi' \in \Pi_{G'}(f)} \mathbf{p}(\pi')$ with the CKY algorithm
4. revert $\pi'^{\star}$ into $\pi^{\star}$, the derivation tree according to **G**
5. map $\pi^{\star}$ into its corresponding target tree π
6. read off the translation e from π

Weighted CKY Parsing

Dynamic Programming:

- **recursive division of problems into subproblems**
- **optimal solutions compose optimal sub-solutions (Bellman's Principle)**
- **tabulation of subproblems and their solutions**

CKY Parsing:

- subproblems: **parsing substrings of the input string** $u_1 \ldots u_n$
- solutions to subproblems tabulated using a **chart**
- **bottom up** algorithm starting with derivation of terminals
- $O(n^3|G|)$ time complexity
- widely used to perform statistical inference over random trees

Weighted CKY Parsing

Problem: $\pi^\star = \text{argmax}_{\pi \in \Pi_G(u = u_1 \ldots u_n)} p(\pi)$

- **DP chart**:

$$M_{i,k,A} = \text{ maximum probability of } A \Rightarrow^\star u_{i+1,k}$$

- **base** case, $k - i = 1$:

$$\forall\ 1 \leq i \leq n\ M_{i-1,i,A} = \mathbf{w}_{A \rightarrow u_i})$$

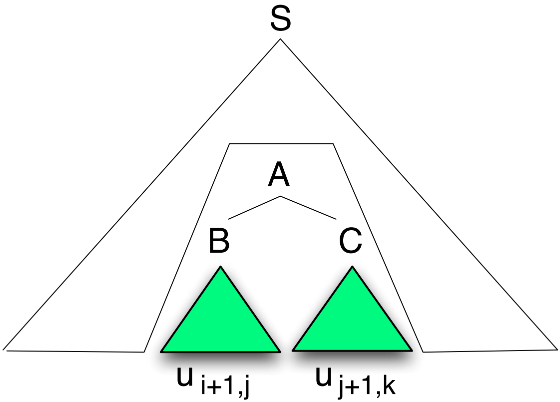
- **inductive** case, $k - i > 1$:

$$M_{i,k,A} = \max_{B,C,i<j<k} \{\mathbf{w}_{A \rightarrow B\ C} \times M_{i,j,B} \times M_{j,k,C}\}$$

- best derivation built by storing B, C, and j for each $M_{i,k,A}$

Weighted CKY Parsing

$$M_{i,k,A} = \max_{B,C,i<j<k} \{\mathbf{w}_{A \rightarrow B\ C} \times M_{i,j,B} \times M_{j,k,C}\}$$



Weighted CKY Pseudo-Code

$\forall A, 0 \leq i, j \leq n \; M_{i,j,A} = 0$;

for $i = 1$ to n **do** {base case: substrings of length 1}

  $M_{i-1,i,A} = \mathbf{w}_{A \rightarrow u_i}$;

{inductive case: substrings of length > 1}

for $l = 2$ to n **do** {l: length of the subtring}

  for $i = 0$ to $n - l$ **do** {i: start position of the substring}

    $k = i + l$; {k: end position of the substring}

    for $j = i + 1$ to $k - 1$ **do** {j: split position}

      for $\forall A \rightarrow BC$ **do**

        $q = \mathbf{w}_{A \rightarrow BC} \times M_{i,j,B} \times M_{j,k,C}$;

        if $q > M_{i,k,A}$ **then**

          $M_{i,k,A} = q$;

          {backpointers to build derivation tree}

          $D_{i,k,A} = \langle j, B, C \rangle$;

Parsing SCFG and Language Modelling

Viterbi Decoding of WSCFGs:

- focus on **most probable** derivation of source (ignoring different target sides associated with the same source side)
- **derivation weights** do not include **language models scores**

? HOW TO EFFICIENTLY COMPUTE TARGET LANGUAGE MODEL SCORES FOR POSSIBLE DERIVATIONS ?

Approaches:

1. **rescoring**: generate k-best candidate translations and rerank k-best list with LM
2. **online**: integrate target m-gram LM scores into dynamic programming parsing
3. **cube pruning** (Huang and Chiang, 2007): rescore k-best sub-translations at each node of the parse forest

Online Translation

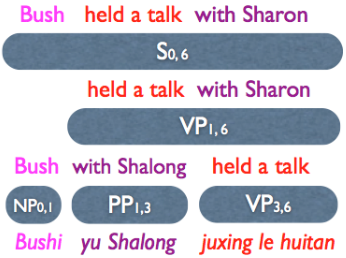
Bàowēier	*yǔ*	*Shālóng*	*jǔxíng*	*le*	*huìtán*
Powell	with	Sharon	hold	[*past*]	meeting

"Powell held a meeting with Sharon"

S	→	$NP^{(1)}\ VP^{(2)}$,	$NP^{(1)}\ VP^{(2)}$
VP	→	$PP^{(1)}\ VP^{(2)}$,	$VP^{(2)}\ PP^{(1)}$
NP	→	*Bàowēier*,	Powell
VP	→	*jǔxíng le huìtán*,	held a meeting
PP	→	*yǔ Shālóng*,	with Sharon

Online Translation: parsing of the source string and building of the corresponding subtranslations **in parallel**

$$\frac{PP_{1,3} : (w_1, t_1)\ \ VP_{3,6} : (w_2, t_2)}{VP_{1,6} : (w \times w_1 \times w_2, t_2 t_1)}$$

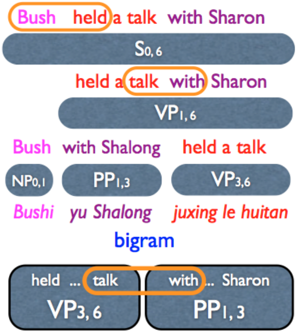


- w_1, w_2: weights of the two antecedents
- w: weight of the synchronous rule
- t_1, t_2: translations

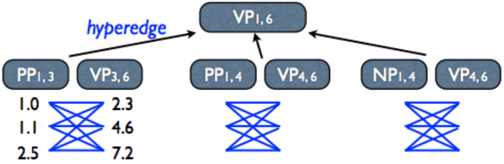
LM Online Integration (Wu, 1996)

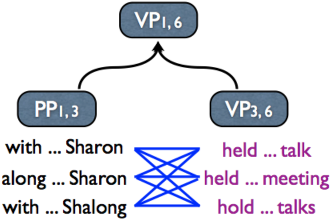
Bigram Online Integration:

$$\frac{PP_{1,3}^{with*Sharon} : (w_1, t_1) \quad VP_{3,6}^{held*talk} : (w_2, t_2)}{VP_{1,6}^{held*Sharon} : (w \times w_1 \times w_2 \times p_{LM}(with|talk), t_2 t_1)}$$



Cube Pruning (Huang and Chiang, 2007)





Beam Search:

- at each step in the derivation, keep at most *k* items integrating target subtranslations in a beam
- enumerate all possible combinations of LM items
- extract the *k*-best combinations

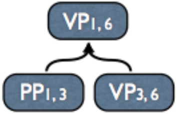
Cube Pruning:

- get *k*-best LM items without **without computing all possible combinations**
- approximate beam search: prone to search errors (in practice, much less significant than efficient decoding)

Cube Pruning

Heuristic Assumption:

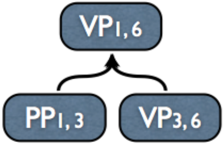
- **best adjacent items** lie towards the **upper-left corner**
- part of the grid can be pruned without computing its cells



non-monotonic grid
due to LM combo costs

		$(PP_{1,3}^{\text{with} \star \text{Sharon}})$	$(PP_{1,3}^{\text{along} \star \text{Sharon}})$	$(PP_{1,3}^{\text{with} \star \text{Shalong}})$
		1.0	3.0	8.0
$(VP_{3,6}^{\text{held} \star \text{meeting}})$	1.0	2.5	9.0	9.5
$(VP_{3,6}^{\text{held} \star \text{talk}})$	1.1	2.4	9.5	9.4
$(VP_{3,6}^{\text{hold} \star \text{conference}})$	3.5	5.1	17.0	12.1

Cube Pruning: Example



bigram (meeting, with)

non-monotonic grid
due to LM combo costs

		$(PP^{with}_{1,3} \star \text{Sharon})$	$(PP^{along}_{1,3} \star \text{Sharon})$	$(PP^{with}_{1,3} \star \text{Shalong})$
		1.0	3.0	8.0
$(VP^{held \star meeting}_{3,6})$	1.0	2.0 + 0.5	4.0 + 5.0	9.0 + 0.5
$(VP^{held \star talk}_{3,6})$	1.1	2.1 + 0.3	4.1 + 5.4	9.1 + 0.3
$(VP^{hold \star conference}_{3,6})$	3.5	4.5 + 0.6	6.5 + 10.5	11.5 + 0.6

Cube Pruning: Example

***k*-best parsing**
(Huang and Chiang, 2005)

- a priority queue of candidates
- extract the best candidate

		$(\mathrm{PP}^{\text{with} \star \text{Sharon}}_{1,3})$	$(\mathrm{PP}^{\text{along} \star \text{Sharon}}_{1,3})$	$(\mathrm{PP}^{\text{with} \star \text{Shalong}}_{1,3})$
		1.0	3.0	8.0
$(\mathrm{VP}^{\text{held} \star \text{meeting}}_{3,6})$	1.0	2.5	9.0	9.5
$(\mathrm{VP}^{\text{held} \star \text{talk}}_{3,6})$	1.1	2.4	9.5	9.4
$(\mathrm{VP}^{\text{hold} \star \text{conference}}_{3,6})$	3.5	5.1	17.0	12.1

Cube Pruning: Example

***k*-best parsing**

(Huang and Chiang, 2005)

- a priority queue of candidates
- extract the best candidate
- push the two successors

		$(\mathrm{PP}^{\text{with} \star \text{Sharon}}_{1,3})$	$(\mathrm{PP}^{\text{along} \star \text{Sharon}}_{1,3})$	$(\mathrm{PP}^{\text{with} \star \text{Shalong}}_{1,3})$
		1.0	3.0	8.0
$(\mathrm{VP}^{\text{held} \star \text{meeting}}_{3,6})$	1.0	2.5	9.0	9.5
$(\mathrm{VP}^{\text{held} \star \text{talk}}_{3,6})$	1.1	2.4	9.5	9.4
$(\mathrm{VP}^{\text{hold} \star \text{conference}}_{3,6})$	3.5	5.1	17.0	12.1

Cube Pruning: Example

k-best parsing
(Huang and Chiang, 2005)

- a priority queue of candidates
- extract the best candidate
- push the two successors

		$(\text{PP}^{\text{with}}_{1,3} \star \text{Sharon})$	$(\text{PP}^{\text{along}}_{1,3} \star \text{Sharon})$	$(\text{PP}^{\text{with}}_{1,3} \star \text{Shalong})$
		1.0	3.0	8.0
$(\text{VP}^{\text{held}}_{3,6} \star \text{meeting})$	1.0	2.5	9.0	9.5
$(\text{VP}^{\text{held}}_{3,6} \star \text{talk})$	1.1	2.4	9.5	9.4
$(\text{VP}^{\text{hold}}_{3,6} \star \text{conference})$	3.5	5.1	17.0	12.1

Cube Pruning: Pseudo-Code

To efficiently compute a small corner of the grid:

- push cost of grid cell 1, 1 onto priority queue
- **repeat** j
  1. extract best cell from queue
  2. push costs of best cell's neighbours onto queue
- **until** k cells have been extracted (other termination conditions are possible)

```
function CUBE(F)                                  ▷ the input is a forest F
    for v ∈ F in (bottom-up) topological order do
        KBEST(v)
    return D_1(TOP)
procedure KBEST(v)
    cand ← {⟨e, 1⟩ | e ∈ IN(v)}                   ▷ for each incoming e
    HEAPIFY(cand)                                 ▷ a priority queue of candidates
    buf ← ∅
    while |cand| > 0 and |buf| < k do
        item ← POP-MIN(cand)
        append item to buf
        PUSHSUCC(item, cand)
    sort buf to D(v)
procedure PUSHSUCC(⟨e, j⟩, cand)
    e is v → u_1 ... u_|e|
    for i in 1 ... |e| do
        j′ ← j + b^i
        if |D(u_i)| ≥ j′_i then
            PUSH(⟨e, j′⟩, cand)
```

Summary

Translation As Parsing:

- Viterbi Approximation
- Weighted CKY Parsing
- Online LM Integration and Cube Pruning

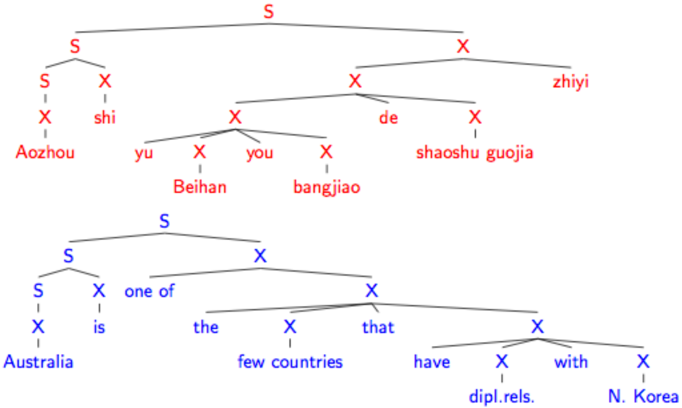
Next Session:

- Learning SCFGs and Hiero

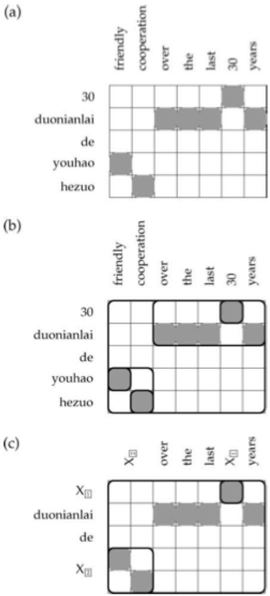
Hierarchical Phrase-Based Models

Hiero (Chiang, 2005, 2007):

- **SCFG of rank 2 with only two nonterminal symbols**
- **discontinuous phrases**
- **long-range reordering rules**



Hiero Synchronous Rules



Rule Extraction:

a **word-aligned** sentence pair

b extract **initial phrase pairs**

c **replace sub-phrases in phrases with symbol** X

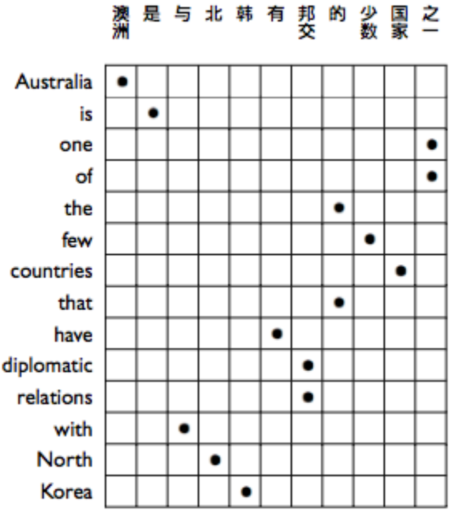
Glue Rules:

$$S \rightarrow S_1 X_2 / S_1 X_2 \quad S \rightarrow X_1 / X_1$$

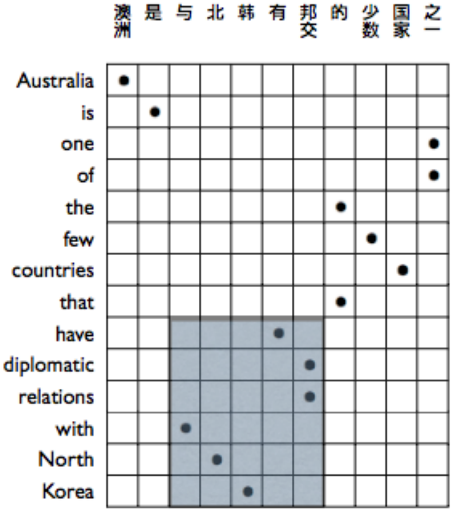
Rule Filtering:

- limited length of initial phrases
- no adjacent nonterminals on source
- at least one pair of aligned words in non-glue rules

Hiero: Rule Extraction

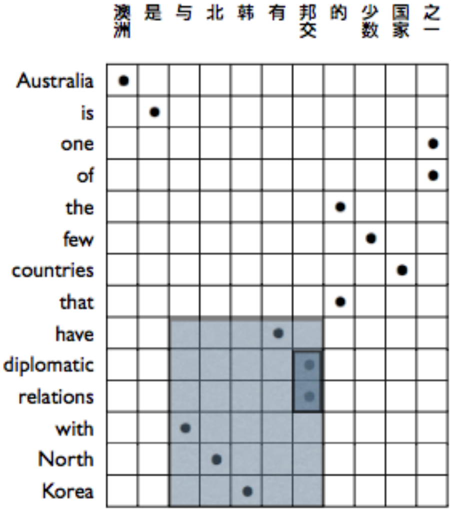


Hiero: Rule Extraction



X → 与 北 韩 有 邦交,
have diplomatic relations
with North Korea

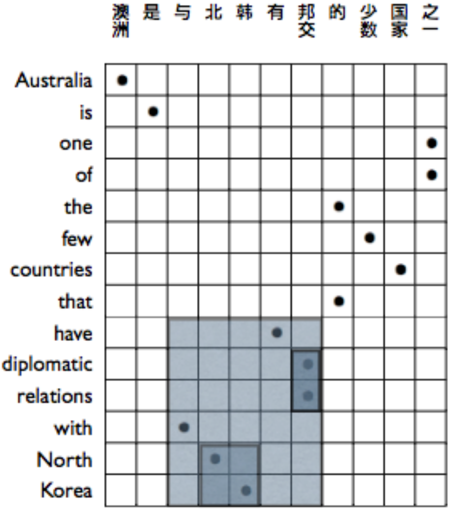
Hiero: Rule Extraction



X → 与 北 韩 有 邦交,
have diplomatic relations
with North Korea

X → 邦交,
diplomatic relations

Hiero: Rule Extraction

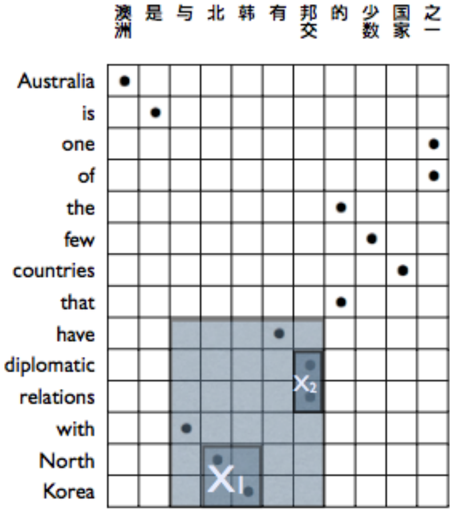


X → 与 北 韩 有 邦交, have diplomatic relations with North Korea

X → 邦交, diplomatic relations

X → 北 韩, North Korea

Hiero: Rule Extraction

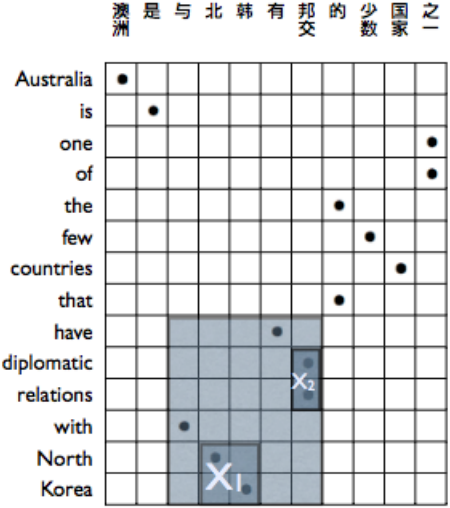


X → 与 北 韩 有 邦交,
have diplomatic relations
with North Korea

X → 邦交,
diplomatic relations

X → 北 韩,
North Korea

Hiero: Rule Extraction



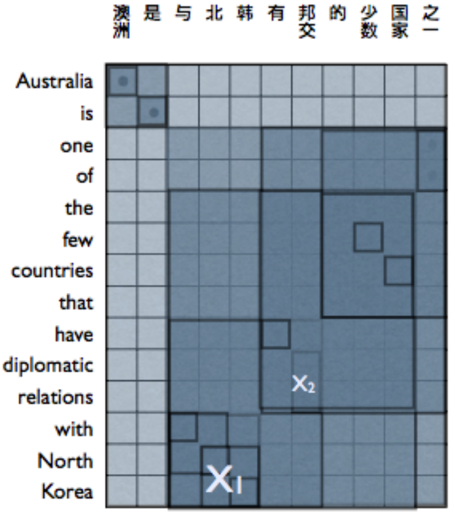
$X \rightarrow$ 与 北 韩 有 邦交, have diplomatic relations with North Korea

$X \rightarrow$ 邦交, diplomatic relations

$X \rightarrow$ 北 韩, North Korea

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

Hiero: Rule Extraction



$X \rightarrow$ 与 北 韩 有 邦交, have diplomatic relations with North Korea

$X \rightarrow$ 邦交, diplomatic relations

$X \rightarrow$ 北 韩, North Korea

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

Hiero: Log-Linear Parametrization

Scoring Rules:

$$\mathcal{S}(A \rightarrow \gamma) = \lambda \cdot h$$

- $h(A \rightarrow \gamma)$: **feature representation** vector $\in \mathbb{R}^m$
- λ: **weight** vector $\in \mathbb{R}^m$
- $h_r(A \rightarrow \gamma)$: **value** of the r-th feature
- λ_r: **weight** of the r-th feature

Scoring Derivations:

$$\mathcal{S}(\pi) = \lambda_{LM} \log p(E(\pi)) + \sum_{\langle Z \rightarrow \gamma, i, j \rangle \in \pi} \mathcal{S}(Z \rightarrow \gamma)$$

- derivation scores decompose into sum of rule scores
- $p(E(\pi))$ is the LM score computed while parsing

Hiero: Feature Representation

Word Translation Features:

$$h_1(X \rightarrow \alpha/\beta) = \log p(\mathcal{T}_\beta|\mathcal{T}_\alpha)$$

$$h_2(X \rightarrow \alpha/\beta) = \log p(\mathcal{T}_\alpha|\mathcal{T}_\beta)$$

Word Penalty Feature:

$$h_3(X \rightarrow \alpha/\beta) = -|\mathcal{T}_\beta|$$

Synchronous Features:

$$h_4(X \rightarrow \alpha/\beta) = \log p(\beta|\alpha)$$

$$h_5(X \rightarrow \alpha/\beta) = \log p(\alpha|\beta)$$

Glue Penalty Feature:

$$h_6(S \rightarrow S_1 X_1 / S_1 X_1) = -1$$

Phrase Penalty Feature:

$$h_7(X \rightarrow \alpha/\beta) = -1$$

- λ_i tuned on dev set using MERT

Summary

Hiero:

- Rule Extraction
- Log-Linear Parametrization
- Feature Representation