

Lecture 4: Semantic Parsing and Machine Translation

Kyle Richardson

kyle@ims.uni-stuttgart.de

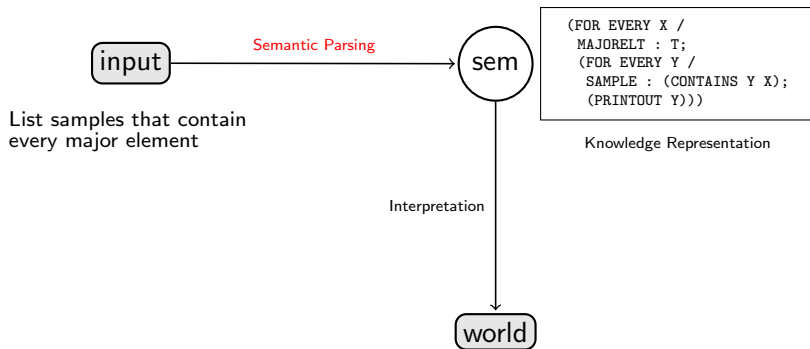
October 31, 2016

Lecture Plan

- ▶ **paper:** Wong and Mooney (2006)
- ▶ **general topics:** Synchronous CFGs, Decoding by parsing, word-alignment and rule extraction.

The Big Picture (reminder)

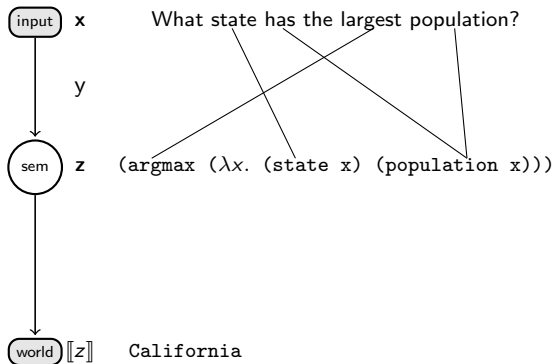
► Standard processing pipeline



$$\llbracket sem \rrbracket = \{S10019, S10059, \dots\}$$

Semantic Parsing: Generating formal representations

- ▶ **Data-driven:** Given data, **learn** a function that can map any given input (x) to a meaning representation (z).
- ▶ What kind of data do we learn from?



Previously: Learning from meaning representations (again)

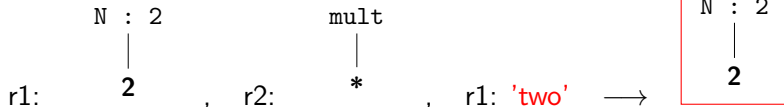
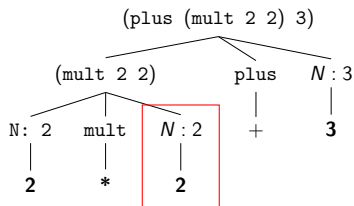
data: ($x = \text{two times two plus three}, y = (\text{plus } (\text{mult } 2 \ 2) \ 3)$)

- ▶ **Compositional model** : a semantic context-free grammar.
- ▶ **Learning Model:** Greedy string \rightarrow tree rule induction (SILT)
- ▶ **Other Topics**
 - ▶ Non-greedy parsing using (P)CFGs and dynamic programming, the CKY algorithm.
 - ▶ Maximum-Likelihood estimation, Expectation Maximization and latent variables, inside-outside probabilities.

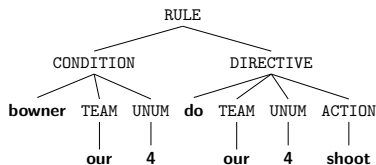
Previous Session: Transformation rules

- Decompose translation into a set of *local* transformations.

data: $(x = \text{two multiplied by two plus three}, y = (\text{plus } (\text{mult } 2 \ 2) \ 3))$



Bottom-up, String → Tree Rule Matching



MR Grammar

RULE	→	CONDITION DIRECTIVE
CONDITION	→	bowner TEAM UNUM
DIRECTIVE	→	do TEAM UNUM ACTION
TEAM	→	our
UNUM	→	4
ACTION	→	shoot

Transformation: If **TEAM** player 4 has the ball, **TEAM** player 4 should shoot.

Input: If **our** player 4 has the ball, **our** player 4 should shoot.

Semantic Parsing and Machine Translation

- ▶ **Conceptually:** problem is treated as a kind of machine translation problem.
 - ▶ **Dataset:** $D = \{(x_i, y_i)\}_{i=1}^n$, x_j sentence, y_j (semantic) translation.

Semantic Parsing and Machine Translation

- ▶ **Conceptually:** problem is treated as a kind of machine translation problem.
 - ▶ **Dataset:** $D = \{(x_i, y_i)\}_{i=1}^n$, x_j sentence, y_j (semantic) translation.
- ▶ **Technically:** transformation rules, common in MT.

Semantic Parsing and Machine Translation

- ▶ **Conceptually:** problem is treated as a kind of machine translation problem.
 - ▶ **Dataset:** $D = \{(x_i, y_i)\}_{i=1}^n$, x_j sentence, y_j (semantic) translation.
- ▶ **Technically:** transformation rules, common in MT.
- ▶ **Idea:** Recast the problem as a statistical MT task.

Semantic Parsing and Machine Translation

- ▶ **Conceptually:** problem is treated as a kind of machine translation problem.
 - ▶ **Dataset:** $D = \{(x_i, y_i)\}_{i=1}^n$, x_j sentence, y_j (semantic) translation.
- ▶ **Technically:** transformation rules, common in MT.
- ▶ **Idea:** Recast the problem as a statistical MT task.
- ▶ **Components:**

Semantic Parsing and Machine Translation

- ▶ **Conceptually:** problem is treated as a kind of machine translation problem.
 - ▶ **Dataset:** $D = \{(x_i, y_i)\}_{i=1}^n$, x_j sentence, y_j (semantic) translation.
- ▶ **Technically:** transformation rules, common in MT.
- ▶ **Idea:** Recast the problem as a statistical MT task.
- ▶ **Components:**
 - ▶ Synchronous grammar model

Semantic Parsing and Machine Translation

- ▶ **Conceptually:** problem is treated as a kind of machine translation problem.
 - ▶ **Dataset:** $D = \{(x_i, y_i)\}_{i=1}^n$, x_j sentence, y_j (semantic) translation.
- ▶ **Technically:** transformation rules, common in MT.
- ▶ **Idea:** Recast the problem as a statistical MT task.
- ▶ **Components:**
 - ▶ Synchronous grammar model
 - ▶ Alignment-based rule extraction

Semantic Parsing and Machine Translation

- ▶ **Conceptually:** problem is treated as a kind of machine translation problem.
 - ▶ **Dataset:** $D = \{(x_i, y_i)\}_{i=1}^n$, x_j sentence, y_j (semantic) translation.
- ▶ **Technically:** transformation rules, common in MT.
- ▶ **Idea:** Recast the problem as a statistical MT task.
- ▶ **Components:**
 - ▶ Synchronous grammar model
 - ▶ Alignment-based rule extraction
 - ▶ Probabilistic decoding and ranking model (more next lecture)

Context-Free Grammars (again)

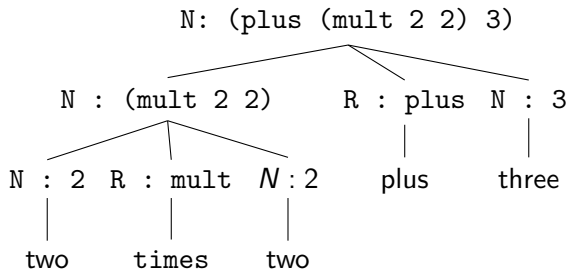
- ▶ **context-free grammar (CFG):**

$$\mathcal{G} = (\Sigma, N, S, R)$$

- ▶ N : set of non-terminal symbols.
 - ▶ Σ : set terminal symbols.
 - ▶ R : set of rules = $\{N \rightarrow \alpha \mid \alpha \in (N \cup \Sigma)^*\}$
 - ▶ S : start symbol
- ▶ **Context-free language:** defines a **set of strings**
- ▶ **Derivation:** A tree representation of rule application on input.
- ▶ **Semantic Parsing:** Semantic representations and composition rules take the form of non-terminal rules in derivations.

Previous examples

- ▶ Derivation trees encode the semantic rules.
- ▶ **example:** $u = \text{two times two plus three}$



language $\mathcal{G} = \{\text{two times two, two times two plus three, ...}\}$

Synchronous Context-Free Grammars (extension)

- ▶ **synchronous context-free grammar (SCFG):**

$$\mathcal{G}^{\text{Syn}} = (\Sigma_e, \Sigma_f, N, S, R)$$

- ▶ N : (shared) set of non-terminal symbols (as before).
- ▶ Σ_e : **english** terminal symbols.
- ▶ Σ_f : **foreign** (or semantic) terminal symbols.
- ▶ R : set of rules of the form:

$$N \rightarrow \langle \alpha, \beta \rangle$$

- ▶ $\alpha \in (N \cup \Sigma_e), \beta \in (N \cup \Sigma_f)$
- ▶ S : start symbol: $\langle S_1, S_2 \rangle$
- ▶ **SCF Language:** defines a **set of string pairs**
- ▶ Allows us to more explicitly relate input and output.

Machine Translation Example

- ▶ **Example:** English \rightarrow Japanese synchronous grammar.
- ▶ **Notation:** subscripts on each non-terminal N are used to relate rules on each side. These rules must be paired in each rule.¹

S	\longrightarrow	$\langle NP_1 VP_2, NP_1 VP_2 \rangle$
VP	\longrightarrow	$\langle V_1 NP_2, NP_2 V_1 \rangle$
NP	\longrightarrow	$\langle I, watashi wa \rangle$
NP	\longrightarrow	$\langle the\ box, hako\ wo \rangle$
V	\longrightarrow	$\langle open, akemasu \rangle$

¹ example from Chiang and Knight (2006)

Machine Translation: Example Derivation

Grammar:

S	\longrightarrow	$\langle NP_1 VP_2, NP_1 VP_2 \rangle$
VP	\longrightarrow	$\langle V_1 NP_2, NP_2 V_1 \rangle$
NP	\longrightarrow	$\langle I, watashi wa \rangle$
NP	\longrightarrow	$\langle the\ box, hako\ wo \rangle$
V	\longrightarrow	$\langle open, akemasu \rangle$

Machine Translation: Example Derivation

Grammar:

$S \longrightarrow \langle NP_1 VP_2, NP_1 VP_2 \rangle$

$VP \longrightarrow \langle V_1 NP_2, NP_2 V_1 \rangle$

$NP \longrightarrow \langle I, watashi \text{ wa} \rangle$

$NP \longrightarrow \langle \text{the box, hako wo} \rangle$

$V \longrightarrow \langle \text{open, akemasu} \rangle$

Derivation

$S \Rightarrow \langle NP_{11} VP_{12}, NP_{11}, VP_{12} \rangle$

Machine Translation: Example Derivation

Grammar:

S	\longrightarrow	$\langle NP_1 VP_2, NP_1 VP_2 \rangle$
VP	\longrightarrow	$\langle V_1 NP_2, NP_2 V_1 \rangle$
NP	\longrightarrow	$\langle I, watashi \text{ wa} \rangle$
NP	\longrightarrow	$\langle \text{the box}, hako \text{ wo} \rangle$
V	\longrightarrow	$\langle \text{open}, akemasu \rangle$

Derivation

S	\Rightarrow	$\langle NP_{11} VP_{12}, NP_{11}, VP_{12} \rangle$
	\Rightarrow	$\langle NP_{11} V_{13} NP_{14}, NP_{11} NP_{14} V_{13} \rangle$

Machine Translation: Example Derivation

Grammar:

S	\longrightarrow	$\langle NP_1 VP_2, NP_1 VP_2 \rangle$
VP	\longrightarrow	$\langle V_1 NP_2, NP_2 V_1 \rangle$
NP	\longrightarrow	$\langle I, watashi \text{ wa} \rangle$
NP	\longrightarrow	$\langle \text{the box, hako wo} \rangle$
V	\longrightarrow	$\langle \text{open, akemasu} \rangle$

Derivation

S	\Rightarrow	$\langle NP_{11} VP_{12}, NP_{11}, VP_{12} \rangle$
	\Rightarrow	$\langle NP_{11} V_{13} NP_{14}, NP_{11} NP_{14} V_{13} \rangle$
	\Rightarrow	$\langle I V_{13} NP_{14}, \text{watashi wa } NP_{14} V_{13} \rangle$

Machine Translation: Example Derivation

Grammar:

S	\longrightarrow	$\langle NP_1 VP_2, NP_1 VP_2 \rangle$
VP	\longrightarrow	$\langle V_1 NP_2, NP_2 V_1 \rangle$
NP	\longrightarrow	$\langle I, watashi \text{ wa} \rangle$
NP	\longrightarrow	$\langle \text{the box, hako wo} \rangle$
V	\longrightarrow	$\langle \text{open, akemasu} \rangle$

Derivation

S	\Rightarrow	$\langle NP_{11} VP_{12}, NP_{11}, VP_{12} \rangle$
	\Rightarrow	$\langle NP_{11} V_{13} NP_{14}, NP_{11} NP_{14} V_{13} \rangle$
	\Rightarrow	$\langle I V_{13} NP_{14}, \text{watashi wa } NP_{14} V_{13} \rangle$
	\Rightarrow	$\langle I \text{ open } NP_{14}, \text{watashi wa } NP_{14} \text{ akemasu} \rangle$

Machine Translation: Example Derivation

Grammar:

S	\longrightarrow	$\langle NP_1 VP_2, NP_1 VP_2 \rangle$
VP	\longrightarrow	$\langle V_1 NP_2, NP_2 V_1 \rangle$
NP	\longrightarrow	$\langle I, watashi \text{ wa} \rangle$
NP	\longrightarrow	$\langle \text{the box}, hako \text{ wo} \rangle$
V	\longrightarrow	$\langle \text{open}, akemasu \rangle$

Derivation

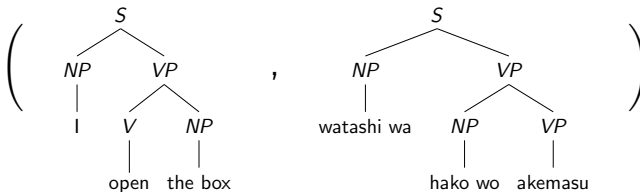
S	\Rightarrow	$\langle NP_{11} VP_{12}, NP_{11}, VP_{12} \rangle$
	\Rightarrow	$\langle NP_{11} V_{13} NP_{14}, NP_{11} NP_{14} V_{13} \rangle$
	\Rightarrow	$\langle I V_{13} NP_{14}, watashi \text{ wa} NP_{14} V_{13} \rangle$
	\Rightarrow	$\langle I \text{ open } NP_{14}, watashi \text{ wa} NP_{14} \text{ akemasu} \rangle$
	\Rightarrow	$\langle I \text{ open the box}, watashi \text{ wa} hako \text{ wo} \text{ akemasu} \rangle$

SCFGs

- **SCFG language:** defines a set of sentence pairs

$$\mathcal{G}^{Syn} = \{(I \text{ open the box}, \text{watashi wa hako wo akemasu}), \dots\}$$

- **derivation:** a pair of trees.



Two Variants of Parsing

- ▶ **Parsing pairs:** Given an english text and foreign text, generate a synchronous derivation using a grammar \mathcal{G}^{Syn} (*bitext parsing*)

(I open the box, watashi wa hako wo akemasu) \rightarrow derivation

- ▶ **Translation or Decoding:** Given an english text, translate it into a foreign text using a grammar \mathcal{G}^{Syn}

I open the box \rightarrow watashi wa hako wo akemasu

Two Variants of Parsing

- ▶ **Parsing pairs:** Given an english text and foreign text, generate a synchronous derivation using a grammar \mathcal{G}^{Syn} (*bitext parsing*)

(I open the box, watashi wa hako wo akemasu) \rightarrow derivation

- ▶ **Translation or Decoding:** Given an english text, translate it into a foreign text using a grammar \mathcal{G}^{Syn}

I open the box \rightarrow watashi wa hako wo akemasu

- ▶ **Surprisingly:** The first problem is much harder than the second (despite more information). We will only consider the second.

Decoding by parsing (i.e., Translation)

- ▶ Assuming we have binary rules, we can use the CKY algorithm (last lecture) for parsing.
- ▶ **Idea:** Parse the english side of the grammar in the normal way, then apply or *project* foreign side of rules.
 - ▶ **Why does this work?** Synchronous rules have the same LHSs.

Decoding by Parsing: Parse English Side

Grammar:

S \longrightarrow $\langle NP_1 VP_2, NP_1 VP_2 \rangle$

VP \longrightarrow $\langle V_1 NP_2, NP_2 V_1 \rangle$

NP \longrightarrow $\langle I, watashi wa \rangle$

NP \longrightarrow $\langle the\ box, hako\ wo \rangle$

V \longrightarrow $\langle open, akemasu \rangle$

₀ I ₁ open ₂ the box ₃

	1	2	3
0			
1			
2			

Decoding by Parsing: Parse English Side

Grammar:

S \longrightarrow $\langle NP_1 VP_2, NP_1 VP_2 \rangle$

VP \longrightarrow $\langle V_1 NP_2, NP_2 V_1 \rangle$

NP \longrightarrow $\langle I, watashi wa \rangle$

NP \longrightarrow $\langle the\ box, hako\ wo \rangle$

V \longrightarrow $\langle open, akemasu \rangle$

₀ I ₁ open ₂ the box ₃

	1	2	3
0	NP \rightarrow I		
1			
2			

Decoding by Parsing: Parse English Side

Grammar:

$S \longrightarrow \langle NP_1 VP_2, NP_1 VP_2 \rangle$

$VP \longrightarrow \langle V_1 NP_2, NP_2 V_1 \rangle$

$NP \longrightarrow \langle I, watashi wa \rangle$

$NP \longrightarrow \langle \text{the box}, hako wo \rangle$

$V \longrightarrow \langle \text{open}, akemasu \rangle$

$_0$ I $_1$ open $_2$ the box $_3$

	1	2	3
0	NP \rightarrow I		
1		V \rightarrow open	
2			

Decoding by Parsing: Parse English Side

Grammar:

$S \longrightarrow \langle NP_1 VP_2, NP_1 VP_2 \rangle$

$VP \longrightarrow \langle V_1 NP_2, NP_2 V_1 \rangle$

$NP \longrightarrow \langle I, watashi wa \rangle$

$NP \longrightarrow \langle \text{the box}, hako wo \rangle$

$V \longrightarrow \langle \text{open}, akemasu \rangle$

$_0$ I $_1$ open $_2$ the box $_3$

	1	2	3
0	NP \rightarrow I		
1		V \rightarrow open	
2			NP \rightarrow the box

Decoding by Parsing: Parse English Side

Grammar:

$S \longrightarrow \langle NP_1 VP_2, NP_1 VP_2 \rangle$

$VP \longrightarrow \langle V_1 NP_2, NP_2 V_1 \rangle$

$NP \longrightarrow \langle I, watashi wa \rangle$

$NP \longrightarrow \langle \text{the box}, hako wo \rangle$

$V \longrightarrow \langle \text{open}, akemasu \rangle$

0 | 1 open 2 the box 3

	1	2	3
0	NP → I		
1		V → open	VP → V NP
2			NP → the box

Decoding by Parsing: Parse English Side

Grammar:

$S \longrightarrow \langle NP_1 VP_2, NP_1 VP_2 \rangle$

$VP \longrightarrow \langle V_1 NP_2, NP_2 V_1 \rangle$

$NP \longrightarrow \langle I, watashi wa \rangle$

$NP \longrightarrow \langle \text{the box}, hako wo \rangle$

$V \longrightarrow \langle \text{open}, akemasu \rangle$

0 I 1 open 2 the box 3

	1	2	3
0	NP \rightarrow I		S \rightarrow NP VP
1		V \rightarrow open	VP \rightarrow V NP
2			NP \rightarrow the box

Decoding by Parsing: Projection

Grammar:

$S \longrightarrow \langle NP_1 VP_2, NP_1 VP_2 \rangle$

$VP \longrightarrow \langle V_1 NP_2, NP_2 V_1 \rangle$

$NP \longrightarrow \langle I, watashi wa \rangle$

$NP \longrightarrow \langle \text{the box}, hako wo \rangle$

$V \longrightarrow \langle \text{open}, akemasu \rangle$

$_0$ I $_1$ open $_2$ the box $_3$

	1	2	3
0	NP \rightarrow I, <i>watashi wa</i>		S \rightarrow NP VP, <i>NP VP</i>
1		V \rightarrow open <i>akemasu</i>	VP \rightarrow V NP, <i>NP V</i>
2			NP \rightarrow the box <i>hako wo</i>

Binarization (brief reminder/review)

- ▶ CKY algorithm (last week) assumes input grammar is in Chomsky normal-form (binary rules and unary pre-terminal rules only).
- ▶ **Why? input:** $w_1 w_2 w_3 w_4$

Binarization (brief reminder/review)

- ▶ CKY algorithm (last week) assumes input grammar is in Chomsky normal-form (binary rules and unary pre-terminal rules only).
- ▶ **Why? input:** $w_1 w_2 w_3 w_4$

binary (one split)

(w_1, w_2)

(w_2, w_3)

(w_3, w_4)

$(w_1, w_2 w_3)$

$(w_1 w_2, w_3)$

$(w_2 w_3, w_4)$

...

binary+ternary (two splits)

(w_1, w_2)

(w_2, w_3)

(w_3, w_4)

$(w_1, w_2 w_3)$

$(w_1 w_2, w_3)$

$(w_2 w_3, w_4)$

...

(w_1, w_2, w_3)

$(w_1 w_2, w_3, w_4)$

...

Binarization (brief reminder/review)

- ▶ CKY algorithm (last week) assumes input grammar is in Chomsky normal-form (binary rules and unary pre-terminal rules only).
- ▶ **Why? input:** $w_1 w_2 w_3 w_4$

binary (one split)

(w_1, w_2)

(w_2, w_3)

(w_3, w_4)

$(w_1, w_2 w_3)$

$(w_1 w_2, w_3)$

$(w_2 w_3, w_4)$

...

binary+ternary (two splits)

(w_1, w_2)

(w_2, w_3)

(w_3, w_4)

$(w_1, w_2 w_3)$

$(w_1 w_2, w_3)$

$(w_2 w_3, w_4)$

...

(w_1, w_2, w_3)

$(w_1 w_2, w_3, w_4)$

...

- ▶ **Problem:** Unlike normal CFGs, SCFGs cannot be binarized in the general case.

History: Syntax-Directed Translation

- ▶ First developed as a method for programming language compilation (i.e. translating high-level languages to lower-level languages)

$$\left(\begin{array}{l} \text{for } i \text{ in range}(10): \\ \quad n += i \end{array} \right. , \left. \begin{array}{l} \text{move ax, 1} \\ \text{loop: add bx, ax} \\ \text{cmp ax, 10} \\ \text{jle loop} \end{array} \right)$$

- ▶ **Analogy:** We can think of semantic parsing as a form of language compilation.

Big Idea: Wong and Mooney (2006)

- ▶ **Transformation Rules:** recast the string-to-tree rewrite rules (last class, Kate et al. (2005)) as synchronous grammars rules.
- ▶ **Rule Extraction:** SCFGs are extracted using a word alignment model (as done in other approaches to MT)

Semantic Parsing and Syntax-driven Translation

Grammar:

RULE	→	$\langle \text{if } \text{CONDITION}_1 \text{ DIRECTIVE}_2, (\text{CONDITION}_1 \text{ DIRECTIVE}_2) \rangle$
CONDITION	→	$\langle \text{TEAM}_1 \text{ player UNUM}_2 \text{ has the ball }, (\text{bowner TEAM}_1 \{ \text{UNUM} \}_2) \rangle$
TEAM	→	$\langle \text{our}, \text{our} \rangle$
UNUM	→	$\langle \text{four}, 4 \rangle$

Semantic Parsing and Syntax-driven Translation

Grammar:

RULE	\longrightarrow	$\langle \text{if } \text{CONDITION}_1 \text{ DIRECTIVE}_2, (\text{CONDITION}_1 \text{ DIRECTIVE}_2) \rangle$
CONDITION	\longrightarrow	$\langle \text{TEAM}_1 \text{ player UNUM}_2 \text{ has the ball }, (\text{bowner TEAM}_1 \{ \text{UNUM} \}_2) \rangle$
TEAM	\longrightarrow	$\langle \text{our}, \text{our} \rangle$
UNUM	\longrightarrow	$\langle \text{four}, 4 \rangle$

Deriv.

RULE $\Rightarrow \langle \text{if } \text{CONDITION}_1 \text{ DIRECTIVE}_2, (\text{CONDITION}_1, \text{DIRECTIVE}_2) \rangle$

Semantic Parsing and Syntax-driven Translation

Grammar:

RULE	→	$\langle \text{if } \text{CONDITION}_1 \text{ DIRECTIVE}_2, (\text{CONDITION}_1 \text{ DIRECTIVE}_2) \rangle$
CONDITION	→	$\langle \text{TEAM}_1 \text{ player UNUM}_2 \text{ has the ball }, (\text{bowler TEAM}_1 \{ \text{UNUM} \}_2) \rangle$
TEAM	→	$\langle \text{our}, \text{our} \rangle$
UNUM	→	$\langle \text{four}, 4 \rangle$

Deriv.

RULE	⇒	$\langle \text{if } \text{CONDITION}_1 \text{ DIRECTIVE}_2, (\text{CONDITION}_1, \text{DIRECTIVE}_2) \rangle$
	⇒	$\langle \text{if } \text{TEAM}_1 \text{ player UNUM}_2 \text{ has the ball DIR.}_2, ((\text{bowler TEAM}_1 \{ \text{UNUM}_2 \}, \text{DIR}_2) \rangle$

Semantic Parsing and Syntax-driven Translation

Grammar:

RULE	→	$\langle \text{if } \text{CONDITION}_1 \text{ DIRECTIVE}_2, (\text{CONDITION}_1 \text{ DIRECTIVE}_2) \rangle$
CONDITION	→	$\langle \text{TEAM}_1 \text{ player UNUM}_2 \text{ has the ball }, (\text{bowner TEAM}_1 \{ \text{UNUM} \}_2) \rangle$
TEAM	→	$\langle \text{our}, \text{our} \rangle$
UNUM	→	$\langle \text{four}, 4 \rangle$

Deriv.

RULE	⇒	$\langle \text{if } \text{CONDITION}_1 \text{ DIRECTIVE}_2, (\text{CONDITION}_1, \text{DIRECTIVE}_2) \rangle$
	⇒	$\langle \text{if } \text{TEAM}_1 \text{ player UNUM}_2 \text{ has the ball DIR.}_2, ((\text{bowler TEAM}_1 \{ \text{UNUM}_2 \}, \text{DIR}_2) \rangle$
	⇒	$\langle \text{if } \text{our} \text{ player UNUM}_2 \text{ has the ball DIR.}_2, ((\text{bowler our } \{ \text{UNUM}_2 \}, \text{DIR}_2) \rangle$

Semantic Parsing and Syntax-driven Translation

Grammar:

RULE	→	$\langle \text{if } \text{CONDITION}_1 \text{ DIRECTIVE}_2, (\text{CONDITION}_1 \text{ DIRECTIVE}_2) \rangle$
CONDITION	→	$\langle \text{TEAM}_1 \text{ player UNUM}_2 \text{ has the ball }, (\text{bowner } \text{TEAM}_1 \{ \text{UNUM} \}_2) \rangle$
TEAM	→	$\langle \text{our}, \text{our} \rangle$
UNUM	→	$\langle \text{four}, 4 \rangle$

Deriv.

RULE \Rightarrow $\langle \text{if } \text{CONDITION}_1 \text{ DIRECTIVE}_2, (\text{CONDITION}_1, \text{DIRECTIVE}_2) \rangle$
 \Rightarrow $\langle \text{if } \text{TEAM}_1 \text{ player UNUM}_2 \text{ has the ball DIR.}_2, ((\text{bowler } \text{TEAM}_1 \{ \text{UNUM}_2 \}, \text{DIR}_2) \rangle$
 \Rightarrow $\langle \text{if } \text{our} \text{ player UNUM}_2 \text{ has the ball DIR.}_2, ((\text{bowler } \text{our} \{ \text{UNUM}_2 \}, \text{DIR}_2) \rangle$
...
 \Rightarrow $\langle \text{ If our player four has the ball, then our player six ... , } \\ ((\text{bowner our } \{4\})(\text{do our } \{6\} (\text{pos (left (half our)))))) \rangle$

Semantic Parsing and Syntax-driven Translation

Grammar:

RULE	\longrightarrow	$\langle \text{if } \text{CONDITION}_1 \text{ DIRECTIVE}_2, (\text{CONDITION}_1 \text{ DIRECTIVE}_2) \rangle$
CONDITION	\longrightarrow	$\langle \text{TEAM}_1 \text{ player UNUM}_2 \text{ has the ball }, (\text{bowner TEAM}_1 \{ \text{UNUM} \}_2) \rangle$
TEAM	\longrightarrow	$\langle \text{our}, \text{our} \rangle$
UNUM	\longrightarrow	$\langle \text{four}, 4 \rangle$

Deriv.

RULE \Rightarrow $\langle \text{if } \text{CONDITION}_1 \text{ DIRECTIVE}_2, (\text{CONDITION}_1, \text{DIRECTIVE}_2) \rangle$
 \Rightarrow $\langle \text{if } \text{TEAM}_1 \text{ player UNUM}_2 \text{ has the ball DIR.}_2, ((\text{bowler TEAM}_1 \{ \text{UNUM}_2 \}, \text{DIR}_2)) \rangle$
 \Rightarrow $\langle \text{if } \text{our} \text{ player UNUM}_2 \text{ has the ball DIR.}_2, ((\text{bowler our } \{ \text{UNUM}_2 \}, \text{DIR}_2)) \rangle$
...
 \Rightarrow $\langle \text{If our player four has the ball, then our player six ... , } \\ ((\text{bowner our } \{4\})(\text{do our } \{6\} (\text{pos (left (half our))))) \rangle$

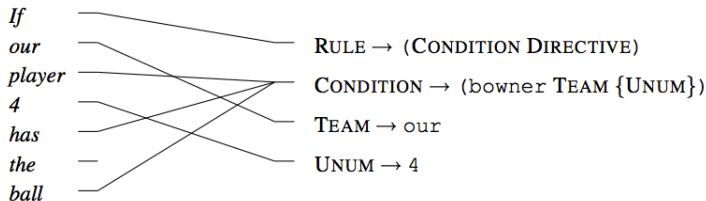
► Is this grammar in CNF?

Rule Extraction and Alignment

- ▶ **Lexical Acquisition:** finding optimal *word alignments* between NL sentences and meaning representation (MR) fragments.
- ▶ Assumes (as in Kate et al. (2005)) a deterministic MR grammar.
 - ▶ For alignment, MR is represented as a sequence of productions.

Rule Extraction and Alignment

- ▶ **Lexical Acquisition:** finding optimal *word alignments* between NL sentences and meaning representation (MR) fragments.
- ▶ Assumes (as in Kate et al. (2005)) a deterministic MR grammar.
 - ▶ For alignment, MR is represented as a sequence of productions.



Word-based alignment models (basics)

- **Basic idea:** Treat translation as a process of translating individual *words*²

Das	Haus	ist	klein
the	house	is	small

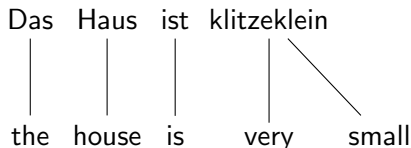
- **Alignment function:** $a : i \rightarrow j$, (i english word to j foreign word)

$a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 4\}$

²Examples from Koehn (2009) and some of his slides.

Word-based alignment models (basics)

- **Basic idea:** Treat translation as a process of translating individual *words*³



- **Alignment function:** $a : i \rightarrow j$, (i english word to j foreign word)

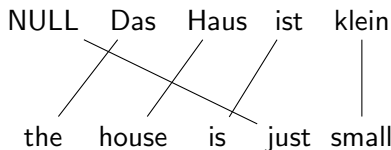
$$a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 4, 5 \rightarrow 4\}$$

- **One-to-many:** foreign might translate to multiple english words.

³Examples from Koehn (2009)

Word-based alignment models (basics)

- **Basic idea:** Treat translation as a process of translating individual *words*⁴



- **Alignment function:** $a : i \rightarrow j$, (i english word to j foreign word)

$$a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 0, 5 \rightarrow 4\}$$

- **Null translation:** english words might not have foreign translations.

⁴Examples from Koehn (2009)

Word-based alignment models (basics)

- **Translation probability:** defined as $t(e_i | f_j)$, or probability of english word e_i given a foreign word f_j , s.t.

$$\sum_e t(e | f_j) = 1.0$$

Word-based alignment models (basics)

- **Translation probability:** defined as $t(e_i | f_j)$, or probability of english word e_i given a foreign word f_j , s.t.

$$\sum_e t(e | f_j) = 1.0$$

$$t(. | \text{klein}) = \begin{cases} 0.5 & e = \text{small} \\ 0.2 & e = \text{tiny} \\ 0.2 & e = \text{little} \\ 0.05 & e = \text{the} \\ 0.05 & e = \text{house} \end{cases}$$

IBM Model 1

- ▶ **IBM Model 1:** Based entirely on translation (or lexical) probabilities (Brown et al. (1993)).
 - ▶ **english sentence:** e_1, \dots, e_{l_e}
 - ▶ **foreign sentence:** f_1, \dots, f_{l_f}

IBM Model 1

- ▶ **IBM Model 1:** Based entirely on translation (or lexical) probabilities (Brown et al. (1993)).
 - ▶ **english sentence:** e_1, \dots, e_{l_e}
 - ▶ **foreign sentence:** f_1, \dots, f_{l_f}
- ▶ **Translation probability with alignment:**

$$p(e, a | f) = \frac{1}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j | f_{a(j)})$$

- ▶ $(l_f + 1)^{l_e}$, the number of total alignments (assuming Null word).

IBM Model 1

- Translation probability with alignment:

$$p(e, a | f) = \frac{1}{(I_f + 1)^{I_e}} \prod_{j=1}^{I_e} t(e_j | f_{a(j)})$$

Das	Haus	ist	klein
the	house	is	small

- $a : \{1 \rightarrow 1_{t(the|Das)=0.7}, 2 \rightarrow 2_{t(house|Haus)=0.8}, 3 \rightarrow 3...0.8, 4 \rightarrow 4...0.4\}$

$$p(e, a | f) = \frac{1}{5^4} * 0.7 * 0.8 * 0.8 * 0.4 = 0.0029$$

IBM Model 1

- ▶ **Translation probability with alignment:**

$$p(e, a | f) = \frac{1}{(I_f + 1)^{I_e}} \prod_{j=1}^{I_e} t(e_j | f_{a(j)})$$

- ▶ **(Overall) Translation probability:**

$$p(e | f) = \sum_a p(e, a | f)$$

IBM Model 1

- ▶ **Translation probability with alignment:**

$$p(e, a | f) = \frac{1}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j | f_{a(j)})$$

- ▶ **(Overall) Translation probability:**

$$p(e | f) = \sum_a p(e, a | f)$$

- ▶ **Problem:** Requires summing over all alignments
- ▶ e.g., $l_e = l_f = 10$ this equals $(10 + 1)^{10} = 25,937,424,601$ alignments (Penn treebank, aver. somewhere near 27 words).

IBM Model 1

- ▶ Luckily, we can get around this (using some basic math).
- ▶ **(Overall) Translation probability:**

$$\begin{aligned} p(e | f) &= \sum_a p(e, a | f) \\ &= \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)=0}^{l_f} p(e, a | f) \\ &= \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)=0}^{l_f} \frac{1}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j | f_{a(j)}) \\ &= \frac{1}{(l_f + 1)^{l_e}} \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)=0}^{l_f} \prod_{j=1}^{l_e} t(e_j | f_{a(j)}) \\ &= \frac{1}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j | f_i) \end{aligned}$$

IBM Model 1

- ▶ Luckily, we can get around this (using some basic math).
- ▶ **(Overall) Translation probability:**

$$\begin{aligned} p(e \mid f) &= \sum_a p(e, a \mid f) \\ &= \frac{1}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j \mid f_j) \end{aligned}$$

- ▶ $e = \text{my friend}$, $f = \text{mein freund}$ (without Null)

$$p(\text{my friend} \mid \text{mein freund}) = ((t(\text{my} \mid \text{mein}) + t(\text{my} \mid \text{freund})) * (t(\text{friend} \mid \text{mein}) + t(\text{friend} \mid \text{freund}))) / 2^2$$

Learning a Model1 aligner

- ▶ Requires learning translation probabilities $t(e_i | f_j)$
- ▶ **Maximum Likelihood Estimation (MLE)** (with full information)

$$t(e_i, f_j) = \frac{\text{count}(e_i, f_j)}{\sum_e \text{count}(e, f_j)}$$

Learning a Model1 aligner

- ▶ Requires learning translation probabilities $t(e_i | f_j)$
- ▶ **Maximum Likelihood Estimation (MLE)** (with full information)

$$t(e_i, f_j) = \frac{\text{count}(e_i, f_j)}{\sum_e \text{count}(e, f_j)}$$

- ▶ **Problem:** We don't have full information (i.e, target alignments)
- ▶ **Expection Maximization (EM):**

Learning a Model1 aligner

- ▶ Requires learning translation probabilities $t(e_i | f_j)$
- ▶ **Maximum Likelihood Estimation (MLE)** (with full information)

$$t(e_i, f_j) = \frac{\text{count}(e_i, f_j)}{\sum_e \text{count}(e, f_j)}$$

- ▶ **Problem:** We don't have full information (i.e, target alignments)
- ▶ **Expectation Maximization (EM):**
 - ▶ Initialize parameters randomly (or uniformly)

Learning a Model1 aligner

- ▶ Requires learning translation probabilities $t(e_i | f_j)$
- ▶ **Maximum Likelihood Estimation (MLE)** (with full information)

$$t(e_i, f_j) = \frac{\text{count}(e_i, f_j)}{\sum_e \text{count}(e, f_j)}$$

- ▶ **Problem:** We don't have full information (i.e, target alignments)
- ▶ **Expectation Maximization (EM):**
 - ▶ Initialize parameters randomly (or uniformly)
 - ▶ **e-step:** Run the current model on your data, collect counts.

Learning a Model1 aligner

- ▶ Requires learning translation probabilities $t(e_i | f_j)$
- ▶ **Maximum Likelihood Estimation (MLE)** (with full information)

$$t(e_i, f_j) = \frac{\text{count}(e_i, f_j)}{\sum_e \text{count}(e, f_j)}$$

- ▶ **Problem:** We don't have full information (i.e, target alignments)
- ▶ **Expectation Maximization (EM):**
 - ▶ Initialize parameters randomly (or uniformly)
 - ▶ **e-step:** Run the current model on your data, collect counts.
 - ▶ **m-step:** Update parameters based on previous step.

Learning a Model1 aligner

- ▶ Requires learning translation probabilities $t(e_i | f_j)$
- ▶ **Maximum Likelihood Estimation (MLE)** (with full information)

$$t(e_i, f_j) = \frac{\text{count}(e_i, f_j)}{\sum_e \text{count}(e, f_j)}$$

- ▶ **Problem:** We don't have full information (i.e, target alignments)
- ▶ **Expectation Maximization (EM):**
 - ▶ Initialize parameters randomly (or uniformly)
 - ▶ **e-step:** Run the current model on your data, collect counts.
 - ▶ **m-step:** Update parameters based on previous step.
 - ▶ Repeat last two steps until convergence.

EM for IBM Model1

```
Input: set of sentence pairs (e,f)
Output: translation prob. t(e|f)
1: initialize t(e|f) uniformly
2: while not converged do
3:   // initialize
4:   count(e|f) = 0 for all e,f
5:   total(f) = 0 for all f
6:   for all sentence pairs (e,f) do
7:     // compute normalization
8:     for all words e in e do
9:       s-total(e) = 0
10:      for all words f in f do
11:        s-total(e) += t(e|f)
12:      end for
13:    end for
14:    // collect counts
15:    for all words e in e do
16:      for all words f in f do
17:        count(e|f) +=  $\frac{t(e|f)}{s-total(e)}$ 
18:        total(f) +=  $\frac{t(e|f)}{s-total(e)}$ 
19:      end for
20:    end for
21:  end for
22:  // estimate probabilities
23:  for all foreign words f do
24:    for all English words e do
25:      t(e|f) =  $\frac{count(e|f)}{total(f)}$ 
26:    end for
27:  end for
28: end while
```

Koehn (2009)

Model1 as a Translation Model

- ▶ **Word decoding** : Model1 can be used as translation model.

$$p(e \mid f) = \sum_a p(e, a \mid f)$$

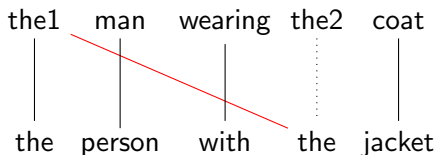
- ▶ Nowadays, such models are used for extracting alignments, which are the basis of more complex translation models (e.g. our syntax-based model).
- ▶ **Viterbi alignment**: Find the most likely alignment given a pair (easy, find for each word e_i the most likely f_j)

$$a_i = \arg \max_{j \in \{0 \dots l_f\}} t(e_i \mid f_j)$$

- ▶ **K-best alignments**: Can be extended to extract top k alignments.

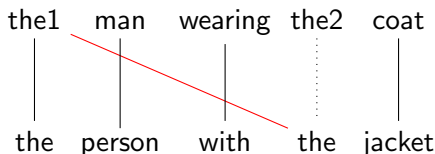
Other IBM Models

- **IBM Models 2-5:** Go beyond using only the lexical translation probabilities.



Other IBM Models

- **IBM Models 2-5:** Go beyond using only the lexical translation probabilities.

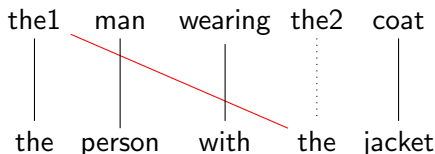


- **IBM Model2:** adds an *alignment probability distribution*: $a(i | j, l_e, l_f)$, which considers relative word position and sentence length:

$$p(e, a | f) = \prod_{j=1}^{l_e} t(e_j | f_{a(j)}) a(a(j) | j, l_e, l_f)$$

Other IBM Models

- **IBM Models 2-5:** Go beyond using only the lexical translation probabilities.



- **IBM Model2:** adds an *alignment probability distribution*: $a(i | j, l_e, l_f)$, which considers relative word position and sentence length:

$$p(e, a | f) = \prod_{j=1}^{l_e} t(e_j | f_{a(j)}) a(a(j) | j, l_e, l_f)$$

- **Model1:** $t(e_4 | f_1) = t(e_4 | f_4)$
- **Model2:** $t(e_4 | f_1) a(1 | 4, 5, 5) < t(e_4 | f_4) a(4 | 4, 5, 5)$

Other IBM Models

- ▶ **Model1:** lexical translation probabilities, bag-of-words.
- ▶ **Model2:** alignment probability distribution: $a(i | j, l_e, l_f)$
- ▶ **Model3:** *fertility distribution* $n(\phi | f)$, or distribution over the number of words each f_j usually translates to.

$$n(1 | \text{haus}) = 1.0, n(2 | \text{klitzeklein}) = 1.0, \dots$$

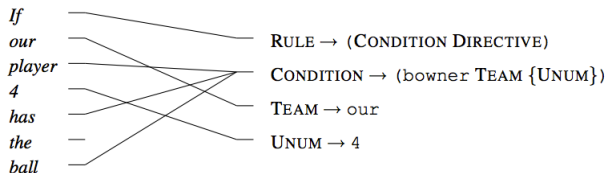
- ▶ **Model4:** relative distortion, word classes.
- ▶ **Model5:** fixes deficiency problem.

Back to Semantic Parsing: Rule extraction (Wong and Mooney (2006))

- ▶ **Extraction:** Train IBM Model5 over english sentences and sequences of MR productions, and extract rules from 10-best alignment.
 - ▶ **Important:** productions are used instead of MR tokens, allows for skipping pieces without meaning.

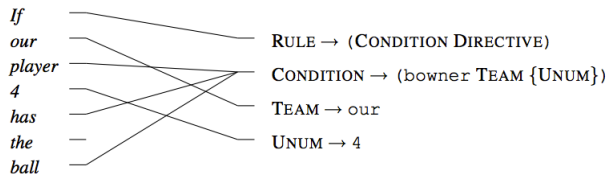
Back to Semantic Parsing: Rule extraction (Wong and Mooney (2006))

- ▶ **Extraction:** Train IBM Model5 over english sentences and sequences of MR productions, and extract rules from 10-best alignment.
- ▶ **Important:** productions are used instead of MR tokens, allows for skipping pieces without meaning.



Rule extraction (Wong and Mooney (2006))

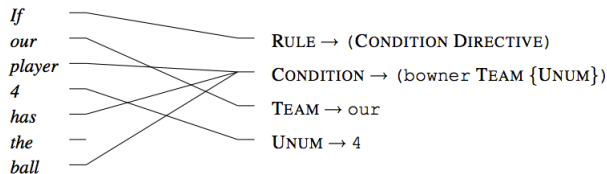
- **Extraction:** Bottom-up (as done last week), starting from alignments with terminal symbols, then working to more complex rules.



- alignment where RHS of production rule is a MR terminal:
 - **TEAM** → ⟨our, our⟩, **UNUM** → ⟨4, 4⟩, ...

Rule extraction (Wong and Mooney (2006))

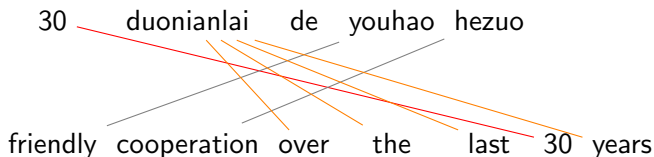
- ▶ **Extraction:** Bottom-up (as done last week), starting from alignments with terminal symbols, then working to more complex rules.



- ▶ alignment where RHS of production rule is a MR terminal:
 - ▶ $\text{TEAM} \rightarrow \langle \text{our}, \text{our} \rangle$, $\text{UNUM} \rightarrow \langle 4, 4 \rangle$, ...
- ▶ Move to more complex rules (adjust to account for sub patterns, skip words by writing (num)):
 - ▶ $\text{COND.} \rightarrow$
 $\langle \text{TEAM}_1 \text{ player UNUM}_2 \text{ has } (1) \text{ ball, (bowner TEAM}_1 \{ \text{UNUM}_2 \}) \rangle$

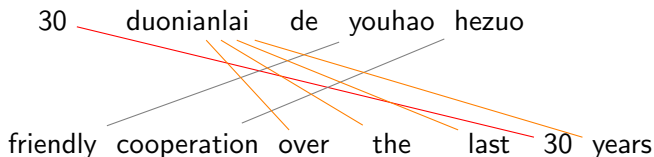
Similar methods: Hiero rule extraction

- Specialized version of methods used for other types of syntax-based decoding, e.g., hierarchical phrase-based translation (Chiang (2005))
 - Does not require syntactic rules or analyses, learns them from scratch.



Similar methods: Hiero rule extraction

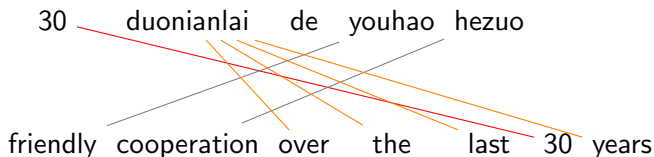
- Specialized version of methods used for other types of syntax-based decoding, e.g., hierarchical phrase-based translation (Chiang (2005))
 - Does not require syntactic rules or analyses, learns them from scratch.



$$X_1 \rightarrow \langle 30, 30 \rangle$$

Similar methods: Hiero rule extraction

- Specialized version of methods used for other types of syntax-based decoding, e.g., hierarchical phrase-based translation (Chiang (2005))
 - Does not require syntactic rules or analyses, learns them from scratch.

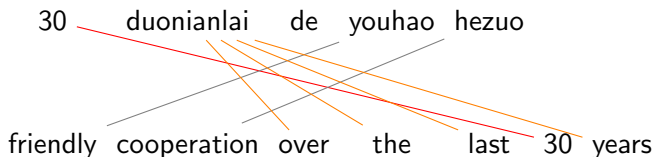


$$X_1 \rightarrow \langle 30, 30 \rangle$$

$$X_2 \rightarrow \langle \text{friendly cooperation}, \text{youhao hezuo} \rangle$$

Similar methods: Hiero rule extraction

- Specialized version of methods used for other types of syntax-based decoding, e.g., hierarchical phrase-based translation (Chiang (2005))
 - Does not require syntactic rules or analyses, learns them from scratch.



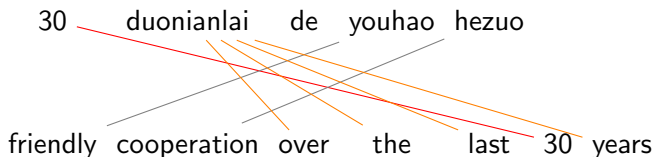
$$X_1 \rightarrow \langle 30, 30 \rangle$$

$$X_2 \rightarrow \langle \text{friendly cooperation}, \text{youhao hezuo} \rangle$$

$$X_3 \rightarrow \langle \text{over the last } X_1 \text{ years}, X_1 \text{ duonianlai} \rangle$$

Similar methods: Hiero rule extraction

- Specialized version of methods used for other types of syntax-based decoding, e.g., hierarchical phrase-based translation (Chiang (2005))
 - Does not require syntactic rules or analyses, learns them from scratch.



$$X_1 \rightarrow \langle 30, 30 \rangle$$

$$X_2 \rightarrow \langle \text{friendly cooperation}, \text{youhao hezuo} \rangle$$

$$X_3 \rightarrow \langle \text{over the last } X_1 \text{ years}, X_1 \text{ duonianlai} \rangle$$

$$X_4 \rightarrow \langle X_2 X_3, X_3 X_2 \rangle$$

Extension to logical variables

- ▶ So far, has been used on *functional representations*.
- ▶ λ -Wasp (Wong and Mooney (2007)) extends rules extraction to handle logical and lambda variables, of the type:

$$A \rightarrow \langle \alpha, \lambda x_1, \dots, \lambda x_n \beta \rangle$$

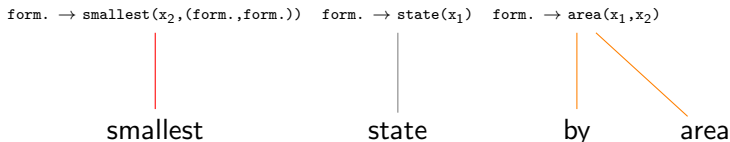
`form. \rightarrow smallest(x_2 , (form., form.))` `form. \rightarrow state(x_1)` `form. \rightarrow area(x_1, x_2)`



Extension to logical variables

- ▶ So far, has been used on *functional representations*.
- ▶ λ -Wasp (Wong and Mooney (2007)) extends rules extraction to handle logical and lambda variables, of the type:

$$A \rightarrow \langle \alpha, \lambda x_1, \dots, \lambda x_n \beta \rangle$$

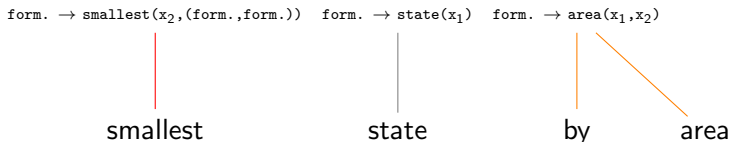


$$\text{form} \rightarrow \langle \text{state}, \lambda x_1. \text{state}(x) \rangle$$

Extension to logical variables

- ▶ So far, has been used on *functional representations*.
- ▶ λ -Wasp (Wong and Mooney (2007)) extends rules extraction to handle logical and lambda variables, of the type:

$$A \rightarrow \langle \alpha, \lambda x_1, \dots, \lambda x_n \beta \rangle$$

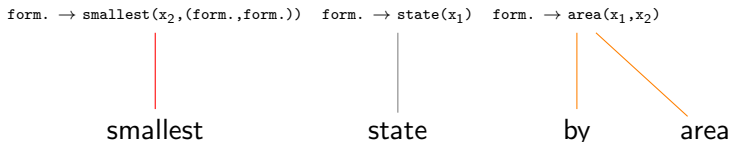


$$\begin{aligned} \text{form} &\rightarrow \langle \text{state}, \lambda x_1. \text{state}(x) \rangle \\ \text{form} &\rightarrow \langle \text{by area}, \lambda x_1. \lambda y_2. \text{area}(x, y) \rangle \end{aligned}$$

Extension to logical variables

- ▶ So far, has been used on *functional representations*.
- ▶ λ -Wasp (Wong and Mooney (2007)) extends rules extraction to handle logical and lambda variables, of the type:

$$A \rightarrow \langle \alpha, \lambda x_1, \dots, \lambda x_n \beta \rangle$$



$$\text{form} \rightarrow \langle \text{state}, \lambda x_1. \text{state}(x) \rangle$$

$$\text{form} \rightarrow \langle \text{by area}, \lambda x_1. \lambda y_2. \text{area}(x, y) \rangle$$

$$\text{form} \rightarrow \langle \text{smallest form}_1 \text{ form}_2, \lambda x_1. \text{smallest}(x_2, (\text{form}_1(x_1), \text{form}_2(x_1, x_2))) \rangle$$

Probabilistic Model

- ▶ **Lexical/rule induction:** Over-generates, leading to many derivations.
- ▶ Extend the SCFG to a *weighted* SCFG (the synchronous analogue of the PCFG), which defines a probability distribution over derivations.
- ▶ Goal is to discriminate different derivations, and find an output translation f^* where

$$f^* = m(\arg \max_{d \in D(G|e)} Pr_{\lambda}(d | e))$$

Probabilistic Model

- ▶ **Lexical/rule induction:** Over-generates, leading to many derivations.
- ▶ Extend the SCFG to a *weighted* SCFG (the synchronous analogue of the PCFG), which defines a probability distribution over derivations.
- ▶ Goal is to discriminate different derivations, and find an output translation f^* where

$$f^* = m(\arg \max_{d \in D(G|e)} Pr_{\lambda}(d | e))$$

- ▶ $D(G | e)$: The set of derivations given an english input e .
 - ▶ Computed using dynamic-programming and something close to the inside-outside algorithm (last week)
- ▶ $Pr_{\lambda}(d | e)$: Training a log-linear model on example derivations (more on this next week).

Overview and Take-aways

- ▶ Recasting the semantic parsing problem as an MT task.

Overview and Take-aways

- ▶ Recasting the semantic parsing problem as an MT task.
 - ▶ **Synchronous grammars:** modeling NL-MR transformations and decoding by parsing.
 - ▶ **Word-Alignment Models:** Basics, extracting semantic grammar transformation rules
 - ▶ **Decoding and ranking models:** Skipped over important details, **more about this next week**

Overview and Take-aways

- ▶ Recasting the semantic parsing problem as an MT task.
 - ▶ **Synchronous grammars:** modeling NL-MR transformations and decoding by parsing.
 - ▶ **Word-Alignment Models:** Basics, extracting semantic grammar transformation rules
 - ▶ **Decoding and ranking models:** Skipped over important details, **more about this next week**
- ▶ Further directions
 - ▶ Different tree-based translation models (Ehsen), more powerful translation models (Mariia)
 - ▶ Different rule extraction techniques: Li et al. (2013)

Roadmap

- ▶ **Lecture 3 (today):** rule extraction, decoding (MT perspective)
- ▶ **Lecture 4:** Structure prediction and classification (**missing today**).

References I

- Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.
- Chiang, D. (2005). A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270. Association for Computational Linguistics.
- Chiang, D. and Knight, K. (2006). An introduction to synchronous grammars. *Tutorial available at <http://www.isi.edu/chiang/papers/synchtut.pdf>*.
- Kate, R. J., Wong, Y. W., and Mooney, R. J. (2005). Learning to transform natural to formal languages. In *Proceedings of AAAI-2005*.
- Koehn, P. (2009). *Statistical machine translation*. Cambridge University Press.
- Li, P., Liu, Y., and Sun, M. (2013). An extended ghkm algorithm for inducing lambda-scfg. In *AAAI*.
<http://www.aaai.org/ocs/index.php/AAAI/AAAI13/paper/view/6189>.
- Wong, Y. W. and Mooney, R. J. (2006). Learning for semantic parsing with statistical machine translation. In *Proceedings of HLT-NAACL-2006*, pages 439–446.
<http://dl.acm.org/citation.cfm?id=1220891>.
- Wong, Y. W. and Mooney, R. J. (2007). Learning synchronous grammars for semantic parsing with lambda calculus. In *Proceedings of ACL-2007*, Prague, Czech Republic. <http://anthology.aclweb.org/P/P07/P07-1121.pdf>.

References II

- Woods, W. A. (1973). Progress in natural language understanding: an application to lunar geology. In *Proceedings of the June 4-8, 1973, National Computer Conference and Exposition*, pages 441–450.
- Zelle, J. M. and Mooney, R. J. (1996). Learning to parse database queries using inductive logic programming. In *Proceedings of AAAI-1996*, pages 1050–1055.