Natural Language Processing

Machine Translation III

Dan Klein – UC Berkeley

Phrase-Based MT

Phrase-Based Translation Overview

Input: lo haré | rápidamente | .

Translations: I'll do it | quickly | .

quickly | I'll do it | .

The decoder...
tries different segmentations,
translates phrase by phrase,
and considers reorderings.

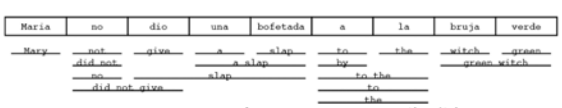
Objective: $\arg\max_{\mathbf{e}} [P(\mathbf{f}|\mathbf{e}) \cdot P(\mathbf{e})]$

$$\arg\max_{\mathbf{e}} \left[\prod_{\langle \bar{e}, \bar{f} \rangle} P(\bar{f}|\bar{e}) \cdot \prod_{i=1}^{|\mathbf{e}|} P(e_i|e_{i-1}, e_{i-2}) \right]$$

Phrase-Based Decoding

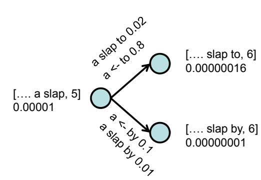
Decoder design is important: [Koehn et al. 03]

Phrase-Based Decoding



Monotonic Word Translation

- Cost is LM * TM
- It's an HMM?
  - $P(e|e_{-1},e_{-2})$
  - $P(f|e)$
- State includes
  - Exposed English
  - Position in foreign
- Dynamic program loop?



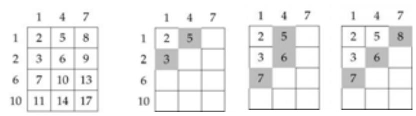

```
for (fPosition in 1…|f|)
  for (eContext in allEContexts)
    for (eOption in translations[fPosition])
      score = scores[fPosition-1][eContext] * LM(eContext+eOption) * TM(eOption, fWord[fPosition])
      scores[fPosition][eContext[2]+eOption] =max score
```

Beam Decoding

- For real MT models, this kind of dynamic program is a disaster (why?)
- Standard solution is beam search: for each position, keep track of only the best k hypotheses

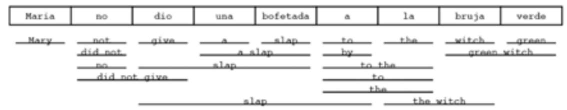
```
for (fPosition in 1…|f|)
  for (eContext in bestEContexts[fPosition])
    for (eOption in translations[fPosition])
      score = scores[fPosition-1][eContext] * LM(eContext+eOption) * TM(eOption, fWord[fPosition])
      bestEContexts.maybeAdd(eContext[2]+eOption, score)
```

- Still pretty slow… why?
- Useful trick: cube pruning (Chiang 2005)



Example from David Chiang

Phrase Translation

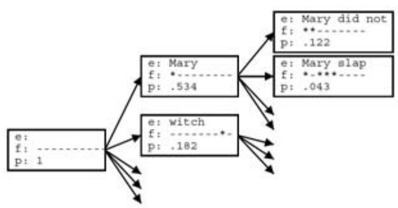


- If monotonic, almost an HMM; technically a semi-HMM

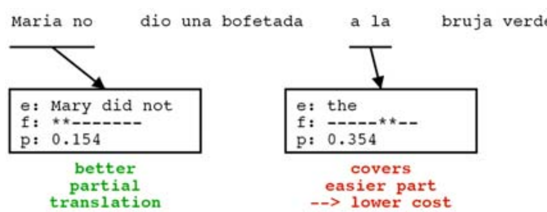
```
for (fPosition in 1…|f|)
  for (lastPosition < fPosition)
    for (eContext in eContexts)
      for (eOption in translations[fPosition])
        … combine hypothesis for (lastPosition ending in eContext) with eOption
```

- If distortion… now what?

Non-Monotonic Phrasal MT



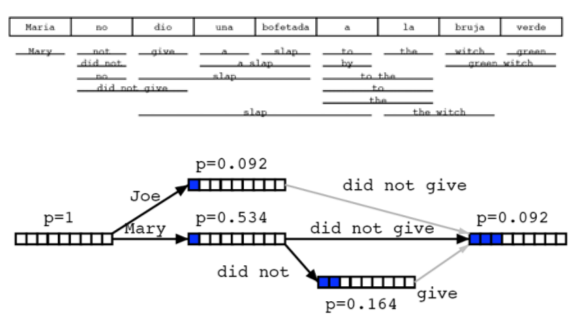
Pruning: Beams + Forward Costs



- Problem: easy partial analyses are cheaper
  - Solution 1: use beams per foreign subset
  - Solution 2: estimate forward costs (A*-like)

The Pharaoh Decoder

Hypotheis Lattices



Parameter Tuning

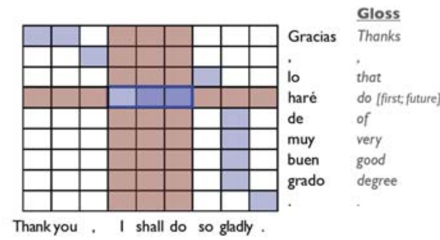
Counting Phrase Pairs

Input:

Gracias , lo haré de muy buen grado .
Thank you , I shall do so gladly .

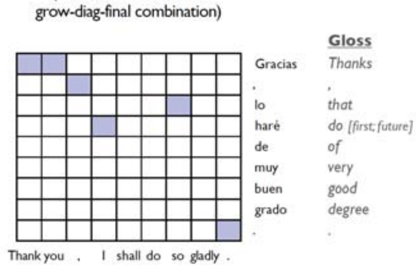
First, we learn word alignments,

then we infer aligned phrases.



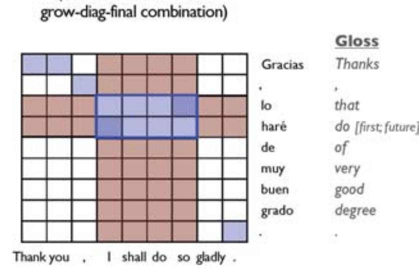
What Happens in Practice

A real word alignment
(GIZA++ Model 4 with
grow-diag-final combination)



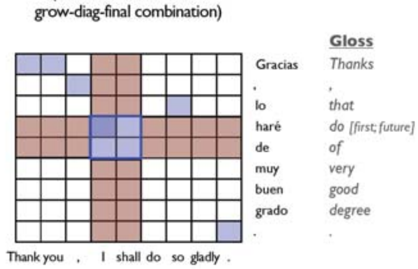
What Happens in Practice

A real word alignment
(GIZA++ Model 4 with
grow-diag-final combination)



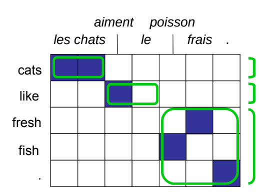
What Happens in Practice

A real word alignment
(GIZA++ Model 4 with
grow-diag-final combination)



Phrase Scoring

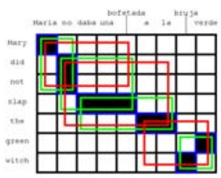
$$\phi_{new}(\bar{e}_j|\bar{f}_i) = \frac{c(\bar{f}_i, \bar{e}_j)}{c(\bar{f}_i)}$$

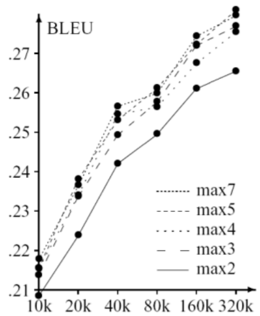


- Learning weights has been tried, several times:
  - [Marcu and Wong, 02]
  - [DeNero et al, 06]
  - … and others
- Seems not to work well, for a variety of partially understood reasons
- Main issue: big chunks get all the weight, obvious priors don't help
  - Though, [DeNero et al 08]

Phrase Size

- Phrases do help
  - But they don't need to be long
  - Why should this be?



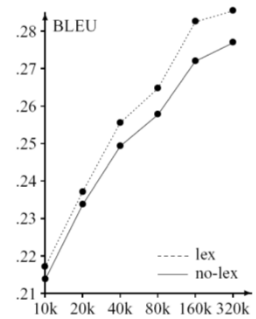


Lexical Weighting

$$\phi(\bar{f}_i|\bar{e}_i) = \frac{count(\bar{f}_i, \bar{e}_i)}{count(\bar{e}_i)} p_w(\bar{f}_i|\bar{e}_i)$$

	f1	f2	f3
NULL	--	--	##
e1	##	--	--
e2	--	##	--
e3	--	##	--

$$\begin{aligned} p_w(\bar{f}|\bar{e}, a) &= p_w(f_1 f_2 f_3|e_1 e_2 e_3, a) \\ &= w(f_1|e_1) \\ &\quad \times \frac{1}{2}(w(f_2|e_2) + w(f_2|e_3)) \\ &\quad \times w(f_3|\text{NULL}) \end{aligned}$$



Tuning for MT

- Features encapsulate lots of information
  - Basic MT systems have around 6 features
  - P(e|f), P(f|e), lexical weighting, language model
- How to tune feature weights?
- Idea 1: Use your favorite classifier

Why Tuning is Hard

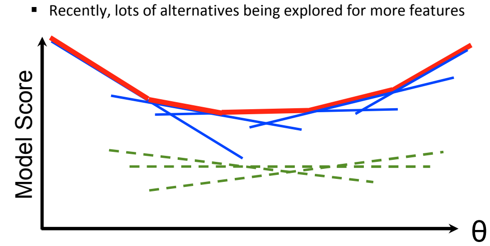
- Problem 1: There are latent variables
  - Alignments and segementations
  - Possibility: forced decoding (but it can go badly)

Why Tuning is Hard

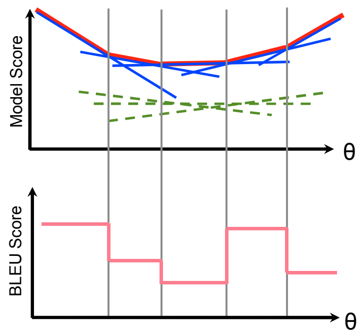
- Problem 3: Computational constraints
  - Discriminative training involves repeated decoding
  - Very slow! So people tune on sets much smaller than those used to build phrase tables

Minimum Error Rate Training

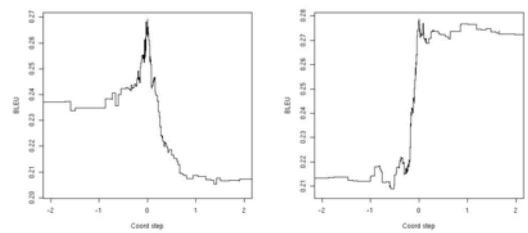
- Standard method: minimize BLEU directly (Och 03)
  - MERT is a discontinuous objective
  - Only works for max ~10 features, but works very well then
  - Here: k-best lists, but forest methods exist (Machery et al 08)
  - Recently, lots of alternatives being explored for more features



MERT



MERT



Translating with Tree Transducers

Input **Output**

lo haré de muy buen grado .

Grammar

Translating with Tree Transducers

Input **Output**

lo haré de muy buen grado .

Grammar

ADV → ⟨ de muy buen grado ; gladly ⟩

Syntactic Models

Translating with Tree Transducers

Input **Output**

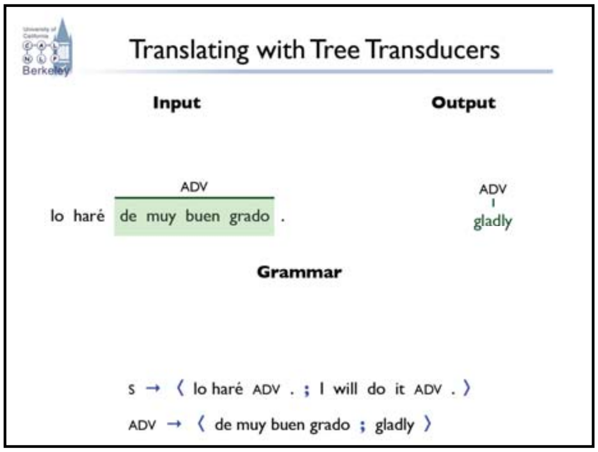
ADV

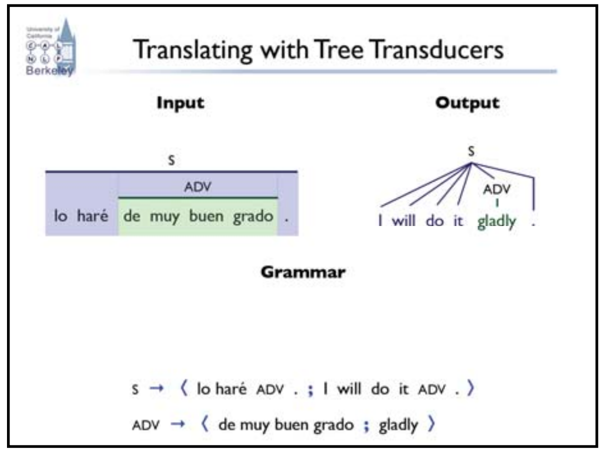
lo haré de muy buen grado .

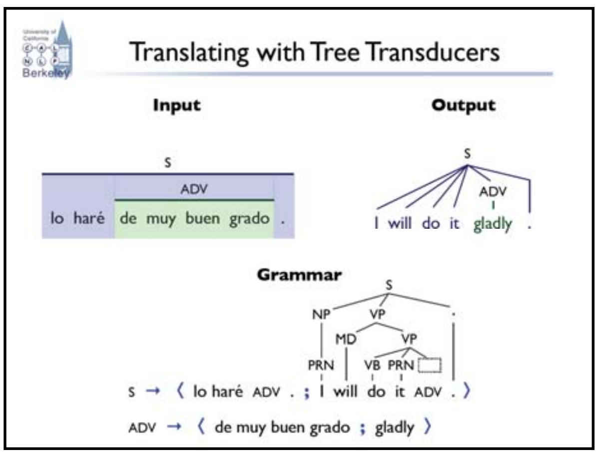
ADV
gladly

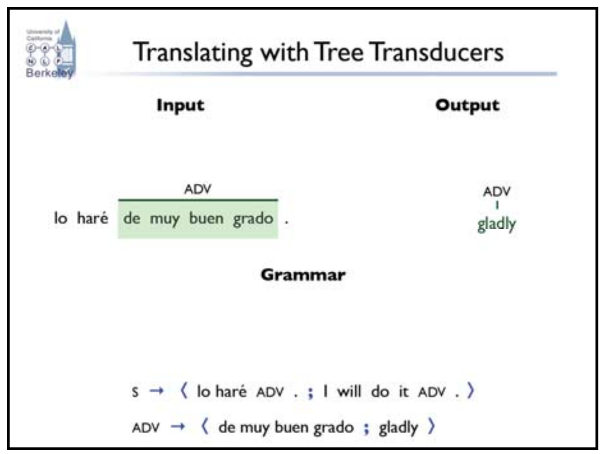
Grammar

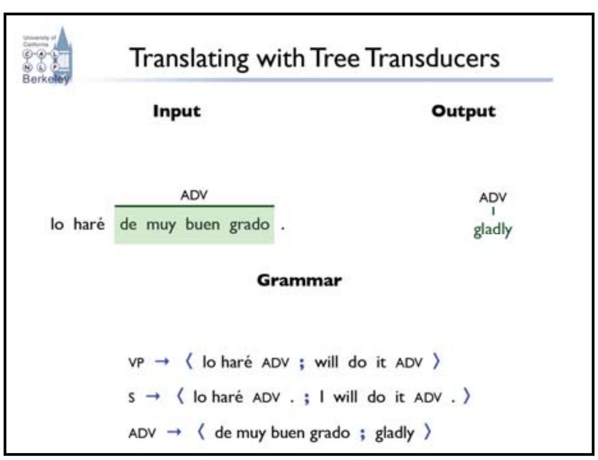
ADV → ⟨ de muy buen grado ; gladly ⟩

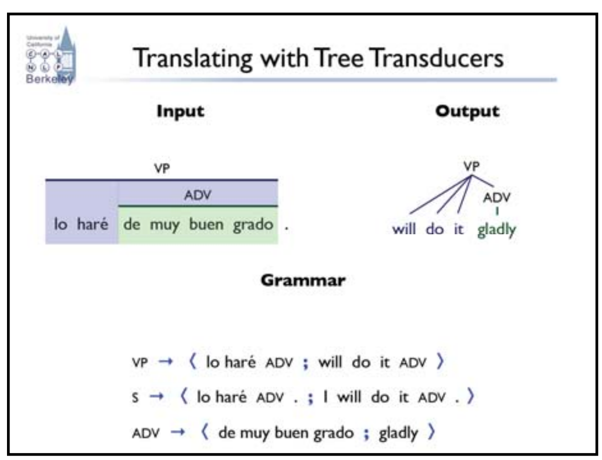

Translating with Tree Transducers
Input
Output
ADV
lo haré de muy buen grado .
ADV
gladly
Grammar
S → ⟨ lo haré ADV . ; I will do it ADV . ⟩
ADV → ⟨ de muy buen grado ; gladly ⟩

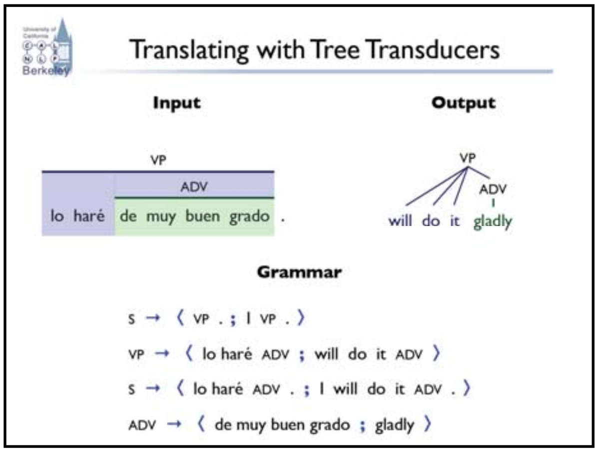

Translating with Tree Transducers
Input
Output
S
ADV
lo haré de muy buen grado .
S
ADV
I will do it gladly .
Grammar
S → ⟨ lo haré ADV . ; I will do it ADV . ⟩
ADV → ⟨ de muy buen grado ; gladly ⟩

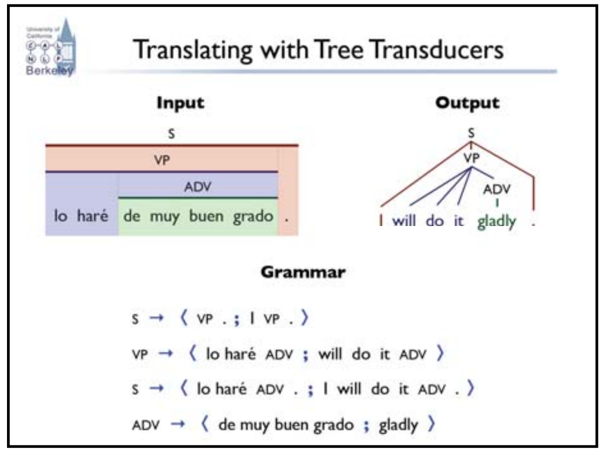

Translating with Tree Transducers
Input
Output
S
ADV
lo haré de muy buen grado .
S
ADV
I will do it gladly .
Grammar
S
NP
VP
MD
VP
PRN
VB
PRN
S → ⟨ lo haré ADV . ; I will do it ADV . ⟩
ADV → ⟨ de muy buen grado ; gladly ⟩

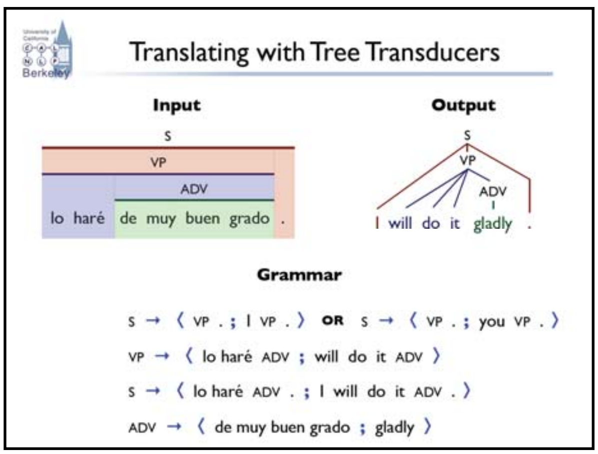

Translating with Tree Transducers
Input
Output
ADV
lo haré de muy buen grado .
ADV
gladly
Grammar
S → ⟨ lo haré ADV . ; I will do it ADV . ⟩
ADV → ⟨ de muy buen grado ; gladly ⟩

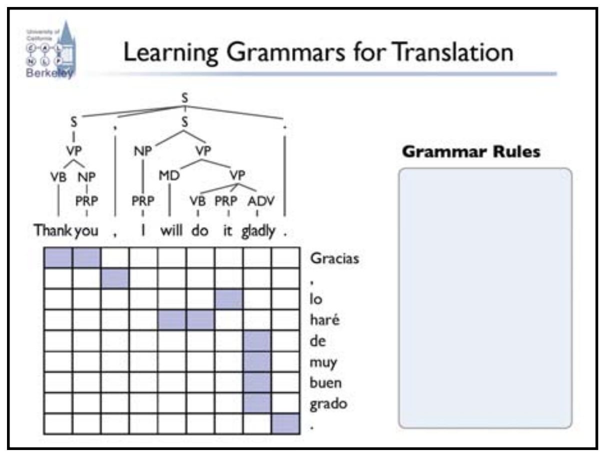

Translating with Tree Transducers
Input
Output
ADV
lo haré de muy buen grado .
ADV
gladly
Grammar
VP → ⟨ lo haré ADV ; will do it ADV ⟩
S → ⟨ lo haré ADV . ; I will do it ADV . ⟩
ADV → ⟨ de muy buen grado ; gladly ⟩

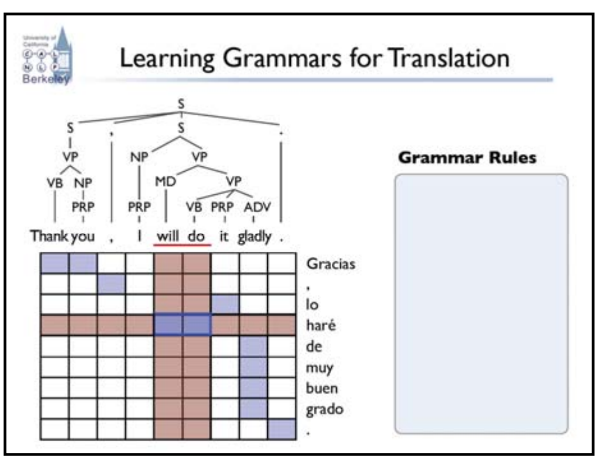

Translating with Tree Transducers
Input
Output
VP
ADV
lo haré de muy buen grado .
VP
ADV
will do it gladly
Grammar
VP → ⟨ lo haré ADV ; will do it ADV ⟩
S → ⟨ lo haré ADV . ; I will do it ADV . ⟩
ADV → ⟨ de muy buen grado ; gladly ⟩

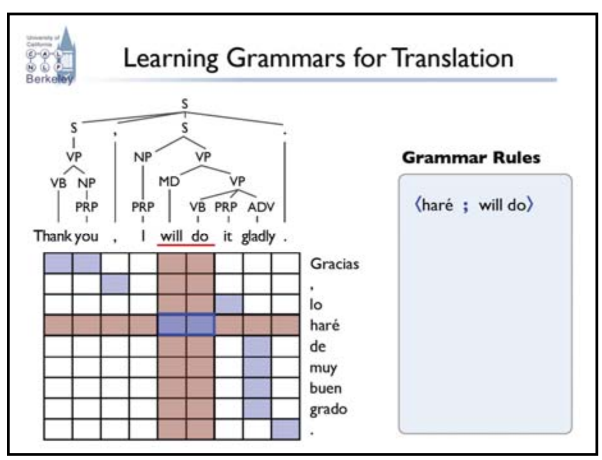

Translating with Tree Transducers
Input
Output
VP
ADV
lo haré de muy buen grado .
will do it gladly
Grammar
S → 〈 VP . ; I VP . 〉
VP → 〈 lo haré ADV ; will do it ADV 〉
S → 〈 lo haré ADV . ; I will do it ADV . 〉
ADV → 〈 de muy buen grado ; gladly 〉

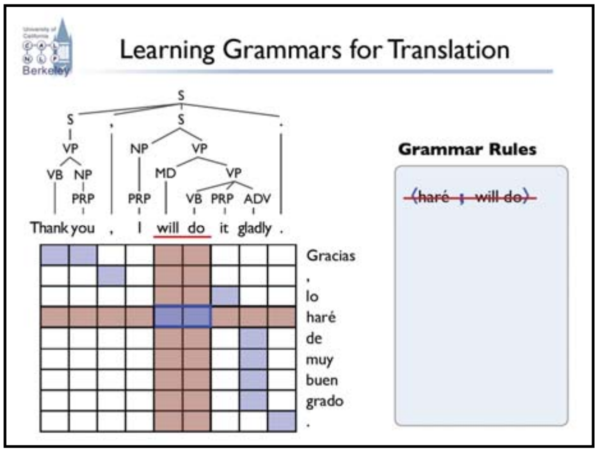

Translating with Tree Transducers
Input
Output
S
VP
ADV
lo haré de muy buen grado .
I will do it gladly .
Grammar
S → 〈 VP . ; I VP . 〉
VP → 〈 lo haré ADV ; will do it ADV 〉
S → 〈 lo haré ADV . ; I will do it ADV . 〉
ADV → 〈 de muy buen grado ; gladly 〉

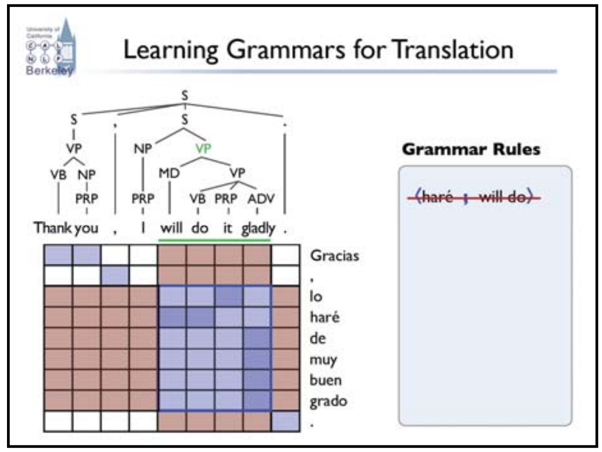

Translating with Tree Transducers
Input
Output
S
VP
ADV
lo haré de muy buen grado .
I will do it gladly .
Grammar
S → 〈 VP . ; I VP . 〉 OR S → 〈 VP . ; you VP . 〉
VP → 〈 lo haré ADV ; will do it ADV 〉
S → 〈 lo haré ADV . ; I will do it ADV . 〉
ADV → 〈 de muy buen grado ; gladly 〉

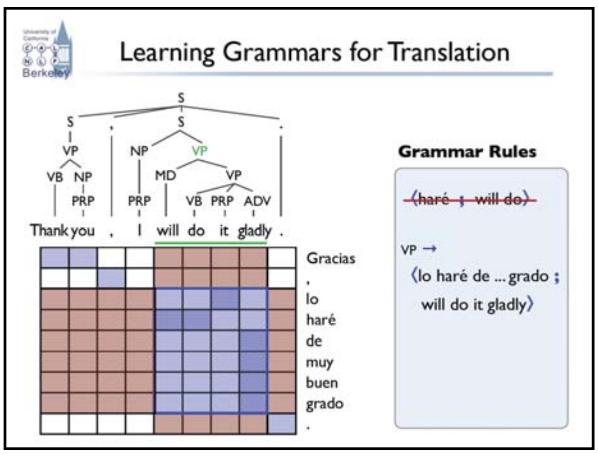

Learning Grammars for Translation
S
S
VP
VB NP
PRP
NP
MD
VP
VB PRP ADV
Thank you , I will do it gladly .
Gracias
,
lo
haré
de
muy
buen
grado
.
Grammar Rules

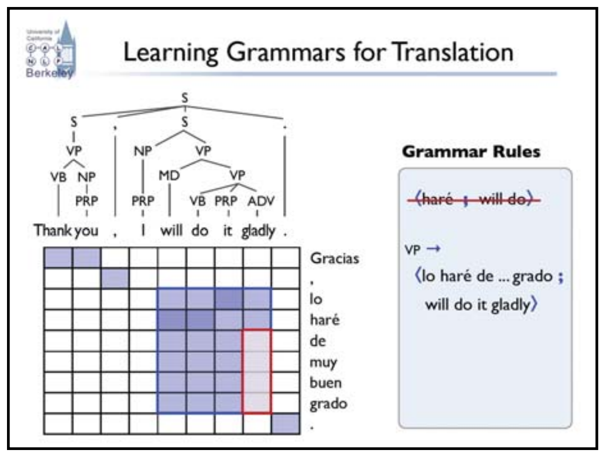

Learning Grammars for Translation
Thank you , I will do it gladly .
Gracias
,
lo
haré
de
muy
buen
grado
.
Grammar Rules

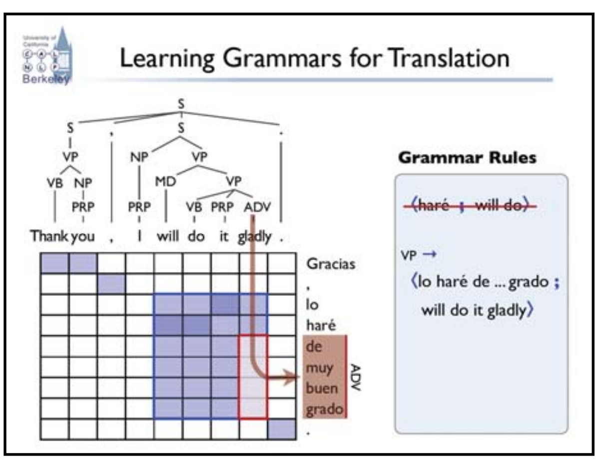

Learning Grammars for Translation
Thank you , I will do it gladly .
Gracias
,
lo
haré
de
muy
buen
grado
.
Grammar Rules
〈haré ; will do〉

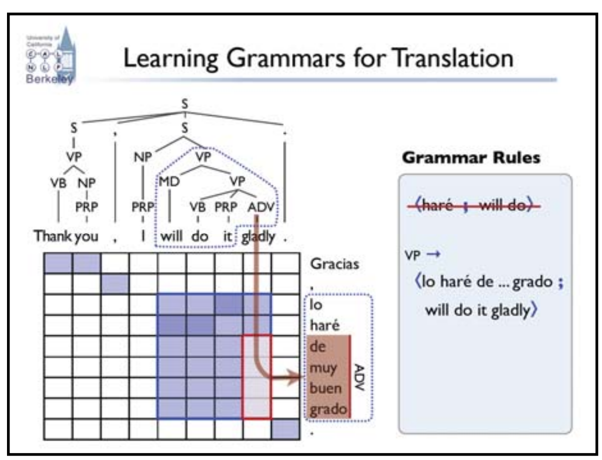

Learning Grammars for Translation
S
S , S .
VP NP VP
VB NP MD VP
PRP PRP VB PRP ADV
Thank you , I will do it gladly .
Gracias
,
lo
haré
de
muy
buen
grado
.
Grammar Rules
~~⟨haré ; will do⟩~~


Learning Grammars for Translation
S
S , S .
VP NP VP
VB NP MD VP
PRP PRP VB PRP ADV
Thank you , I will do it gladly .
Gracias
,
lo
haré
de
muy
buen
grado
.
Grammar Rules
~~⟨haré ; will do⟩~~

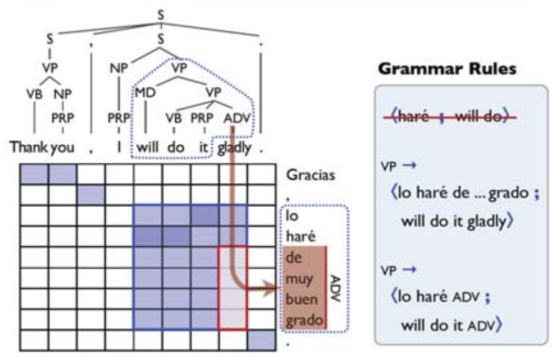

Learning Grammars for Translation
S
S , S .
VP NP VP
VB NP MD VP
PRP PRP VB PRP ADV
Thank you , I will do it gladly .
Gracias
,
lo
haré
de
muy
buen
grado
.
Grammar Rules
~~⟨haré ; will do⟩~~
VP →
⟨lo haré de ... grado ;
will do it gladly⟩


Learning Grammars for Translation
S
S , S .
VP NP VP
VB NP MD VP
PRP PRP VB PRP ADV
Thank you , I will do it gladly .
Gracias
,
lo
haré
de
muy
buen
grado
.
Grammar Rules
~~⟨haré ; will do⟩~~
VP →
⟨lo haré de ... grado ;
will do it gladly⟩


Learning Grammars for Translation
S
S , S .
VP NP VP
VB NP MD VP
PRP PRP VB PRP ADV
Thank you , I will do it gladly .
Gracias
,
lo
haré
de
muy
buen
grado
.
ADV
Grammar Rules
~~⟨haré ; will do⟩~~
VP →
⟨lo haré de ... grado ;
will do it gladly⟩


Learning Grammars for Translation
S
S , S .
VP NP VP
VB NP MD VP
PRP PRP VB PRP ADV
Thank you , I will do it gladly .
Gracias
,
lo
haré
de
muy
buen
grado
.
ADV
Grammar Rules
~~⟨haré ; will do⟩~~
VP →
⟨lo haré de ... grado ;
will do it gladly⟩

Learning Grammars for Translation



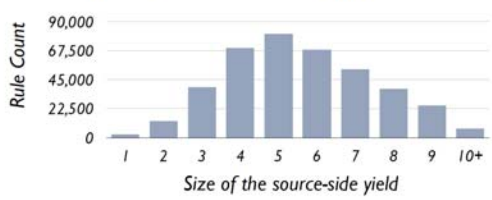
The Size of Tree Transducer Grammars

Extracted a transducer grammar from a 220 million word bitext → Relativized the grammar to each test sentence → Kept all rules with at most 6 non-terminals

The Size of Tree Transducer Grammars

Extracted a transducer grammar from a 220 million word bitext → Relativized the grammar to each test sentence → Kept all rules with at most 6 non-terminals

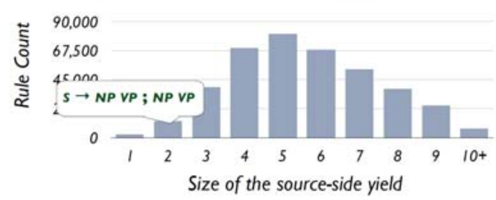
Rules matching an example 40-word sentence



The Size of Tree Transducer Grammars

Extracted a transducer grammar from a 220 million word bitext → Relativized the grammar to each test sentence → Kept all rules with at most 6 non-terminals

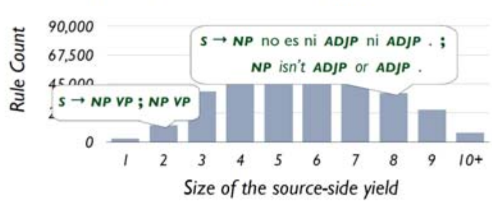
Rules matching an example 40-word sentence



The Size of Tree Transducer Grammars

Extracted a transducer grammar from a 220 million word bitext → Relativized the grammar to each test sentence → Kept all rules with at most 6 non-terminals

Rules matching an example 40-word sentence



Syntactic Decoding

Tree Transducer Grammars

S
NN NNP
No se olvide de subir un canto rodado en Colorado

Synchronous Grammar

NNP → Colorado ; *Colorado*

NN → canto rodado ; *boulder*

S → No se olvide de subir un ***NN*** en ***NNP*** ; *Don't forget to climb a* ***NN*** *in* ***NNP***

Output

S
NN NNP
Don't forget to climb a boulder in Colorado

CKY-style Bottom-up Parsing

For each span length:

CKY-style Bottom-up Parsing

For each span length: For each span [i,j]:

CKY-style Bottom-up Parsing

For each span length: For each span [i,j]: Apply all grammar rules to [i,j]

CKY-style Bottom-up Parsing

For each span length: For each span [i,j]: Apply all grammar rules to [i,j]

Binary rule: X → Y Z

CKY-style Bottom-up Parsing

For each span length: For each span [i,j]: Apply all grammar rules to [i,j]

Binary rule: X → Y Z

Split points: i < k < j

Operations: O(j - i)

Time scales with: Grammar constant

CKY-style Bottom-up Parsing

For each span length: For each span [i,j]: Apply all grammar rules to [i,j]

$_i$ No se olvide de subir un canto rodado en Colorado $_j$

CKY-style Bottom-up Parsing

For each span length: For each span [i,j]: Apply all grammar rules to [i,j]

S → No se ***VB*** de subir un ***NN*** en ***NNP***

$_i$ No se olvide de subir un canto rodado en Colorado $_j$

CKY-style Bottom-up Parsing

For each span length: For each span [i,j]: Apply all grammar rules to [i,j]

S → No se ***VB*** de subir un ***NN*** en ***NNP***

$_i$ No se olvide de subir un canto rodado en Colorado $_j$

CKY-style Bottom-up Parsing

For each span length: For each span [i,j]: Apply all grammar rules to [i,j]

S → No se ***VB*** de subir un ***NN*** en ***NNP***

$_i$ No se olvide de subir un canto rodado en Colorado $_j$

CKY-style Bottom-up Parsing

For each span length: For each span [i,j]: Apply all grammar rules to [i,j]

S → No se ***VB*** de subir un ***NN*** en ***NNP***

$_i$ No se olvide de subir un canto rodado en Colorado $_j$

CKY-style Bottom-up Parsing

For each span length: For each span [i,j]: Apply all grammar rules to [i,j]

S → No se ***VB*** de subir un ***NN*** en ***NNP***

$_i$ No se olvide de subir un canto rodado en Colorado $_j$

CKY-style Bottom-up Parsing

For each span length: For each span [i,j]: Apply all grammar rules to [i,j]

S → No se ***VB*** de subir un ***NN*** en ***NNP***

$_i$ No se olvide de subir un canto rodado en Colorado $_j$

CKY-style Bottom-up Parsing

For each span length: For each span [i,j]: Apply all grammar rules to [i,j]

S → No se ***VB*** de subir un ***NN*** en ***NNP***

$_i$ No se olvide de subir un canto rodado en Colorado $_j$

Many untransformed lexical rules can be applied in linear time

CKY-style Bottom-up Parsing

For each span length: For each span [i,j]: Apply all grammar rules to [i,j]

S → No se ***VP*** ***NP*** ***PP***

$_i$ No se olvide de subir un canto rodado en Colorado $_j$

CKY-style Bottom-up Parsing

For each span length: For each span [i,j]: Apply all grammar rules to [i,j]

S → No se ***VP*** ***NP*** ***PP***

$_i$ No se olvide de subir un canto rodado en Colorado $_j$

CKY-style Bottom-up Parsing

For each span length: For each span [i,j]: Apply all grammar rules to [i,j]

S → No se ***VP*** ***NP*** ***PP***

$_i$ No se olvide de subir $_k$ un canto rodado $_{k'}$ en Colorado $_j$

CKY-style Bottom-up Parsing

For each span length: For each span [i,j]: Apply all grammar rules to [i,j]

S → No se ***VP*** ***NP*** ***PP***

$_i$ No se olvide de subir $_k$ un canto rodado $_{k'}$ en Colorado $_j$

Problem: *Applying adjacent non-terminals is slow*

Eliminating Non-terminal Sequences

Lexical Normal Form (LNF)

(a) lexical rules have at most one adjacent non-terminal
(b) all unlexicalized rules are binary.

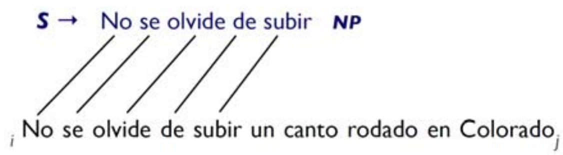
Original rule: ***S*** → No se ***VB VB*** un ***NN PP***

Transformed rules: ***S*** → No se ***VB~VB*** un ***NN~PP***
VB~VB → ***VB VB***
NN~PP → ***NN PP***

Parsing stages:
- Lexical rules are applied by matching
- Unlexicalized rules are applied by iterating over split points

Speeding up Lexical Rule Application

Problem: *Lexical rules can apply to many spans*



Speeding up Lexical Rule Application

Problem: *Lexical rules can apply to many spans*

Speeding up Lexical Rule Application

Problem: *Lexical rules can apply to many spans*

Speeding up Lexical Rule Application

Problem: *Lexical rules can apply to many spans*

Flexible Syntax

Soft Syntactic MT: From Chiang 2010

reference: An official from Japan 's science and technology ministry said , " We are highly encouraged by Abraham 's comment .

Hiero: Officials of the Japanese ministry of education and science , " said Abraham speeches , we are deeply encouraged by .

string-to-tree: Japan 's ministry of education , culture , sports , science and technology , " Abraham 's statement , which is most encouraging , " the official said .

Previous work

string-to-string	ITG (Wu 1997)	Hiero (Chiang 2005)
string-to-tree	Yamada & Knight 2001	Galley et al 2004 / 2006
tree-to-string		Huang et al 2006 Y Liu et al 2006
tree-to-tree	DOT (Poutsma 2000) Eisner 2003	Stat-XFER (Lavie et al 2008) M Zhang et al. 2008 Y Liu et al., 2009

Hiero Rules

$$S \to \langle S_{\boxed{1}} X_{\boxed{2}}, S_{\boxed{1}} X_{\boxed{2}} \rangle$$
$$S \to \langle X_{\boxed{1}}, X_{\boxed{1}} \rangle$$
$$X \to \langle \text{yu } X_{\boxed{1}} \text{ you } X_{\boxed{2}}, \text{have } X_{\boxed{2}} \text{ with } X_{\boxed{1}} \rangle$$
$$X \to \langle X_{\boxed{1}} \text{ de } X_{\boxed{2}}, \text{the } X_{\boxed{2}} \text{ that } X_{\boxed{1}} \rangle$$
$$X \to \langle X_{\boxed{1}} \text{ zhiyi}, \text{one of } X_{\boxed{1}} \rangle$$
$$X \to \langle \text{Aozhou}, \text{Australia} \rangle$$
$$X \to \langle \text{shi}, \text{is} \rangle$$
$$X \to \langle \text{shaoshu guojia}, \text{few countries} \rangle$$
$$X \to \langle \text{bangjiao}, \text{diplomatic relations} \rangle$$
$$X \to \langle \text{Bei Han}, \text{North Korea} \rangle$$

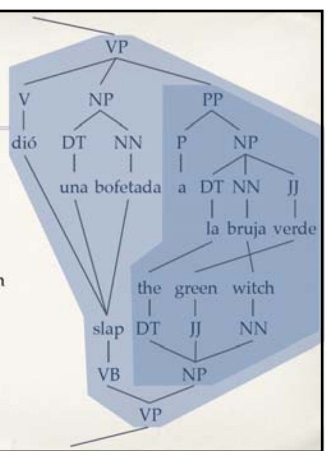
From [Chiang et al, 2005]

STSG extraction

1. Phrases
   - respect word alignments
   - are syntactic constituents on *both* sides
2. Phrase pairs form rules
3. Subtract phrases to form rules

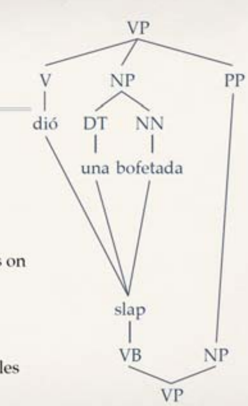
STSG extraction

1. Phrases
   - respect word alignments
   - are syntactic constituents on *both* sides
2. Phrase pairs form rules
3. Subtract phrases to form rules



STSG extraction

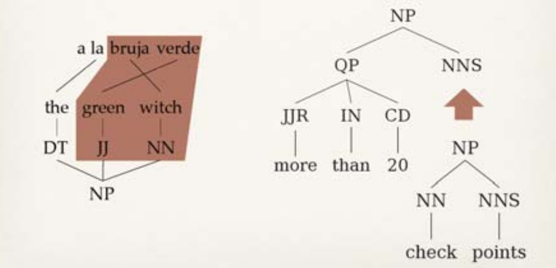
1. Phrases
   - respect word alignments
   - are syntactic constituents on *both* sides
2. Phrase pairs form rules
3. Subtract phrases to form rules



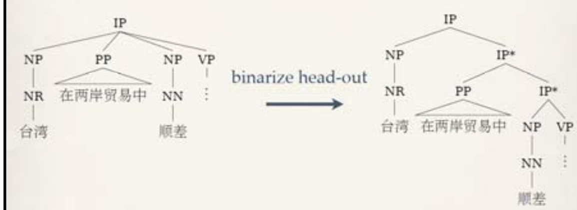
Why is tree-to-tree hard?

too few rules

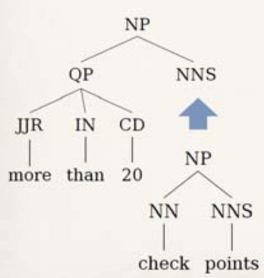
too few derivations



Extracting more rules

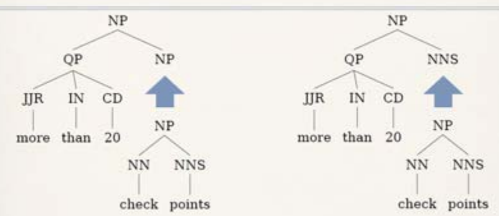


Allow more derivations



- STSG: allow only matching substitutions
- Hiero-like: allow any substitutions
- Let the model learn to choose:
  - matching substitutions
  - mismatching substitutions
  - monotone phrase-based

Allow more derivations



fire *subst*:NP→NP
fire *subst:match*

fire *subst*:NNS→NP
fire *subst:unmatch*

Allow more derivations

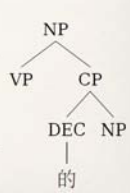
Hiero-like decoding

$$\frac{[X,i,j] \qquad [X,j+1,k]}{[X,i,k]}$$

X → X 的 X

STSG decoding

$$\frac{[VP,i,j] \qquad [NP,j+1,k]}{[NP,i,k]}$$



fuzzy STSG decoding

$$\frac{[A,i,j] \qquad [B,j+1,k]}{[NP,i,k]}$$

Results

	Chinese-English			Arabic-English		
extraction	rules	feats	BLEU	rules	feats	BLEU
Hiero	440M	1k	23.7	790M	1k	48.9
fuzzy STSG	50M	5k	23.9	38M	5k	47.5
fuzzy STSG +binarize	64M	5k	24.3	40M	6k	48.1
fuzzy STSG +SAMT	440M	160k	24.3	790M	130k	49.7

Example tree-to-tree translation

日本 文部 科学 省 官员 表示 , " 亚 伯拉罕 的 发言 , 令 我们 深感 鼓舞
Japan MEXT official said , " Abraham 's comment make us deeply-feel courage

reference: An official from Japan 's science and technology ministry said , " We are highly encouraged by Abraham 's comment .

Hiero: Officials of the Japanese ministry of education and science , " said Abraham speeches , we are deeply encouraged by .

string-to-tree: Japan 's ministry of education , culture , sports , science and technology , " Abraham 's statement , which is most encouraging , " the official said .

Fuzzy STSG, binarize: Officials of the Japanese ministry of education , culture , sports , science and technology , said , " we are very encouraged by the speeches of Abraham .