Natural Language Processing

Machine Translation III

Dan Klein – UC Berkeley
Phrase-Based MT
Phrase-Based Translation Overview

**Input:** \( \text{lo haré rápidamente} \).

**Translations:** I’ll do it quickly.

quickly I’ll do it.

**Objective:**
\[
\text{arg max}_e \left[ P(f|e) \cdot P(e) \right]
\]

\[
\text{arg max}_e \left[ \prod_{\langle e, f \rangle} P(f|e) \cdot \prod_{i=1}^{|e|} P(e_i|e_{i-1}, e_{i-2}) \right]
\]

The decoder... tries different segmentations, translates phrase by phrase, and considers reorderings.
Phrase-Based Decoding

| the 7 people including by some and the russian the astronauts |
|---|---|---|---|
| it 7 people included by france and the russian international astronautical of rapporteur |
| this 7 out including the from the french and the russian the fifth |
| these 7 among including from the french and of the russian of space members |
| that 7 persons including from the of france and to russian of the aerospace members |
| 7 include from the of france and russian astronauts the |
| 7 numbers include from france and russian of astronauts who |
| 7 populations include those from france and russian astronauts |
| 7 deportees included come from france and russian in astronautical personnel |
| 7 philtrum including those from france and russian a space member |
| including representatives from france and the russian astronaut |
| include came from france and russia by cosmonauts |
| include representatives from french and russia cosmonauts |
| include came from france and russia’s cosmonauts |
| includes coming from french and russia’s cosmonaut |
| french and russian russia’s astronaut |
| french and russia astronauts |
| and russian’s special rapporteur |
| , and russia rapporteur |
| , and russia rapporteur |
| , and russia |
| or russia’s |

Decoder design is important: [Koehn et al. 03]
Phrase-Based Decoding

<table>
<thead>
<tr>
<th>Maria</th>
<th>no</th>
<th>dio</th>
<th>una</th>
<th>bofetada</th>
<th>a</th>
<th>la</th>
<th>bruja</th>
<th>verde</th>
</tr>
</thead>
</table>

Mary

did not

did not give

e a slap

to the

slap

to the witch
Monotonic Word Translation

<table>
<thead>
<tr>
<th>Maria</th>
<th>no</th>
<th>dio</th>
<th>una</th>
<th>bofetada</th>
<th>a</th>
<th>la</th>
<th>bruja</th>
<th>verde</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>not</td>
<td>give</td>
<td>a</td>
<td>slap</td>
<td>to</td>
<td>the</td>
<td>witch</td>
<td>green</td>
</tr>
</tbody>
</table>

- Cost is $LM \times TM$
- It’s an HMM?
  - $P(e|e_1,e_2)$
  - $P(f|e)$
- State includes
  - Exposed English
  - Position in foreign
- Dynamic program loop?

```plaintext
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

```plaintext
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

<table>
<thead>
<tr>
<th>Maria</th>
<th>no</th>
<th>dio</th>
<th>una</th>
<th>bofetada</th>
<th>a</th>
<th>la</th>
<th>bruja</th>
<th>verde</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>not</td>
<td>give</td>
<td>a</td>
<td>slap</td>
<td>to</td>
<td>the</td>
<td>witch</td>
<td>green</td>
</tr>
<tr>
<td>did not</td>
<td>slap</td>
<td>by</td>
<td>green witch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>no</td>
<td>slap</td>
<td>to the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>did not give</td>
<td></td>
<td>to</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>slap</td>
<td>the witch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- 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

<table>
<thead>
<tr>
<th>Maria</th>
<th>no</th>
<th>dio</th>
<th>una</th>
<th>bofetada</th>
<th>a</th>
<th>la</th>
<th>bruja</th>
<th>verde</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>not</td>
<td>give</td>
<td>a</td>
<td>slap</td>
<td>to</td>
<td>the</td>
<td>witch</td>
<td>green</td>
</tr>
<tr>
<td>did not</td>
<td></td>
<td></td>
<td>a</td>
<td>slap</td>
<td>by</td>
<td>green witch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>no</td>
<td></td>
<td></td>
<td>slap</td>
<td>to the</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>did not give</td>
<td></td>
<td></td>
<td></td>
<td>to</td>
<td>the</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>slap</td>
<td></td>
<td>the</td>
<td>witch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Maria</th>
<th>no</th>
<th>dio</th>
<th>una</th>
<th>bofetada</th>
<th>a</th>
<th>la</th>
<th>bruja</th>
<th>verde</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>did not</td>
<td>slap</td>
<td>the</td>
<td>green</td>
<td>witch</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Hypothesis Lattices

<table>
<thead>
<tr>
<th>Maria</th>
<th>no</th>
<th>dio</th>
<th>una</th>
<th>bofetada</th>
<th>a</th>
<th>la</th>
<th>bruja</th>
<th>verde</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>not</td>
<td>give</td>
<td>a</td>
<td>slap</td>
<td>to</td>
<td>the</td>
<td>witch</td>
<td>green</td>
</tr>
<tr>
<td>did not</td>
<td></td>
<td>a</td>
<td>slap</td>
<td>by</td>
<td></td>
<td></td>
<td>green witch</td>
<td></td>
</tr>
<tr>
<td>no</td>
<td></td>
<td>slap</td>
<td></td>
<td>to the</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>did not give</td>
<td></td>
<td></td>
<td></td>
<td>to</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>slap</td>
<td></td>
<td>the</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Mary did not give a slap to the witch.

Joe did not give p=0.092

Mary did not give p=0.0534

Mary did not give p=0.164

Mary gave p=0.092
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.*

**Gloss**

Gracias
that
lo
of
don [first; future]
muy
very
buen
good
grado
degree
What Happens in Practice

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

<table>
<thead>
<tr>
<th>Gracias</th>
<th>Thanks</th>
</tr>
</thead>
<tbody>
<tr>
<td>,</td>
<td>,</td>
</tr>
<tr>
<td>lo</td>
<td>that</td>
</tr>
<tr>
<td>haré</td>
<td>do [first; future]</td>
</tr>
<tr>
<td>de</td>
<td>of</td>
</tr>
<tr>
<td>muy</td>
<td>very</td>
</tr>
<tr>
<td>buen</td>
<td>good</td>
</tr>
<tr>
<td>grado</td>
<td>degree</td>
</tr>
</tbody>
</table>

Thank you, I shall do so gladly.
What Happens in Practice

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

<table>
<thead>
<tr>
<th>Gracias</th>
<th>Thanks</th>
</tr>
</thead>
<tbody>
<tr>
<td>,</td>
<td>,</td>
</tr>
<tr>
<td>lo</td>
<td>that</td>
</tr>
<tr>
<td>haré</td>
<td>do [first; future]</td>
</tr>
<tr>
<td>de</td>
<td>of</td>
</tr>
<tr>
<td>muy</td>
<td>very</td>
</tr>
<tr>
<td>buen</td>
<td>good</td>
</tr>
<tr>
<td>grado</td>
<td>degree</td>
</tr>
</tbody>
</table>

Thank you, I shall do so gladly.
What Happens in Practice

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

Gloss

Gracias
Thanks

, 
,

lo
that

dé

haré
do [first; future]

muy
of

dear
very

to

buen
good

de

grado
degree

Thank you , I shall do so gladly .
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]
Phrases do help

- But they don’t need to be long
- Why should this be?
**Lexical Weighting**

\[
\phi(f_i | e_i) = \frac{\text{count}(f_i, e_i)}{\text{count}(e_i)} \times p_w(f_i | e_i)
\]

- \( f1 \) \( f2 \) \( f3 \)
- NULL --- --- ##
- e1 ## --- --
- e2 -- ## --
- e3 -- ## --

\[
p_w(f | e, a) = p_w(f_1 f_2 f_3 | e_1 e_2 e_3, a) = w(f_1 | e_1) \\
\times \frac{1}{2} (w(f_2 | e_2) + w(f_2 | e_3)) \\
\times w(f_3 | \text{NULL})
\]

**Graph:**
- **BLEU**
- \( \text{10k} \) \( \text{20k} \) \( \text{40k} \) \( \text{80k} \) \( \text{160k} \) \( \text{320k} \)
- ----- lex
- ------ no-lex
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 segmentations
  - 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

![Model Score vs \( \theta \)]
MERT
MERT
Translating with Tree Transducers

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>lo haré</td>
<td>de muy buen grado</td>
</tr>
</tbody>
</table>

Grammar
Translating with Tree Transducers

Input

lo haré de muy buen grado.

Output

Grammar

ADV $\rightarrow$ { de muy buen grado; gladly }
Syntactic Models
Translating with Tree Transducers

Input          Output

```
<table>
<thead>
<tr>
<th>ADV</th>
</tr>
</thead>
<tbody>
<tr>
<td>lo haré de muy buen grado .</td>
</tr>
</tbody>
</table>
```

Grammar

```
ADV → 〈 de muy buen grado ; gladly 〉
```
Translating with Tree Transducers

**Input**

<table>
<thead>
<tr>
<th>ADV</th>
</tr>
</thead>
<tbody>
<tr>
<td>lo haré</td>
</tr>
<tr>
<td>de muy buen grado</td>
</tr>
</tbody>
</table>

**Output**

<table>
<thead>
<tr>
<th>ADV</th>
</tr>
</thead>
<tbody>
<tr>
<td>gladly</td>
</tr>
</tbody>
</table>

**Grammar**

\[
S \rightarrow \langle \text{lo haré ADV} \ ; \ I \text{ will do it ADV} \ ; \rangle \\
ADV \rightarrow \langle \text{de muy buen grado} ; \text{gladly} \ ; \rangle
\]
Translating with Tree Transducers

**Input**

<table>
<thead>
<tr>
<th>S</th>
<th>ADV</th>
</tr>
</thead>
<tbody>
<tr>
<td>lo haré</td>
<td>de muy buen grado</td>
</tr>
</tbody>
</table>

**Output**

```
S
  
  ADV
  I will do it
  gladly
```

**Grammar**

\[
S \rightarrow \langle \text{lo haré ADV . ; I will do it ADV .} \rangle
\]

\[
ADV \rightarrow \langle \text{de muy buen grado ; gladly} \rangle
\]
Translating with Tree Transducers

**Input**

<table>
<thead>
<tr>
<th>S</th>
<th>ADV</th>
</tr>
</thead>
<tbody>
<tr>
<td>lo haré</td>
<td>de muy buen grado</td>
</tr>
</tbody>
</table>

**Output**

```
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**

| ADV | lo haré | de muy buen grado |

**Output**

| ADV | I | gladly |

**Grammar**

\[
S \rightarrow \{ \text{lo haré ADV . ; I will do it ADV .} \} \\
ADV \rightarrow \{ \text{de muy buen grado ; gladly} \}
\]
Translating with Tree Transducers

**Input**

<table>
<thead>
<tr>
<th>ADV</th>
<th>de muy buen grado</th>
</tr>
</thead>
<tbody>
<tr>
<td>lo haré</td>
<td>.</td>
</tr>
</tbody>
</table>

**Output**

| ADV | I gladly |

**Grammar**

\[
\begin{align*}
  VP & \rightarrow \langle \text{lo haré ADV ; will do it ADV} \rangle \\
  S & \rightarrow \langle \text{lo haré ADV . ; I will do it ADV .} \rangle \\
  ADV & \rightarrow \langle \text{de muy buen grado ; gladly} \rangle
\end{align*}
\]
Translating with Tree Transducers

**Input**

<table>
<thead>
<tr>
<th>VP</th>
<th>ADV</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>lo haré</em></td>
<td><em>de muy buen grado</em></td>
</tr>
</tbody>
</table>

**Output**

```
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**

<table>
<thead>
<tr>
<th></th>
<th>VP</th>
</tr>
</thead>
<tbody>
<tr>
<td>lo haré</td>
<td>de muy buen grado</td>
</tr>
</tbody>
</table>

**Output**

VP

will do it gladly

**Grammar**

\[
S \rightarrow \langle \text{VP} ; \text{I VP} \rangle
\]

\[
\text{VP} \rightarrow \langle \text{lo haré ADV} ; \text{will do it ADV} \rangle
\]

\[
S \rightarrow \langle \text{lo haré ADV} ; \text{I will do it ADV} \rangle
\]

\[
\text{ADV} \rightarrow \langle \text{de muy buen grado} ; \text{gladly} \rangle
\]
Translating with Tree Transducers

**Input**

```
S

VP

lo haré de muy buen grado
```

**Output**

```
S

VP

ADV

I will do it gladly
```

**Grammar**

\[
S \rightarrow \langle \text{ VP } ; \text{ I VP } \rangle \\
VP \rightarrow \langle \text{ lo haré ADV ; will do it ADV } \rangle \\
S \rightarrow \langle \text{ lo haré ADV } ; \text{ I will do it ADV } \rangle \\
ADV \rightarrow \langle \text{ de muy buen grado ; gladly } \rangle
\]
Translating with Tree Transducers

Input

```
S
├── VP
│   ├── ADV
│       └── lo haré de muy buen grado
```

Output

```
S
├── VP
│   ├── ADV
│       └── I will do it
tygladly
```

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

Grammar Rules

Gracias, lo haré de muy buen grado.

Thank you, I will do it gladly.
Learning Grammars for Translation

Thank you, I will do it gladly.

Gracias, lo haré de muy buen grado.
Learning Grammars for Translation

Grammar Rules

〈haré ; will do〉

Gracias
, lo
haré
de
muy
buen
grado
.
Learning Grammars for Translation

Grammar Rules

(haré ; will do)

Thank you, I will do it gladly.

Gracias,
lo haré de muy buen grado.
Learning Grammars for Translation

Grammar Rules

(haré ; will do)

Gracias, lo haré de muy buen grado.
Learning Grammars for Translation

Grammar Rules

<haré ; will do>

VP →

<lo haré de ... grado ; will do it gladly>
Learning Grammars for Translation

Thank you, I will do it gladly.

Gracias, lo haré de... grado; will do it gladly.

Grammar Rules

VP → ...

(“haré” ; will do)
Learning Grammars for Translation

Grammar Rules

haré ; will do

VP →

lo haré de ... grado ;
will do it gladly

Gracias,
lo haré
de muy
grado.

Thank you,
I will do it gladly.
Learning Grammars for Translation

Grammar Rules

\[
\begin{align*}
\langle \text{haré} \rangle & \rightarrow \langle \text{will do} \rangle \\
\langle \text{lo haré de ... grado} \rangle & \rightarrow \langle \text{will do it gladly} \rangle
\end{align*}
\]
Learning Grammars for Translation

Grammar Rules

\[ \langle \text{haré} ; \text{will do} \rangle \]

\[
\begin{align*}
\text{VP} & \rightarrow \\
\langle \text{lo haré de ... grado} ; \\
& \text{will do it gladly} \rangle \\
\text{ADV} & \rightarrow \\
\langle \text{lo haré ADV} ; \\
& \text{will do it ADV} \rangle
\end{align*}
\]
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

Size of the source-side yield

<table>
<thead>
<tr>
<th>Size</th>
<th>Rule Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>200</td>
</tr>
<tr>
<td>4</td>
<td>300</td>
</tr>
<tr>
<td>5</td>
<td>400</td>
</tr>
<tr>
<td>6</td>
<td>500</td>
</tr>
<tr>
<td>7</td>
<td>600</td>
</tr>
<tr>
<td>8</td>
<td>700</td>
</tr>
<tr>
<td>9</td>
<td>800</td>
</tr>
<tr>
<td>10+</td>
<td>900</td>
</tr>
</tbody>
</table>
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

![Bar chart showing rule counts for different sizes of the source-side yield.]

- $S \rightarrow NP \, VP \, ; \, NP \, VP$
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

\[
S \rightarrow NP \text{ no es ni ADJP ni ADJP . ;}
\]

\[
NP \text{ isn’t ADJP or ADJP .}
\]

\[
S \rightarrow NP \text{ VP ; NP VP}
\]

Rule Count

Size of the source-side yield
Syntactic Decoding
Tree Transducer Grammars

S

**Synchronous Grammar**

\[ \text{NNP} \rightarrow \text{Colorado} ; \text{Colorado} \]

\[ \text{NN} \rightarrow \text{canto rodado} ; \text{boulder} \]

\[ S \rightarrow \text{No se olvide de subir un} \text{ NN en NNP} ; \text{Don’t forget to climb a} \text{ NN in NNP} \]

**Output**

\[ S \]

\[ \text{NN} \quad \text{NNP} \]

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: \( X \rightarrow Y Z \)
CKY-style Bottom-up Parsing

For each span length:

For each span \([i,j]\):

Apply all grammar rules to \([i,j]\)

Binary rule: \(X \rightarrow Y Z\)

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 \([i,j]\)

\[\text{i No se olvide de subir un canto rodado en Colorado } j\]
CKY-style Bottom-up Parsing

For each span length:

For each span \([i,j]\):

Apply all grammar rules to \([i,j]\)

\[
S \rightarrow \text{No se } \text{VB de subir un } \text{NN en } \text{NNP}
\]

\(i\) 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]

\[ S \rightarrow \text{No se } \textbf{VB} \text{ de subir un } \textbf{NN} \text{ en } \textbf{NNP} \]

\[ i \text{ No se olvide de subir un canto rodado en Colorado } j \]
CKY-style Bottom-up Parsing

For each span length:

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

$$S \rightarrow \text{No se } \textbf{VB} \text{ de subir un } \textbf{NN} \text{ en } \textbf{NNP}$$

$i$ 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 \([i,j]\)

\[ S \rightarrow \text{No se } \text{VB } \text{de subir un } \text{NN } \text{en } \text{NNP} \]

\( S \rightarrow \text{No se olvide } \text{de subir un canto rodado en Colorado} \)
CKY-style Bottom-up Parsing

For each span length:

For each span \([i,j]\):
Apply all grammar rules to \([i,j]\)

\[S \rightarrow \text{No se } \text{VB } \text{de subir un } \text{NN } \text{en } \text{NNP}\]

\[\text{No se olvide de subir un canto rodado en Colorado}\]
CKY-style Bottom-up Parsing

For each span length:

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

\[ S \rightarrow \text{No se VB de subir un NN en NNP} \]

\[ \text{No se olvide de subir un canto rodado en Colorado} \]
CKY-style Bottom-up Parsing

For each span length:

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

\[S \rightarrow \text{No se } \text{VB } \text{de subir un } \text{NN } \text{en } \text{NNP}\]

\[i \text{ No se olvide de subir un canto rodado en Colorado } j\]

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 \([i,j]\)

\[ S \rightarrow \text{No se} \quad \text{VP} \quad \text{NP} \quad \text{PP} \]

\[ i \text{ No se olvide de subir un canto rodado en Colorado } j \]
CKY-style Bottom-up Parsing

For each span length:

For each span $[i,j]$:

Apply all grammar rules to $[i,j]$

$S \rightarrow \text{No se } \text{VP NP PP}$

$\text{i No se olvide de subir un canto rodado en Colorado j}$
CKY-style Bottom-up Parsing

For each span length:

For each span \([i,j]\):

Apply all grammar rules to \([i,j]\)

\[
S \rightarrow \text{No se} \quad \text{VP} \quad \text{NP} \quad \text{PP}
\]

\[i \quad \text{No se olvide de subir un canto rodado en Colorado} \quad j\]
CKY-style Bottom-up Parsing

For each span length:

For each span \([i,j]\):
Apply all grammar rules to \([i,j]\)

\[ S \rightarrow \text{No se} \quad \text{VP} \quad \text{NP} \quad \text{PP} \]

\[ \text{No se olvide de subir un canto rodado en Colorado} \]

**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: \[ S \rightarrow \text{No se} \ VB \ VB \ un \ NN \ PP \]

Transformed rules: \[ S \rightarrow \text{No se} \ VB\sim VB \ un \ NN\sim PP \]
\[ VB\sim VB \rightarrow \ VB \ VB \]
\[ NN\sim PP \rightarrow \ NN \ PP \]

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

$S \rightarrow$ No se olvide de subir $NP$

$i$ No se olvide de subir un canto rodado en Colorado $j$
Speeding up Lexical Rule Application

**Problem:** Lexical rules can apply to many spans

\[ S \rightarrow \text{No se olvide de subir} \quad NP \]

\[ S \rightarrow \text{No se olvide de subir un canto rodado en Colorado} \]

\[ i \quad \text{No se olvide de subir} \quad j \]
Speeding up Lexical Rule Application

**Problem:** Lexical rules can apply to many spans

\[ S \rightarrow \text{No se olvide de subir} \quad \text{NP} \]

\[ i \text{ No se olvide de subir un canto rodado } j \text{ en Colorado} \]
Speeding up Lexical Rule Application

**Problem:** Lexical rules can apply to many spans

\[ S \rightarrow \text{No se olvide de subir} \quad \text{NP} \]

\[
\begin{align*}
&\text{No se olvide de subir} \\
&\quad \text{un canto} \\
&\quad \text{rodado en Colorado}
\end{align*}
\]
Flexible Syntax
Soft Syntactic MT: From Chiang 2010

Japanese 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.
## Previous work

<table>
<thead>
<tr>
<th>Type</th>
<th>Method</th>
<th>Year</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>String-to-string</td>
<td>ITG (Wu 1997)</td>
<td></td>
<td>Hiero</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(Chiang 2005)</td>
</tr>
<tr>
<td>String-to-tree</td>
<td>Yamada &amp; Knight</td>
<td>2001</td>
<td>Galley et al</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2004/2006</td>
</tr>
<tr>
<td>Tree-to-string</td>
<td>DOT (Poutsma 2000)</td>
<td></td>
<td>Huang et al</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Eisner 2003</td>
<td>Y Liu et al</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2006</td>
</tr>
<tr>
<td>Tree-to-tree</td>
<td>Stat-XFER (Lavie et al 2008)</td>
<td></td>
<td>M Zhang et al</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2008</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Y Liu et al., 2009</td>
</tr>
</tbody>
</table>
Hiero Rules

\[ S \rightarrow \langle S_1 X_2, S_1 X_2 \rangle \]
\[ S \rightarrow \langle X_1, X_1 \rangle \]
\[ X \rightarrow \langle yu X_1 \text{ you } X_2, have X_2 \text{ with } X_1 \rangle \]
\[ X \rightarrow \langle X_1 \text{ de } X_2, the X_2 \text{ that } X_1 \rangle \]
\[ X \rightarrow \langle X_1 \text{ zhiyi, one of } X_1 \rangle \]
\[ X \rightarrow \langle Aozhou, Australia \rangle \]
\[ X \rightarrow \langle shi, is \rangle \]
\[ X \rightarrow \langle shaoshu guojia, few countries \rangle \]
\[ X \rightarrow \langle bangjiao, diplomatic relations \rangle \]
\[ X \rightarrow \langle Bei Han, North Korea \rangle \]

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

Diagram:

```
NP
  /\    /
 a la bruja verde
  /\    /
the green witch
  /\    /
DT JJ NN
```

```
NP
  /\    /
QP  NNS
  /\    /
JJR IN CD
   /\    /
more than 20
   /\    /
NN NNS
   /\    /
check points
```
Extracting more rules

binarize head-out
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

<table>
<thead>
<tr>
<th>Hiero-like decoding</th>
<th>([X,i,j])</th>
<th>([X,j+1,k])</th>
<th>(\rightarrow) X 的 X</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>([X,i,k])</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STSG decoding</th>
<th>([VP,i,j])</th>
<th>([NP,j+1,k])</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>([NP,i,k])</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fuzzy STSG decoding</th>
<th>([A,i,j])</th>
<th>([B,j+1,k])</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>([NP,i,k])</td>
<td></td>
<td>的</td>
</tr>
</tbody>
</table>

Diagram:
- NP
- VP
- CP
- DEC
- NP
## Results

<table>
<thead>
<tr>
<th>extraction</th>
<th>Chinese-English</th>
<th>Arabic-English</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>rules</td>
<td>feats</td>
</tr>
<tr>
<td>Hiero</td>
<td>440M</td>
<td>1k</td>
</tr>
<tr>
<td>fuzzy STSG</td>
<td>50M</td>
<td>5k</td>
</tr>
<tr>
<td>fuzzy STSG +binarize</td>
<td>64M</td>
<td>5k</td>
</tr>
<tr>
<td>fuzzy STSG +SAMT</td>
<td>440M</td>
<td>160k</td>
</tr>
</tbody>
</table>
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."
