Phrase-Based MT

Phrase-Based Translation Overview

**Input:**
- To hard | "rapidamente"
  - tries different segmentations,

**Translations:**
- I'll do it | quickly | 
- quickly | I'll do it |
  - translates phrase by phrase,
  - and considers reorderings.

**Objective:**
$$\arg \max_e [P(f) \cdot P(e)]$$
$$\arg \max_e \left[ \prod_{(e_i, f)} P(f | e_i) \cdot \prod_{i=1}^{n} P(e_i, e_{i-1}, e_{i-2}) \right]$$

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

Phrase-Based Decoding

```
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+eOption] = max score
```

\[\ldots \text{a slap, 5}\]
\[\ldots \text{slap to, 6}\]
\[\ldots \text{slap by, 6}\]

Monotonic Word Translation

- Cost is LM * TM
- It's an HMM?
  - P(e|e_{i-1}, e_{i-2})
  - P(f|e)
- State includes
  - Exposed English
  - Position in foreign
- Dynamic program loop?
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 (Position in 1...8)  
  for (eContext in bestEContexts[Position])  
    for (eOption in translations[Position])  
      score = scores[Position-1][eContext] * LM(eContext+eOption) * TM(eOption, fWord[Position])
      bestEContexts.maybeAdd(eContext+eOption, score)
```

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

Phrase Translation

- If monotonic, almost an HMM; technically a semi-HMM
- If distortion... now what?

```
for (Position in 1...8)  
  for (eContext in eContexts)  
    for (eOption in translations[Position])  
      ... combine hypothesis for (lastPosition ending in eContext) with eOption
```

Non-Monotonic Phrasal MT

- If monotonic, almost an HMM; technically a semi-HMM
- If distortion... now what?

```
for (Position in 1...8)  
  for (lastPosition < Position)  
    for (eContext in eContexts)  
      ... combine hypothesis for (lastPosition ending in eContext) with eOption
```

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

- The Pharaoh Decoder Hypothesis Lattices
Parameter Tuning

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

\[ \phi_{new}(\tilde{f}_i) = \frac{e^x}{e^{(f_i)} + e^{(f_j)}} \]

- Learning weights has been tried, several times:
  - [Marco 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., 06]
Phrase Size

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

Lexical Weighting

\[ \phi(f, a) = \frac{\text{count}(f, a)}{\text{count}(a)} \]

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)

- 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

Why Tuning is Hard

- 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
Translating with Tree Transducers

Input

Output

lo haré de muy buen grado.

Grammar

adv → (de muy buen grado 1 gladly)

Syntactic Models
Learning Grammars for Translation

Grammar Rules

- (hare + will do)
- (lo hare de ... grado ; will do it gladly)
- (lo hare ADV ; will do it ADV)

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

Syntactic Decoding
Tree Transducer Grammars

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

Synchronous Grammar

\[ \text{NN} \rightarrow \text{Colorado} \quad ; \quad \text{Colorado} \]
\[ \text{NN} \rightarrow \text{canto rodado} \quad ; \quad \text{boulder} \]
\[ S \rightarrow \text{No se olvide de subir un NN en NNP} \quad ; \quad \text{Don't forget to climb a NN in NNP} \]

Output

\[ S \rightarrow \text{NN} \rightarrow \text{NNP} \]
Don't forget to climb a boulder in Colorado

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
No se olvide de subir un canto rodado en Colorado.
CKY-style Bottom-up Parsing

For each span length:

For each span \([ij]\):

Apply all grammar rules to \([ij]\)

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

\text{No se olvide de subir un canto rodado en Colorado}_{ji}

Many untransformed lexical rules can be applied in linear time

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: $S \rightarrow \text{No se} \ VB \ VB \ \text{un} \ NN \ PP$

Transformed rules:
- $S \rightarrow \text{No se} \ VB-VB \ \text{un} \ NN-PP$
- $VB-VB \rightarrow \ VB \ VB$
- $NN-PP \rightarrow \ NN \ PP$

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

Original rule: $S \rightarrow \text{No se olvide de subir} \ NP$

Transformed rule: $S \rightarrow \text{No se olvide de subir un canto rodado en Colorado}$

Flexible Syntax
Soft Syntactic MT: From Chiang 2010

Hierarchical Rules

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

STSG extraction

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

Previous work

- string-to-string: ITG (Wu 1997)
- Hiero (Chiang 2003)
- string-to-tree: Yamada & Knight 2001
- tree-to-string: Huang et al 2006
- Y Liu et al 2006
- tree-to-tree: DOT (Pourdehdust 2000)
- Eisele 2003
- Stat-XFER (Levie et al 2008)
- M Zhang et al 2008
- Y Liu et al, 2009
Why is tree-to-tree hard?

too few rules

too few derivations

NP

a la brevi verdes

the green witch

DT

JJ

NN

IN

CD

more than 20

NP

JJR

IN

CD

more than 20

NP

JJR

IN

CD

more than 20

NP

JJR

IN

CD

more than 20

NP

JJR

IN

CD

more than 20

NP

check points

Extracting more rules

NP

PP

VP

check points

binary head-out

NP

check points

Allow more derivations

NP

QP

J JR

IN

CD

more than 20

NP

NN

NNS

check points

STSG: allow only matching substitutions

Hierarchically allow any substitutions

Let the model learn to choose:

- matching substitutions
- mismatching substitutions
- monotone phrase-based

Allow more derivations

NP

QP

J JR

IN

CD

more than 20

NP

NN

NNS

check points

Allow more derivations

Hierarchical decoding

\[[X_{ij}]\]

\[[X_{ij}, X_{j+1,k}]\]

\([X_{i,k}]\]

\([NP_{i,k}]\]

STSG decoding

\([VP_{i,j}]\]

\([NP_{i,j}+1,k]\]

\([NP_{i,k}]\]

\([VP_{i,k}]\]

fuzzy STSG decoding

\([A_{ij}]\]

\([B_{ij}+1,k]\]

\([NP_{i,k}]\]

\([NP_{i,j}]\]

Results

<table>
<thead>
<tr>
<th></th>
<th>Chinese-English</th>
<th>Arabic-English</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>extraction</td>
<td>rules</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

Japanese: 日本 文部科学省 官員 官員, 今、私人言及 本部

English: An official said, "Abraham's comment made us deeply feel courage.

English: "Officials of the Japanese ministry of education and science, said, "We are highly encouraged by Abraham's comment."

English: "Officials of the Japanese ministry of education, culture, sports, science and technology, said, "We are very encouraged by the speeches of Abraham."

Edge STSG: 本部