Phrase Based Translation

Luke Zettlemoyer

Slides from Philipp Koehn and Dan Klein
Phrase-Based Systems

Morgen ||| fliege ||| ich ||| nach Kanada ||| zur Konferenz

Tomorrow ||| I ||| will fly ||| to the conference ||| in Canada

Sentence-aligned corpus ||| Word alignments

Phrase table (translation model)

cat ||| chat ||| 0.9
the cat ||| le chat ||| 0.8
dog ||| chien ||| 0.8
house ||| maison ||| 0.6
my house ||| ma maison ||| 0.9
language ||| langue ||| 0.9
...
Phrase Translation Tables

- Defines the space of possible translations
  - each entry has an associated “probability”

- One learned example, for “den Vorschlag” from Europarl data

| English            | $\phi(\bar{e}|f)$ | English            | $\phi(\bar{e}|f)$ |
|--------------------|-------------------|--------------------|-------------------|
| the proposal       | 0.6227            | the suggestions    | 0.0114            |
| ’s proposal        | 0.1068            | the proposed       | 0.0114            |
| a proposal         | 0.0341            | the motion         | 0.0091            |
| the idea           | 0.0250            | the idea of        | 0.0091            |
| this proposal      | 0.0227            | the proposal ,     | 0.0068            |
| proposal           | 0.0205            | its proposal       | 0.0068            |
| of the proposal    | 0.0159            | it                 | 0.0068            |
| the proposals      | 0.0159            | ...               | ...               |

- This table is noisy, has errors, and the entries do not necessarily match our linguistic intuitions about consistency....
Phrase-Based Decoding

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

- We will use word alignments to find phrases

<table>
<thead>
<tr>
<th>Mary</th>
<th>did</th>
<th>not</th>
<th>slap</th>
<th>the</th>
<th>green</th>
<th>witch</th>
</tr>
</thead>
<tbody>
<tr>
<td>María</td>
<td>no</td>
<td>daba</td>
<td>una</td>
<td>bofetada</td>
<td>a</td>
<td>la</td>
</tr>
</tbody>
</table>

Question: what is the best set of phrases?
Extracting Phrases

- **Phrase alignment must**
  - Contain at least two aligned words
  - Contain all alignments for phrase pair

- Extract all such phrase pairs!
Phrase Pair Extraction Example

(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green)

(Maria no, Mary did not), (no daba una bofetada, did not slap), (daba una bofetada a la, slap the), (bruja verde, green witch)

(Maria no daba una bofetada, Mary did not slap), (no daba una bofetada a la, did not slap the), (a la bruja verde, the green witch)

(Maria no daba una bofetada a la, Mary did not slap the), (daba una bofetada a la bruja verde, slap the green witch)

(Maria no daba una bofetada a la bruja verde, Mary did not slap the green witch)
Phrase Size

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

english to spanish

Spanish to English

Intersection
Alignment Heuristics
Phrase Scoring

\[ g(f, e) = \log \frac{c(e, f)}{c(e)} \]

- 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]
Scoring:

- Basic approach, sum up phrase translation scores and a language model
  - Define $y = p_1p_2...p_L$ to be a translation with phrase pairs $p_i$
  - Define $e(y)$ be the output English sentence in $y$
  - Let $h()$ be the log probability under a tri-gram language model
  - Let $g()$ be a phrase pair score (from last slide)
  - Then, the full translation score is:
    \[
    f(y) = h(e(y)) + \sum_{k=1}^{L} g(p_k)
    \]
- Goal, compute the best translation
  \[
  y^*(x) = \arg \max_{y \in \mathcal{Y}(x)} f(y)
  \]
The Pharaoh Decoder

- Scores at each step include LM and TM
In practice, much like for alignment models, also include a distortion penalty
- Define $y = p_1 p_2 \ldots p_L$ to be a translation with phrase pairs $p_i$
- Let $s(p_i)$ be the start position of the foreign phrase
- Let $t(p_i)$ be the end position of the foreign phrase
- Define $\eta$ to be the distortion score (usually negative!)
- Then, we can define a score *with distortion penalty*:

$$f(y) = h(e(y)) + \sum_{k=1}^{L} g(p_k) + \sum_{k=1}^{L-1} \eta \times |t(p_k) + 1 - s(p_{k+1})|$$

Goal, compute the best translation

$$y^*(x) = \arg\max_{y \in \mathcal{Y}(x)} f(y)$$
• ... until all foreign words **covered**
  – find *best hypothesis* that covers all foreign words
  – *backtrack* to read off translation
Hypothesis Explosion!

- **Q:** How much time to find the best translation?
  - NP-hard, just like for word translation models
  - So, we will use approximate search techniques!
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>
</table>

Mary did not give a slap to the witch green
did not a slap by green witch
no slap to the
did not give

Joe

Mary
did not give

did not
give

p=0.092

p=1

p=0.534

p=0.164
Pruning

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

Hypothesis recombination is not sufficient
⇒ Heuristically discard weak hypotheses early
- Organize hypotheses in priority queues, e.g.
  - same foreign words covered
  - same number of foreign words covered
  - same number of English words produced
- Compare hypotheses in queue, discard bad ones
  - histogram pruning: keep top n hypotheses in each queue e.g., n = 100
  - threshold pruning: keep hypotheses that are at most $\alpha$ times the cost of best hypothesis in queue e.g., $\alpha = 0.001$

Miles Osborne Machine Translation 16 February 2012
Tons of Data?

- Discussed for LMs, but can new understand full model!
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)

\[ x: \text{le parlement, adopte, la, résolution, législative} \]

\[ h: \]

\[ y: \text{parliament, has, adopted, the, resolution} \]
Why Tuning is Hard

- Problem 2: There are many right answers
  - The reference or references are just a few options
  - No good characterization of the whole class

- BLEU isn’t perfect, but even if you trust it, it’s a corpus-level metric, not sentence-level
Perceptron training

For each training example \((x, y)\): [Collins ’02]

\[
\begin{align*}
  w &\leftarrow w + \Phi(x, y_t) & y_t &= y \\
  &\quad -\Phi(x, y_p) & y_p &= \text{DECODE}(x)
\end{align*}
\]

\[
\begin{align*}
  w &\leftarrow w + \Phi(x, y_t, h_t) & y_t, h_t &= ??? \\
  &\quad -\Phi(x, y_p, h_p) & y_p, h_p &= \text{DECODE}(x)
\end{align*}
\]
Update strategies

\[ w \leftarrow w + \Phi(x, y_t, h_t) - \Phi(x, y_p, h_p) \]

Training example (reference)

- \( x \): voté sur demande d’urgence
- \( y \): vote on a request for urgent procedure
Update strategies

\[ w \leftarrow w + \Phi(x, y_t, h_t) - \Phi(x, y_p, h_p) \]

Training example (reference)

- **x**: voté sur demande d'urgence
- **y**: vote on a request for urgent procedure

Reachable translations

**x**: voté sur demande d'urgence

**x**: vote on emergency request

Current prediction

**y_p**: vote on emergency request

**y_t**: vote on a request for urgent procedure

Bold update
Update strategies

\[ w \leftarrow w + \Phi(x, y_t, h_t) - \Phi(x, y_p, h_p) \]

Training example (reference)
\[ x: \text{voté sur demande d’urgence} \]
\[ y: \text{vote on a request for urgent procedure} \]

Reachable translations

\[ x: \text{voté sur demande d’urgence} \]
\[ h_t: \]
\[ y_t: \text{vote on an urgent request} \]

Local update

Current prediction
\[ x: \text{voté sur demande d’urgence} \]
\[ h_p: \]
\[ y_p: \text{vote on emergency request} \]

Bold update
\[ x: \text{voté sur demande d’urgence} \]
\[ h_t: \]
\[ y_t: \text{vote on a request for urgent procedure} \]
Update strategies

\[ w \leftarrow w + \Phi(x, y_t, h_t) - \Phi(x, y_p, h_p) \]

Training example (reference)
- \( x \): voté sur demande d'urgence
- \( y \): vote on a request for urgent procedure

Reachable translations

- \( x \): voté sur demande d'urgence
- \( h_t \):
- \( y_t \): vote on an urgent request

Local update

Current prediction
- \( x \): voté sur demande d'urgence
- \( h_p \): vote on emergency request
- \( y_p \): vote on emergency request

Bold update: skip example
Update strategies

\[ \mathbf{w} \leftarrow \mathbf{w} + \Phi(\mathbf{x}, \mathbf{y}_t, \mathbf{h}_t) - \Phi(\mathbf{x}, \mathbf{y}_p, \mathbf{h}_p) \]

Training example (reference)
- \( \mathbf{x} \): voté sur demande d’urgence
- \( \mathbf{y} \): vote on a request for urgent procedure

<table>
<thead>
<tr>
<th>Decoder</th>
<th>Bold</th>
<th>Local</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monotonic</td>
<td>34.3</td>
<td>34.6</td>
</tr>
<tr>
<td>Limited distortion</td>
<td>33.5</td>
<td>34.7</td>
</tr>
</tbody>
</table>

Current prediction

Bold update: skip example
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)
MERT

Model Score vs. $\theta$

BLEU Score vs. $\theta$
MERT