Phrase-Based MT: Decoding

February 19, 2015
Administrative

- Final proposal draft due Tuesday
- It needs to be revised
- Bring 3 printed copies again
- HW 2 is due two weeks from today
Phrase Based MT

\[ e^* = \arg \max_e p(e \mid f) \]
\[ = \arg \max_e p(f \mid e) \times p(e) \]
\[ \approx \arg \max_e p(f, a \mid e) \times p(e) \]

• Recipe: Ingredients
• Segmentation / Reordering model
• Phrase model
• Language Model
Marginal Decoding

\[ e^* = \arg \max_e p(e \mid f) \]
\[ = \arg \max_e p(f \mid e) \times p(e) \]
\[ \approx \arg \max_e p(f, a \mid e) \times p(e) \]

Does this last approximation matter?

- Variational & MCMC explored
- *slight* benefits, depending on training
- Really hard problem (Sima’an, 1997)
Reordering Model

Scoring function: $d(x) = \alpha^{|x|}$ — exponential with distance
## Phrase Tables

<table>
<thead>
<tr>
<th>( \bar{f} )</th>
<th>( \bar{e} )</th>
<th>( p(\bar{f} \mid \bar{e}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>das Thema</td>
<td>the issue</td>
<td>0.41</td>
</tr>
<tr>
<td></td>
<td>the point</td>
<td>0.72</td>
</tr>
<tr>
<td></td>
<td>the subject</td>
<td>0.47</td>
</tr>
<tr>
<td></td>
<td>the thema</td>
<td>0.99</td>
</tr>
<tr>
<td>es gibt</td>
<td>there is</td>
<td>0.96</td>
</tr>
<tr>
<td></td>
<td>there are</td>
<td>0.72</td>
</tr>
<tr>
<td>morgen</td>
<td>tomorrow</td>
<td>0.9</td>
</tr>
<tr>
<td>fliege ich</td>
<td>will I fly</td>
<td>0.63</td>
</tr>
<tr>
<td></td>
<td>will fly</td>
<td>0.17</td>
</tr>
<tr>
<td></td>
<td>I will fly</td>
<td>0.13</td>
</tr>
</tbody>
</table>
Recipe: Instructions
Translation Process

• Task: translate this sentence from German into English

er  geht  ja  nicht  nach  hause
Translation Process

- Task: translate this sentence from German into English

er geht ja nicht nach hause

- Pick phrase in input, translate
Translation Process

• Task: translate this sentence from German into English

er geht ja nicht nach hause

- it is allowed to pick words out of sequence reordering
- phrases may have multiple words: many-to-many translation
Translation Process

• Task: translate this sentence from German into English

er geht ja nicht nach hause

he does not go

• Pick phrase in input, translate
Translation Process

• Task: translate this sentence from German into English

er
gernt
ja
nicht
nach
hausen

- er
- geht
- ja
- nicht
- nach
- hause

he
does not
go
home

• Pick phrase in input, translate
Computing Translation Probability

- Probabilistic model for phrase-based translation:
  \[ e_{\text{best}} = \arg\max_e \prod_{i=1}^{I} \phi(f_i|e_i) \, d(\text{start}_i - \text{end}_{i-1} - 1) \, p_{LM}(e) \]

- Score is computed incrementally for each partial hypothesis

- Components
  - **Phrase translation** Picking phrase \( f_i \) to be translated as a phrase \( e_i \)
    \( \rightarrow \) look up score \( \phi(f_i|e_i) \) from phrase translation table
  - **Reordering** Previous phrase ended in \( \text{end}_{i-1} \), current phrase starts at \( \text{start}_i \)
    \( \rightarrow \) compute \( d(\text{start}_i - \text{end}_{i-1} - 1) \)
  - **Language model** For \( n \)-gram model, need to keep track of last \( n - 1 \) words
    \( \rightarrow \) compute score \( p_{LM}(w_i|w_{i-(n-1)}, \ldots, w_{i-1}) \) for added words \( w_i \)
Translation Options

<table>
<thead>
<tr>
<th>er</th>
<th>geht</th>
<th>ja</th>
<th>nicht</th>
<th>nach</th>
<th>hause</th>
</tr>
</thead>
<tbody>
<tr>
<td>he</td>
<td>is</td>
<td>yes</td>
<td>not</td>
<td>after</td>
<td>house</td>
</tr>
<tr>
<td>it</td>
<td>are</td>
<td>is</td>
<td>do not</td>
<td>to</td>
<td>home</td>
</tr>
<tr>
<td>, it</td>
<td>goes</td>
<td>, of course</td>
<td>does not</td>
<td>according to</td>
<td>chamber</td>
</tr>
<tr>
<td>, he</td>
<td>go</td>
<td></td>
<td>is not</td>
<td>in</td>
<td>at home</td>
</tr>
<tr>
<td>it is</td>
<td>not</td>
<td>is not</td>
<td>not after</td>
<td>home</td>
<td>return home</td>
</tr>
<tr>
<td>he will be</td>
<td>is not</td>
<td>does not</td>
<td>not to</td>
<td>under house</td>
<td>do not</td>
</tr>
<tr>
<td>it goes</td>
<td>do not</td>
<td>to</td>
<td>following</td>
<td></td>
<td></td>
</tr>
<tr>
<td>he goes</td>
<td>is after all</td>
<td></td>
<td>not after</td>
<td></td>
<td></td>
</tr>
<tr>
<td>does</td>
<td>not</td>
<td>is not</td>
<td>not to</td>
<td></td>
<td></td>
</tr>
<tr>
<td>not</td>
<td>is not</td>
<td>are not</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>is not a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Many translation options to choose from
  - in Europarl phrase table: 2727 matching phrase pairs for this sentence
  - by pruning to the top 20 per phrase, 202 translation options remain
The machine translation decoder does not know the right answer
- picking the right translation options
- arranging them in the right order

→ Search problem solved by heuristic beam search

Chapter 6: Decoding
Decoding algorithm

- **Translation as a search problem**
- **Partial hypothesis** keeps track of:
  - which source words have been translated (*coverage vector*)
  - $n-1$ most recent words of English (for LM!)
  - a *back pointer* list to the previous hypothesis + (e,f) phrase pair used
  - the (partial) translation probability
  - the *estimated probability* of translating the remaining words
    (precomputed, a function of the coverage vector)

- **Start state**: no translated words, $E=<s>$, bp=nil

- **Goal state**: all translated words
Decoding: Precompute Translation Options

consult phrase translation table for all input phrases
Decoding: Start with Initial Hypothesis

initial hypothesis: no input words covered, no output produced
Decoding: Hypothesis Expansion

pick any translation option, create new hypothesis
Decoding: Hypothesis Expansion

create hypotheses for all other translation options
Decoding: Hypothesis Expansion

also create hypotheses from created partial hypothesis
Decoding: Find Best Path

backtrack from highest scoring complete hypothesis
Complexity

- This is an NP-complete problem
  - Reduction to TSP (sketch)
    - Each source word is a city
    - A bigram LM encodes the distance between pairs of cities
  - Knight (1999) has careful proof

- How do we solve such problems?
  - Dynamic programming [risk free]
    - The state is the current city C & the set of previous visited cities
    - Doesn’t matter the order the previous list was visited in as long as we keep the best path to C through
  - How many states are there?
  - Approximate search [risky]
Recombination

- Two hypothesis paths lead to two matching hypotheses
  - same number of foreign words translated
  - same English words in the output
  - different scores

- Worse hypothesis is dropped
Recombination

- Two hypothesis paths lead to hypotheses indistinguishable in subsequent search
  - same number of foreign words translated
  - same last two English words in output (assuming trigram language model)
  - same last foreign word translated
  - different scores

- Worse hypothesis is dropped
Restrictions on Recombination

- **Translation model:** Phrase translation independent from each other
  \[\rightarrow\] no restriction to hypothesis recombination

- **Language model:** Last \(n - 1\) words used as history in \(n\)-gram language model
  \[\rightarrow\] recombined hypotheses must match in their last \(n - 1\) words

- **Reordering model:** Distance-based reordering model based on distance to end position of previous input phrase
  \[\rightarrow\] recombined hypotheses must have that same end position

Chapter 6: Decoding
Pruning

- Recombination reduces search space, but not enough
  (we still have a NP complete problem on our hands)

- Pruning: remove bad hypotheses early
  - put comparable hypothesis into stacks
    (hypotheses that have translated same number of input words)
  - limit number of hypotheses in each stack
• Hypothesis expansion in a stack decoder
  – translation option is applied to hypothesis
  – new hypothesis is dropped into a stack further down
Stack Decoding Algorithm

1: place empty hypothesis into stack 0
2: for all stacks 0…n − 1 do
3:   for all hypotheses in stack do
4:     for all translation options do
5:       if applicable then
6:         create new hypothesis
7:         place in stack
8:         recombine with existing hypothesis if possible
9:         prune stack if too big
10:       end if
11:     end for
12:   end for
13: end for
f: María no dio una bofetada a la bruja verde

Q[0]   Q[1]   Q[2]   ...

e: <s>
c: 

p: 1.0
f: Maria no dio una bofetada a la bruja verde

Maria no dio una bofetada a la bruja verde

\[ Q[0] \quad Q[1] \quad Q[2] \quad \ldots \]

\[ \begin{align*}
\mathbb{e}: & <s> Mary \\
\mathbb{c}: & *-------- \\
p: & 0.9 \\
\end{align*} \]
f: Maria no dio una bofetada a la bruja verde

Q[0]  Q[1]  Q[2]  ...

玛丽

玛丽
f: María no dio una bofetada a la bruja verde

Mary did not

| Q[0] | Q[1] | Q[2] | ...
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mary</td>
<td>Maria</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>0.9</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Mary

Maria

Maria did not

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>did not</td>
<td>0.3</td>
<td></td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>c:</td>
<td>0.3</td>
</tr>
<tr>
<td>p:</td>
<td>0.3</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>c:</td>
<td>0.3</td>
</tr>
<tr>
<td>p:</td>
<td>0.3</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>c:</td>
<td>0.3</td>
</tr>
<tr>
<td>p:</td>
<td>0.3</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>c:</td>
<td>0.3</td>
</tr>
<tr>
<td>p:</td>
<td>0.3</td>
</tr>
</tbody>
</table>
Maria no dio una bofetada a la bruja verde.

Diagrama:
- Fase 0: Mary
- Fase 1: Maria
- Fase 2: did not
- Fase 3: Mary did not

Método Q: [Q[0], Q[1], Q[2], ...]
f: Maria no dio una bofetada a la bruja verde

Q[0]   Q[1]   Q[2]   ...

Mary

Maria
did not

Mary did not
f: María no dio una bofetada a la bruja verde

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>Mary</td>
<td>not</td>
</tr>
<tr>
<td>Maria</td>
<td>Maria</td>
<td>not</td>
</tr>
<tr>
<td></td>
<td></td>
<td>slap</td>
</tr>
</tbody>
</table>
Pruning

- Pruning strategies
  - histogram pruning: keep at most $k$ hypotheses in each stack
  - stack pruning: keep hypothesis with score $\alpha \times$ best score ($\alpha < 1$)

- Computational time complexity of decoding with histogram pruning

  $O(\text{max stack size} \times \text{translation options} \times \text{sentence length})$

- Number of translation options is linear with sentence length, hence:

  $O(\text{max stack size} \times \text{sentence length}^2)$

- Quadratic complexity
Reordering Limits

• Limiting reordering to maximum reordering distance

• Typical reordering distance 5–8 words
  – depending on language pair
  – larger reordering limit hurts translation quality

• Reduces complexity to linear

\[ O(\text{max stack size} \times \text{sentence length}) \]

• Speed / quality trade-off by setting maximum stack size
Translating the Easy Part First?

the tourism initiative addresses this for the first time

both hypotheses translate 3 words
worse hypothesis has better score
Estimating Future Cost

• Future cost estimate: how expensive is translation of rest of sentence?

• Optimistic: choose cheapest translation options

• Cost for each translation option
  – translation model: cost known
  – language model: output words known, but not context
    → estimate without context
  – reordering model: unknown, ignored for future cost estimation
Cost Estimates from Translation Options

the tourism initiative addresses this for the first time

-1.0  -2.0  -1.5  -2.4  -1.4  -1.0  -1.0  -1.9  -1.6
-4.0  -2.5  -2.2  -1.3  -2.4  -2.7  -2.3  -2.3  -2.3

cost of cheapest translation options for each input span (log-probabilities)
Cost Estimates for all Spans

- Compute cost estimate for all contiguous spans by combining cheapest options

<table>
<thead>
<tr>
<th>first word</th>
<th>future cost estimate for ( n ) words (from first)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>the</td>
<td>-1.0</td>
</tr>
<tr>
<td>tourism</td>
<td>-2.0</td>
</tr>
<tr>
<td>initiative</td>
<td>-1.5</td>
</tr>
<tr>
<td>addresses</td>
<td>-2.4</td>
</tr>
<tr>
<td>this</td>
<td>-1.4</td>
</tr>
<tr>
<td>for</td>
<td>-1.0</td>
</tr>
<tr>
<td>the</td>
<td>-1.0</td>
</tr>
<tr>
<td>first</td>
<td>-1.9</td>
</tr>
<tr>
<td>time</td>
<td>-1.6</td>
</tr>
</tbody>
</table>

- Function words cheaper (the: -1.0) than content words (tourism: -2.0)
- Common phrases cheaper (for the first time: -2.3)
  than unusual ones (tourism initiative addresses: -5.9)
Combining Score and Future Cost

- Hypothesis score and future cost estimate are combined for pruning
  - left hypothesis starts with hard part: the tourism initiative
    score: -5.88, future cost: -6.1 → total cost -11.98
  - middle hypothesis starts with easiest part: the first time
    score: -4.11, future cost: -9.3 → total cost -13.41
  - right hypothesis picks easy parts: this for ... time
    score: -4.86, future cost: -9.1 → total cost -13.96
f: María no dio una bofetada a la bruja verde

\[
\begin{array}{ccc}
\end{array}
\]

\[
\begin{align*}
\bar{e}: & \; <s> \; \text{Mary} \\
\bar{c}: & \; <s> \; \text{Maria} \\
\end{align*}
\]

Future costs make these hypotheses comparable.
Other Decoding Algorithms

• A* search

• Greedy hill-climbing

• Using finite state transducers (standard toolkits)
A* Search

- Uses *admissible* future cost heuristic: never overestimates cost
- Translation agenda: create hypothesis with lowest score + heuristic cost
- Done, when complete hypothesis created

Chapter 6: Decoding
Greedy Hill-Climbing

- Create one complete hypothesis with depth-first search (or other means)

- Search for better hypotheses by applying change operators
  - change the translation of a word or phrase
  - combine the translation of two words into a phrase
  - split up the translation of a phrase into two smaller phrase translations
  - move parts of the output into a different position
  - swap parts of the output with the output at a different part of the sentence

- Terminates if no operator application produces a better translation
Decoding algorithm

- Translation as a search problem
- Partial hypothesis keeps track of
  - which source words have been translated (coverage vector)
  - $n$-1 most recent words of English (for LM!)
  - a back pointer list to the previous hypothesis + (e,f) phrase pair used
  - the (partial) translation probability
  - the estimated probability of translating the remaining words
    (precomputed, a function of the coverage vector)

- Start state: no translated words, $E=<s>$, $bp=nil$
- Goal state: all translated words
Decoding algorithm

• $Q[0] \leftarrow \text{Start state}$
• for $i = 0$ to $|f|-1$
  • Keep $b$ best hypotheses at $Q[i]$
  • for each hypothesis $h$ in $Q[i]$
    • for each untranslated span in $h.c$ for which there is a translation $<e,f>$ in the phrase table
      • $h' = h$ extend by $<e,f>$
      • Is there an item in $Q[|h'.c|]$ with $=$ LM state?
        • yes: update the item bp list and probability
        • no: $Q[|h'.c|] \leftarrow h'$
• Find the best hypothesis in $Q[|f|]$, reconstruction translation by following back pointers
f: María no dio una bofetada a la bruja verde

Q[0] Q[1] Q[2] ...
f: Maria no dio una bofetada a la bruja verde

Q[0]  Q[1]  Q[2]  ...
f: María no dio una bofetada a la bruja verde

Q[0]  Q[1]  Q[2]  ...

Mary

Maria

Mary

Maria

Mary

<table>
<thead>
<tr>
<th>ε: &lt;s&gt;</th>
<th>Mary</th>
<th>c: *--------</th>
<th>p: 0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>c: ------</td>
<td>----------</td>
<td>p: 1.0</td>
<td></td>
</tr>
<tr>
<td>c: *--------</td>
<td>*--------</td>
<td>p: 0.3</td>
<td></td>
</tr>
</tbody>
</table>
f: Maria no dio una bofetada a la bruja verde

Q[0] Q[1] Q[2] ...

Mary

Mary

Maria

Maria

Mary did not

did not

< s > Mary

c: *--------
p: 0.9

< s > Maria

c: *--------
p: 0.3

< s > did not

c: **--------
p: 0.3
f: Maria no dio una bofetada a la bruja verde

Q[0] Q[1] Q[2] ...

```
\[e: \text{<s>} \text{Mary} \\
c: \text{*------} \\
p: 0.9\]

\[e: \text{<s>} \text{Maria} \\
c: \text{*------} \\
p: 0.3\]

\[e: \text{did not} \\
c: \text{**------} \\
p: 0.3\]

Maria did not

Mary
did not

Q[0]

Q[1]

Q[2]

...
f: María no dio una bofetada a la bruja verde

Q[0]  Q[1]  Q[2]  ...

Mary

Maria
did not

Mary did not

<e> <s> Mary
c: *-------
p: 0.9

c: *-------
p: 0.3

c: **-------
p: 0.45
f: Maria no dio una bofetada a la bruja verde

Q[0]  Q[1]  Q[2]  ...

Mary

Mari did not

c: **-------  p: 0.45

Mary

c: *--------  p: 0.9

Mary

c: *--------  p: 0.3

not

c: **-------  p: 0.316

not

c: **-------  p: 0.1

not

c: **-------  p: 0.1

slap

c: *****-----  p: 0.316

...
Reordering

- Language express words in different orders
  - bruja verde vs. green witch
- Phrase pairs can “memorize” some of these
- More general: in decoding, “skip ahead”
- Problem:
  - Won’t “easy parts” of the sentence be translated first?
- Solution:
  - Future cost estimate
  - For every coverage vector, estimate what it will cost to translate the remaining untranslated words
  - When pruning, use $p \times \text{future cost}$!
f: Maria no dio una bofetada a la bruja verde

$$Q[0] \quad Q[1] \quad Q[2] \quad ...$$
f: Maria no dio una bofetada a la bruja verde

Q[0]  Q[1]  Q[2]  ...

Mary
Mary
Maria
Maria
Not
Not
Future costs make these hypotheses comparable.
Decoding summary

- Finding the best hypothesis is NP-hard
  - Even with no language model, there are an exponential number of states!
- Solution 1: limit reordering
- Solution 2: (lossy) pruning