Phrase-Based MT: Decoding

February 7, 2013
Phrase Based MT

\[ e^* = \arg \max_{e} p(e | f) \]
\[ = \arg \max_{e} p(f | e) \times p(e) \]
\[ \approx \arg \max_{e} p(f, a | e) \times p(e) \]

- Recipe
  - Segmentation / Alignment model
  - Phrase model
  - Language Model
### 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>
Reordering Model

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

    er    ja nicht

    he    does not

• Pick phrase in input, translate
  – it is allowed to pick words out of sequence reordering
  – phrases may have multiple words: many-to-many translation

Chapter 6: Decoding
Translation Process

• Task: translate this sentence from German into English

er geht ja nicht nach hause

er geht ja nicht

he does not go

• Pick phrase in input, translate
Translation Process

• Task: translate this sentence from German into English

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:

\[
e_{\text{best}} = \arg\max_e \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i) \ d(start_i - end_{i-1} - 1) \ p_{LM}(e)
\]

- Score is computed incrementally for each partial hypothesis

- Components
  
  **Phrase translation**  Picking phrase \( \bar{f}_i \) to be translated as a phrase \( \bar{e}_i \)
  
  \( \rightarrow \) look up score \( \phi(\bar{f}_i|\bar{e}_i) \) from phrase translation table

  **Reordering**  Previous phrase ended in \( end_{i-1} \), current phrase starts at \( start_i \)
  
  \( \rightarrow \) compute \( d(start_i - end_{i-1} - 1) \)

  **Language model**  For \( n \)-gram model, need to keep track of last \( n - 1 \) words
  
  \( \rightarrow \) compute score \( p_{LM}(w_i|w_{i-(n-1)}, ..., w_{i-1}) \) for added words \( w_i \)
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
Decoding: Precompute 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>
</table>

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

- he
- are
- it

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
  → no restriction to hypothesis recombination

- **Language model:** Last \( n - 1 \) words used as history in \( n \)-gram language model
  → 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
  → recombined hypotheses must have that same end position
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
Maria no dio una bofetada a la bruja verde
f: Maria no dio una bofetada a la bruja verde

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

Mary

\[ \begin{array}{l}
\text{e: } \langle s \rangle \text{ Mary} \\
\text{c: } \ast \mbox{-------} \\
p: 0.9 \\
\end{array} \]
f: Maria no dio una bofetada a la bruja verde

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

玛丽 

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

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

Mary

Maria

Mary did not

did not

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

- **Q[0]:**
  - **e:** <s> Mary
  - **c:** *-------
  - **p:** 1.0

- **Q[1]:**
  - **e:** <s> Maria
  - **c:** *-------
  - **p:** 0.3

- **Q[2]:**
  - **e:** did not
  - **c:** **-------
  - **p:** 0.3

**Maria** did not **Mary**
Maria no dio una bofetada a la bruja verde

\[ f: \text{Maria no dio una bofetada a la bruja verde} \]

**Diagram:**

- **$e$:** \(<s>\) Mary
  - **$c$:** *---------
  - **$p$:** 0.9

- **$e$:** \(<s>\) Maria
  - **$c$:** *---------
  - **$p$:** 0.3

- **$e$:** did not
  - **$c$:** ***-------**
  - **$p$:** 0.45

**Tree:**

- Mary
  - **$e$:** \(<s>\)
  - **$c$:** *---------
  - **$p$:** 1.0

- Maria
  - **$e$:** \(<s>\)
  - **$c$:** *---------
  - **$p$:** 0.9

- **$e$:** did not
  - **$c$:** ***-------**
  - **$p$:** 0.45
f: Maria no dio una bofetada a la bruja verde

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

Mary

Maria

Mary did not

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

Mary

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

Mary not

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

Mary did not

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

not

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

not

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

slap

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

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

Tuesday, February 19, 13
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
    \[\rightarrow\] estimate without context
  - reordering model: unknown, ignored for future cost estimation
Cost Estimates from Translation Options

<table>
<thead>
<tr>
<th>Tourism Initiative Addresses</th>
<th>this for the first time</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1.0</td>
<td>-2.0</td>
</tr>
<tr>
<td>-1.5</td>
<td>-2.4</td>
</tr>
<tr>
<td>-4.0</td>
<td>-2.5</td>
</tr>
<tr>
<td>-1.4</td>
<td>-1.0</td>
</tr>
<tr>
<td>-1.0</td>
<td>-1.9</td>
</tr>
<tr>
<td>-1.6</td>
<td>-1.3</td>
</tr>
<tr>
<td>-2.4</td>
<td>-2.2</td>
</tr>
<tr>
<td>-2.7</td>
<td>-2.3</td>
</tr>
<tr>
<td>-2.3</td>
<td>-2.3</td>
</tr>
<tr>
<td>-2.3</td>
<td></td>
</tr>
</tbody>
</table>

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 $\rightarrow$ total cost -11.98
  - middle hypothesis starts with easiest part: the first time
    score: -4.11, future cost: -9.3 $\rightarrow$ total cost -13.41
  - right hypothesis picks easy parts: this for ... time
    score: -4.86, future cost: -9.1 $\rightarrow$ total cost -13.96
f: Maria no dio una bofetada a la bruja verde

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
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
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
- **marginal** benefits, depending on training
- Really hard problem (Sima’an, 1997)
Maria no dio una bofetada a la bruja verde

Adapted from Koehn (2006)
Maria no dio una bofetada a la bruja verde

Adapted from Koehn (2006)
Maria no dio una bofetada a la bruja verde.
Decoding algorithm

• Translation as a search problem
• Partial hypothesis keeps track of
  • which source words have been translated \( (\text{coverage vector}) \)
  • \( n \)-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, \( E=\langle s \rangle \), bp=nil
• **Goal state**: all translated words
Decoding algorithm

- \( Q[0] \leftarrow \) 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
Maria no dio una bofetada a la bruja verde

f

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

é: <s>
c: --------
p: 1.0
f: María no dio una bofetada a la bruja verde

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

Mary

\( e: <s> \) Mary
\( c: *-------- \)
\( p: 0.9 \)

Mary

\( e: <s> \)
\( c: *-------- \)
\( p: 1.0 \)
f: Maria no dio una bofetada a la bruja verde

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

Mary

Maria

Mary

Maria

Maria

no dio una bofetada a la bruja verde

Tuesday, February 19, 13
$$f: \text{Maria no dio una bofetada a la bruja verde}$$

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

---

Mary

Maria

Mary did not

did not

---

Q[0] Q[1] Q[2] ...
f: Maria 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]  ...

Mary did not

Maria

did not

Mary did not

Maria

did not

Mary
f: Maria no dio una bofetada a la bruja verde

Mary: *-------
p: 0.9

Mary did not: **-------
p: 0.1

Maria: *-------
p: 0.3

Mary did not: **-------
p: 0.45

not: Mary not

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

slap: not 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 $p \times \text{future cost}$!
Maria no dio una bofetada a la bruja verde

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

Mary

Maria

\begin{itemize}
  \item $\mathbf{e}$: <s> Mary
  \item $\mathbf{c}$: *--------
  \item $p$: 0.9 \quad fc: 8.6e-9
\end{itemize}

\begin{itemize}
  \item $\mathbf{e}$: <s> Maria
  \item $\mathbf{c}$: *--------
  \item $p$: 0.3 \quad fc: 8.6e-9
\end{itemize}
f: Maria no dio una bofetada a la bruja verde

Maria

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

Not

Mary

Not: <s> Not
p: 0.4  fc: 1.0e-9

c: *--------
p: 0.4  fc: 8.6e-9

Maria

e: <s> Maria
c: *--------
p: 0.9  fc: 8.6e-9

Mary

e: <s> Mary
c: *--------
p: 0.9  fc: 8.6e-9

p: 1.0  fc: 1.5e-9

e: <s> No
Maria no dio una bofetada a la bruja verde

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>0.9</td>
<td>8.6e-9</td>
</tr>
<tr>
<td>Maria</td>
<td>0.3</td>
<td>8.6e-9</td>
</tr>
<tr>
<td>Not</td>
<td>0.4</td>
<td>1.0e-9</td>
</tr>
</tbody>
</table>

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