Phrase-Based Machine Translation

CMSC 723 / LING 723 / INST 725

MARINE CARPUAT
marine@cs.umd.edu
Noisy Channel Model for Machine Translation

\[ \hat{E} = \arg\max_{E \in \text{English}} P(F|E) P(E) \]

- The **noisy channel model** decomposes machine translation into two independent subproblems
  - Language modeling
  - Translation modeling / Alignment
Word Alignment with IBM Models 1, 2

- Probabilistic models with **strong independence assumptions**

- Alignments are hidden variables
  - unlike words which are observed
  - require **unsupervised learning** (EM algorithm)

- Word alignments often used as building blocks for more complex translation models
  - E.g., phrase-based machine translation
PHRASE-BASED MODELS
Phrase-based models

- Most common way to model $P(F|E)$ nowadays (instead of IBM models)

$$P(F|E) = \prod_{i=1}^{I} \phi(\bar{f}_i, \bar{e}_i) d(a_i - b_{i-1})$$

- Probability of two consecutive English phrases being separated by a particular span in French
- Start position of $f_i$
- End position of $f_{(i-1)}$
Phrase alignments are derived from word alignments. This means that the IBM model represents $P(\text{Spanish} \mid \text{English})$. To get high confidence alignment links, intersect the IBM word alignments from both directions.
Phrase alignments are derived from word alignments

Improve recall by adding some links from the union of alignments
Phrase alignments are derived from word alignments

(Maria, Mary), (no, did not),
(slap, dió una bofetada), (verde, green),
(a la, the), (bruja, witch),
(Maria no, Mary did not),
(no dió una bofetada, did not slap),
(dió una bofetada a la, slap the),
(bruja verde, green witch),
(a la bruja verde, the green witch),…
Phrase Translation Probabilities

- Given such phrases we can get the required statistics for the model from

\[
\phi(\bar{f}, \bar{e}) = \frac{\text{count}(\bar{f}, \bar{e})}{\sum_{\bar{f}} \text{count}(\bar{f}, \bar{e})}
\]
Phrase-based Machine Translation

\[ \hat{E} = \operatorname{arg\,max}_{E \in \text{English}} \left( \prod_{i \in S} \phi(f_i, e_i) \right) P(E) \]
DECODING
Decoding for phrase-based MT

• Basic idea
  – search the space of possible English translations in an efficient manner.
  – According to our model

\[
\hat{E} = \arg\max_{E \in \text{English}} P(F|E) \cdot P(E) \\
\prod_{i \in S} \phi(f_i, e_i) d(a_i - b_{i-1}) P(E)
\]
Decoding as Search

• Starting point: null state. No French content covered, no English included.

• We’ll drive the search by
  – Choosing French word/phrases to “cover”,
  – Choosing a way to cover them

• Subsequent choices are pasted left-to-right to previous choices.

• Stop: when all input words are covered.
Decoding

Maria no dio una bofetada a la bruja verde
Decoding

Maria no dio una bofetada a la bruja verde

Mary
Decoding

Maria no dio una bofetada a la bruja verde

Mary did not
Decoding

Maria no dio una bofetada a la bruja verde

Mary Did not slap
Decoding

Maria no dio una bofetada a la bruja verde

Mary Did not slap the
Maria did not slap the green bruja.
Decoding

Maria no dio una bofetada a la bruja verde

Mary Did not slap the green witch
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>

<table>
<thead>
<tr>
<th>Mary</th>
<th>did not</th>
<th>slap</th>
<th>the</th>
<th>green</th>
<th>witch</th>
</tr>
</thead>
</table>

Mary did not slap the green witch.
Decoding

• In practice: we need to incrementally pursue a large number of paths.

• Solution: heuristic search algorithm called “multi-stack beam search”
Space of possible English translations given phrase-based model

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

Mary __ not ___ give ___ a ___ slap ___ to ___ the ___ witch ___ green ___

did not ___ a slap ___ to ___ green witch ___

no ___ slap ___ to the ___

did not give ___

to ___ the ___

green witch ___

to ___

copyright ___

slap _______ the witch _______
Stack decoding: a simplified view

function STACK DECODING(source sentence) returns target sentence

initialize stack with a null hypothesis

loop do
    pop best hypothesis $h$ off of stack
    if $h$ is a complete sentence, return $h$
    for each possible expansion $h'$ of $h$
        assign a score to $h'$
        push $h'$ onto stack

Note: here “stack” = priority queue
Three stages of stack decoding

a) after expanding NULL

b) after expanding “No”

c) after expanding “Mary”
"multi-stack beam search"

Function `BEAM SEARCH STACK DECODE`:

- Initialize `hypothesisStack[0..nf]`.
- Push initial null hypothesis on `hypothesisStack[0]`.

For `i ← 0` to `nf-1`:

For each `hyp` in `hypothesisStack[i]`:

For each `new_hyp` that can be derived from `hyp`:

- `nf_new_hyp ← number of foreign words covered by new_hyp`.
- Add `new_hyp` to `hypothesisStack[nf_new_hyp]`.
- Prune `hypothesisStack[nf_new_hyp]`.

Find best hypothesis `best_hyp` in `hypothesisStack[nf]`.

Return best path that leads to `best_hyp` via backtrace.

Beam-search pruning for each stack: prune high cost states (those “outside the beam”)

One stack per number of French words covered: so that we make apples-to-apples comparisons when pruning.
“multi-stack beam search”

```
function BEAM SEARCH STACK DECODER(source sentence) returns target sentence

initialize hypothesisStack[0..nf]
push initial null hypothesis on hypothesisStack[0]
for i ← 0 to nf-1
    for each hyp in hypothesisStack[i]
        for each new_hyp that can be derived from hyp
            nf_new_hyp ← number of foreign words covered by new_hyp
            add new_hyp to hypothesisStack[nf_new_hyp]
            prune hypothesisStack[nf_new_hyp]
find best hypothesis best_hyp in hypothesisStack[nf]
return best path that leads to best_hyp via backtrace
```
Cost = current cost + future cost

- Future cost = cost of translating remaining words in the French sentence

- Exact future cost = minimum probability of all remaining translations
  - Too expensive to compute!

- Approximation
  - Find sequence of English phrases that has the minimum product of language model and translation model costs
Recombination

• Two distinct hypothesis paths might lead to the same translation hypotheses
  – Same number of source words translated
  – Same output words
  – Different scores

• Recombination
  – Drop worse hypothesis
Recombination

• Two distinct hypothesis paths might lead to hypotheses that are indistinguishable in subsequent search
  – Same number of source words translated
  – Same last 2 output words (assuming 3-gram LM)
  – Different scores

• Recombination
  – Drop worse hypothesis
Complexity Analysis

- Time complexity of decoding as described so far
  \( O(\text{max stack size} \times \text{sentence length}^2) \)
  
  - \( O(\text{max stack size} \times \text{number of ways to expand hyps.} \times \text{sentence length}) \)
Reordering Constraints

Idea: limit reordering to maximum reordering distance

Typically: 5 to 8 words
- Depending on language pair
- Empirically: larger limit hurts translation quality

Resulting complexity: $O(\text{max stack size } \times \text{ sentence length})$
- because we limit reordering distance, so that only a constant number of hypothesis expansions are considered
RECAP
Noisy Channel Model for Machine Translation

\[ \hat{E} = \arg\max_{E \in \text{English}} P(F|E) \]

- The **noisy channel model** decomposes machine translation into two independent subproblems
  - Language modeling
  - Translation modeling / Alignment
Phrase-Based Machine Translation

• Phrase-translation dictionary
Phrase-Based Machine Translation

• A simple model of translation
  – Phrase translation dictionary ("phrase-table")
    • Extract all phrase pairs consistent with given alignment
    • Use relative frequency estimates for translation probabilities
  – Distortion model
    • Allows for reorderings
Decoding in Phrase-Based Machine Translation

• Approach: Heuristic search
• With several strategies to reduce the search space
  – Pruning
  – Recombination
  – Reordering constraints
What are the pros and cons of phrase-based vs. neural MT?