Statistical Machine Translation: Decoding

Alexander Fraser
(Many slides from Aleš Tamchyna, Philipp Koehn)

fraser@cis.uni-muenchen.de

SS 2023
Outline

- Which features are used in PBMT?
- How to compute the score of a translation?
- Search for the best translation: decoding.
  - Overview of the translation process.
  - Making decoding tractable: beam search.
- Other decoding algorithms.
Log-Linear Model

We know how to score a full translation hypothesis:

$$ P(e, a|f) \propto \exp \sum_{i} \lambda_i f_i(e, a, f) $$

$\lambda_i$ ... feature weights

$f_i$ ... feature functions
Log-Linear Model: Features

Typical baseline feature set for PBMT:

- Phrase translation probability, both direct and inverse:
  - $P_{TM}(e|f)$
  - $P_{TM_{inv}}(f|e)$

- Lexical translation probability (direct and inverse):
  - $P_{lex}(e|f)$
  - $P_{lex_{inv}}(f|e)$

- Language model probability:
  - $P_{LM}(e)$

- Phrase penalty.
- Word penalty.
- Distortion penalty.
Lexical Weights ($P_{lex}$)

The problem: many extracted phrases are rare.
(Esp. long phrases might only be seen once in the parallel corpus.)
Lexical Weights ($P_{lex}$)

The problem: many extracted phrases are rare. (Esp. long phrases might only be seen once in the parallel corpus.)

\[
P("\text{modrý autobus přistál na Marsu}\) | "a blue bus lands on Mars") = 1
\]
\[
P("a blue bus lands on Mars" | "\text{modrý autobus přistál na Marsu}\) = 1
\]

Is that a reliable probability estimate?
Lexical Weights ($P_{lex}$)

The problem: many extracted phrases are rare. (Esp. long phrases might only be seen once in the parallel corpus.)

\[
P(\text{"carried - over"} | \text{"; zkreslení"}) = 1 \\
P(\text{"; zkreslení"} | \text{"; distortion carried - over"}) = 1
\]

Data from the “wild” are noisy. Word alignment contains errors. “carried - over” is wrong. This is a real phrase pair from a very good English-Czech SMT system. Both $P_{TM}(e|f)$ and $P_{TM_{inv}}(f|e)$ say that this is a perfect translation.
Lexical Weights ($P_{lex}$)

Decompose the phrase pair into word pairs. Look at the word-level translation probabilities.
Lexical Weights ($P_{\text{lex}}$)

Decompose the phrase pair into word pairs. Look at the word-level translation probabilities. Several possible definitions, e.g.:
Lexical Weights ($P_{\text{lex}}$)

Decompose the phrase pair into word pairs. Look at the word-level translation probabilities.
Several possible definitions, e.g.:

$$P_{\text{lex}}(e|f, a) = \prod_{j=1}^{l_e} \frac{1}{|i|(i,j) \in a} \sum_{\forall (i,j) \in a} w(e_j, f_i)$$
Lexical Weights \((P_{\text{lex}})\)

Decompose the phrase pair into word pairs. Look at the word-level translation probabilities.
Several possible definitions, e.g.:

\[
P_{\text{lex}}(e|f, a) = \prod_{j=1}^{l_e} \frac{1}{|\{i|(i,j) \in a|\sum_{(i,j) \in a} w(e_j, f_i)
\]

\begin{align*}
\text{psací} & \quad 0.1 \quad \text{a} \\
\text{stroj} & \quad 0.2 \quad \text{typewriter}
\end{align*}
Lexical Weights ($P_{\text{lex}}$)

Decompose the phrase pair into word pairs. Look at the word-level translation probabilities.
Several possible definitions, e.g.:

\[
P_{\text{lex}}(e|f, a) = \prod_{j=1}^{l_e} \frac{1}{|i|(i,j) \in a} \sum_{\forall (i,j) \in a} w(e_j, f_i)
\]

\[
\begin{array}{c}
\text{psací} \quad 0.1 \quad a \\
\quad 0.3 \\
\text{stroj} \quad 0.2 \quad \text{typewriter}
\end{array}
\]

\[
P_{\text{lex}}(\text{“a typewriter”} | \text{“psací stroj”}) = \left[\frac{1}{1} \cdot 0.1\right] \cdot \left[\frac{1}{2} \cdot (0.3 + 0.2)\right] = 0.025
\]
Word Penalty

Not all languages use the same number of words on average.

vidím problém ||| I can see a problem
Word Penalty

Not all languages use the same number of words on average.

vidím problém ||| I can see a problem

- We want to control how many words are generated.
Word Penalty

Not all languages use the same number of words on average.

vidím problém ||| I can see a problem

▶ We want to control how many words are generated.
▶ Word penalty simply adds 1 for each produced word in the translation.
Word Penalty

Not all languages use the same number of words on average.

vidím problém ||| I can see a problem

- We want to control how many words are generated.
- Word penalty simply adds 1 for each produced word in the translation.
- Depending on the λ for word penalty, we will either generate shorter or longer outputs.
Word Penalty

Not all languages use the same number of words on average.

vidím problém ||| I can see a problem

▶ We want to control how many words are generated.
▶ Word penalty simply adds 1 for each produced word in the translation.
▶ Depending on the $\lambda$ for word penalty, we will either generate shorter or longer outputs.

$$\hat{e} = \arg \max_{e,a} \sum_i \lambda_i f_i(e, a, f)$$
Phrase Penalty

- Add 1 for each produced phrase in the translation.
Phrase Penalty

- Add 1 for each produced *phrase* in the translation.
- Varying the $\lambda$ for phrase penalty can lead to more literal (word-by-word) translations (made from a lot of short phrases) or to more idiomatic outputs (use fewer, longer phrases – if available).
Distortion Penalty

- The simplest way to capture **phrase reordering**.
- Can be sufficient for some language pairs
- Several possible definitions!
- Definition I tend to use:
  - Distance between the end of the previous phrase (on the source side) and the beginning of the current phrase.
How to Score a Translation?

score(e|f) = 0
How to Score a Translation?

\[
score(e|f) = \lambda_{TM} \cdot \log P_{TM}("he"|"er") + \lambda_{TM_{inv}} \cdot \log P_{TM_{inv}}("er"|"he") + \lambda_{lex} \cdot \log P_{lex}("he"|"er") + \lambda_{lex_{inv}} \cdot \log P_{lex_{inv}}("er"|"he") + \lambda_{D} \cdot 0 + \lambda_{WP} \cdot 1 + \lambda_{PP} \cdot 1 + \lambda_{LM} \cdot \log P_{LM}("he"|"<S>"')
\]
How to Score a Translation?

\[
\text{score}(e|f) + = \lambda_{TM} \cdot \log P_{TM}("\text{does not}"|"ja nicht") \\
+ \lambda_{TM_{inv}} \cdot \log P_{TM_{inv}}("ja nicht"|"does not") \\
+ \lambda_{lex} \cdot \log P_{lex}("does not"|"ja nicht") \\
+ \lambda_{lex_{inv}} \cdot \log P_{lex_{inv}}("ja nicht"|"does not") \\
+ \lambda_{D} \cdot 1 \\
+ \lambda_{WP} \cdot 2 \\
+ \lambda_{PP} \cdot 1 \\
+ \lambda_{LM} \cdot \log P_{LM}("does not"|"<S>he")
\]
How to Score a Translation?

$$\text{score}(e|f) = \lambda_{TM} \cdot \log P_{TM}("\text{go}" | "\text{geht}")$$
$$+ \lambda_{TM_{inv}} \cdot \log P_{TM_{inv}}("\text{geht}" | "\text{go}")$$
$$+ \lambda_{lex} \cdot \log P_{lex}("\text{go}" | "\text{geht}")$$
$$+ \lambda_{lex_{inv}} \cdot \log P_{lex_{inv}}("\text{geht}" | "\text{go}")$$
$$+ \lambda_{D} \cdot 3$$
$$+ \lambda_{WP} \cdot 1$$
$$+ \lambda_{PP} \cdot 1$$
$$+ \lambda_{LM} \cdot \log P_{LM}("\text{go}" | "\text{does not}")$$
How to Score a Translation?

\[
score(e|f) + = \ldots
\]
How to Score a Translation?

\[ \textit{score}(e|f)^+ = \ldots \]
Decoding

• We have a mathematical model for translation

\[ p(e|f) \]

• Task of decoding: find the translation \( e_{\text{best}} \) with highest probability

\[ e_{\text{best}} = \arg\max_e p(e|f) \]

• Two types of error
  – the most probable translation is bad \( \rightarrow \) fix the model
  – search does not find the most probably translation \( \rightarrow \) fix the search

• Decoding is evaluated by search error, not quality of translations
  (although these are often correlated)
Translation Process

- Task: translate this sentence from German into English

  er geht ja nicht nach hause

Slide by Philipp Koehn (Statistical Machine Translation, Chapter 6)
Translation Process

- Task: translate this sentence from German into English

\[ \text{er geht ja nicht nach hause} \]

- Pick phrase in input, translate

Slide by Philipp Koehn (Statistical Machine Translation, Chapter 6)
Translation Process

- Task: translate this sentence from German into English

\[ \text{er geht ja nicht nach hause} \]

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

\begin{tabular}{c c c c c}
<table>
<thead>
<tr>
<th>English</th>
<th>German</th>
</tr>
</thead>
<tbody>
<tr>
<td>he</td>
<td>er</td>
</tr>
<tr>
<td>does not</td>
<td>ja nicht</td>
</tr>
<tr>
<td>nach</td>
<td>hause</td>
</tr>
</tbody>
</table>
\end{tabular}

Slide by Philipp Koehn (Statistical Machine Translation, Chapter 6)
Translation Process

- Task: translate this sentence from German into English

\[ \text{er geht ja nicht nach hause} \]

- Pick phrase in input, translate

Slide by Philipp Koehn (Statistical Machine Translation, Chapter 6)
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

Slide by Philipp Koehn (Statistical Machine Translation, Chapter 6)
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

Slide by Philipp Koehn (Statistical Machine Translation, Chapter 6)
Decoding: Precompute Translation Options

consult phrase translation table for all input phrases

Slide by Philipp Koehn (Statistical Machine Translation, Chapter 6)
Decoding: Start with Initial Hypothesis

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

pick any translation option, create new hypothesis

Slide by Philipp Koehn (Statistical Machine Translation, Chapter 6)
Decoding: Hypothesis Expansion

create hypotheses for all other translation options
Decoding: Hypothesis Expansion

also create hypotheses from created partial hypothesis

Slide by Philipp Koehn (Statistical Machine Translation, Chapter 6)
Decoding: Find Best Path

backtrack from highest scoring complete hypothesis

Slide by Philipp Koehn (Statistical Machine Translation, Chapter 6)
Computational Complexity

- The suggested process creates exponential number of hypothesis
- Machine translation decoding is NP-complete
- Reduction of search space:
  - recombination (risk-free)
  - pruning (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

Slide by Philipp Koehn (Statistical Machine Translation, Chapter 6)
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

- Other feature function may introduce additional restrictions

---

Slide by Philipp Koehn (Statistical Machine Translation, Chapter 6)
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

Slide by Philipp Koehn (Statistical Machine Translation, Chapter 6)
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
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

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

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

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

- 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. The worse hypothesis has a better score.

Slide by Philipp Koehn (Statistical Machine Translation, Chapter 6)
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

cost of cheapest translation options for each input span (log-probabilities)

Slide by Philipp Koehn (Statistical Machine Translation, Chapter 6)
## Cost Estimates for all Spans

- Compute cost estimate for all contiguous spans by combining cheapest options first future cost estimate for \( n \) words (from first)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1.0</td>
<td>2.0</td>
<td>3.0</td>
<td>4.0</td>
<td>5.0</td>
<td>6.0</td>
<td>7.0</td>
<td>8.0</td>
<td>9.0</td>
</tr>
<tr>
<td>the</td>
<td></td>
<td>-1.0</td>
<td>-3.0</td>
<td>-4.5</td>
<td>-6.9</td>
<td>-8.3</td>
<td>-9.3</td>
<td>-9.6</td>
<td>-10.6</td>
<td>-10.6</td>
</tr>
<tr>
<td>tourism</td>
<td></td>
<td>-2.0</td>
<td>-3.5</td>
<td>-5.9</td>
<td>-7.3</td>
<td>-8.3</td>
<td>-8.6</td>
<td>-9.6</td>
<td>-9.6</td>
<td>-9.6</td>
</tr>
<tr>
<td>initiative</td>
<td></td>
<td>-1.5</td>
<td>-3.9</td>
<td>-5.3</td>
<td>-6.3</td>
<td>-6.6</td>
<td>-7.6</td>
<td>-7.6</td>
<td>-7.6</td>
<td>-7.6</td>
</tr>
<tr>
<td>addresses</td>
<td></td>
<td>-2.4</td>
<td>-3.8</td>
<td>-4.8</td>
<td>-5.1</td>
<td>-6.1</td>
<td>-6.1</td>
<td>-6.1</td>
<td>-6.1</td>
<td>-6.1</td>
</tr>
<tr>
<td>this</td>
<td></td>
<td>-1.4</td>
<td>-2.4</td>
<td>-2.7</td>
<td>-3.7</td>
<td>-3.7</td>
<td>-3.7</td>
<td>-3.7</td>
<td>-3.7</td>
<td>-3.7</td>
</tr>
<tr>
<td>for</td>
<td></td>
<td>-1.0</td>
<td>-1.3</td>
<td>-2.3</td>
<td>-2.3</td>
<td>-2.3</td>
<td>-2.3</td>
<td>-2.3</td>
<td>-2.3</td>
<td>-2.3</td>
</tr>
<tr>
<td>the</td>
<td></td>
<td>-1.0</td>
<td>-2.2</td>
<td>-2.3</td>
<td>-2.3</td>
<td>-2.3</td>
<td>-2.3</td>
<td>-2.3</td>
<td>-2.3</td>
<td>-2.3</td>
</tr>
<tr>
<td>first</td>
<td></td>
<td>-1.9</td>
<td>-2.4</td>
<td>-2.4</td>
<td>-2.4</td>
<td>-2.4</td>
<td>-2.4</td>
<td>-2.4</td>
<td>-2.4</td>
<td>-2.4</td>
</tr>
<tr>
<td>time</td>
<td></td>
<td>-1.6</td>
<td>-1.6</td>
<td>-1.6</td>
<td>-1.6</td>
<td>-1.6</td>
<td>-1.6</td>
<td>-1.6</td>
<td>-1.6</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)

---

Slide by Philipp Koehn (Statistical Machine Translation, Chapter 6)
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

Slide by Philipp Koehn (Statistical Machine Translation, Chapter 6)
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

Slide by Philipp Koehn (Statistical Machine Translation, Chapter 6)
Finite-state transducers

- It is also possible to output a pruned search graph to an external finite-state transducer package.
- This then carries out the search, but I will omit the details of this.
- Allows efficient search of this (pruned) graph
- Can be useful for rescoring the hypotheses using models that are difficult to implement directly in, e.g., Moses.
Summary

- Log-linear model: standard features in PBMT.
- Computing the score of a translation.
- Overview of the translation process.
- Beam search algorithm.
  - Hypothesis recombination.
  - Pruning.
  - Limiting distortion.
  - Future cost.
- Other decoding algorithms.
Questions?
Thank you for your attention.