Phrase-based Statistical Machine Translation

Ulrich Germann, University of Edinburgh
September 16, 2016

An Obituary?

So long and thanks for all the fish!

Past, Present, and Now What?

Background: Noisy Channel Model

Bayes' Rule

$$p(e \mid f) = \frac{p(f \mid e) \cdot p(e)}{p(f)}$$

Background: Noisy Channel Model

Optimization Criterion

$$\arg\max_{e} \; p\left(e \,|\, f\right) = \arg\max_{e} \; \frac{p\left(f \,|\, e\right) \cdot p\left(e\right)}{p\left(f\right)}$$

Background: Noisy Channel Model

Optimization Criterion

$$\arg\max_{e} \; p(e \mid f) = \arg\max_{e} \; p(f \mid e) \cdot p(e)$$

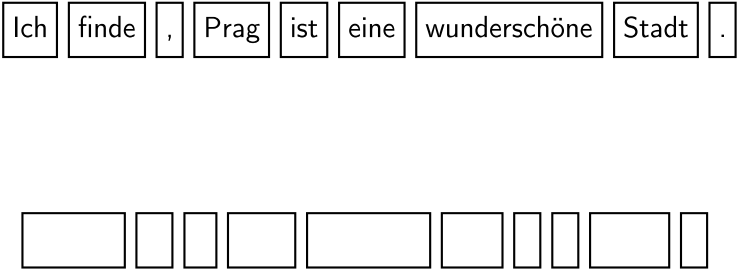
Nowadays: Linear Model

$$\arg\max_{e} score\,(e \mid f) = \sum_{i=1}^{n} w_i \cdot h_{e;i}$$

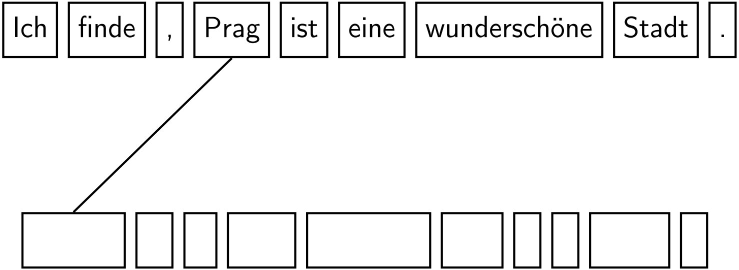
A Simple Word-to-word Translation Model

Ich	finde	,	Prag	ist	eine	wunderschöne	Stadt	.

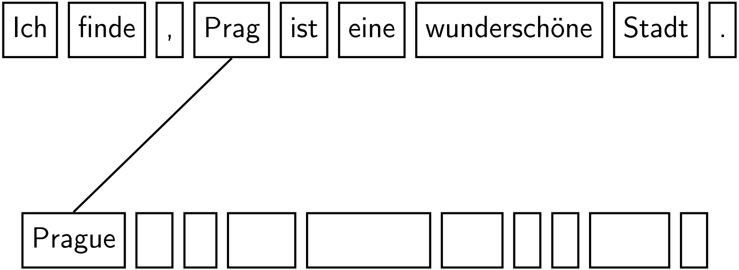
A Simple Word-to-word Translation Model



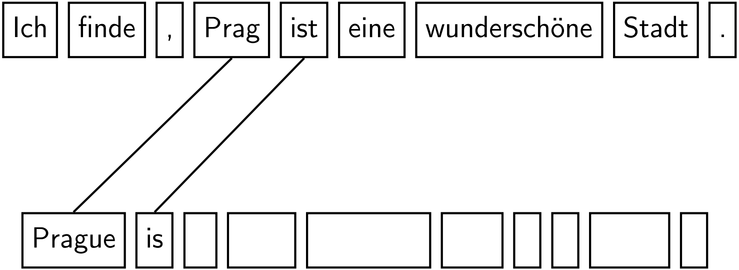
A Simple Word-to-word Translation Model



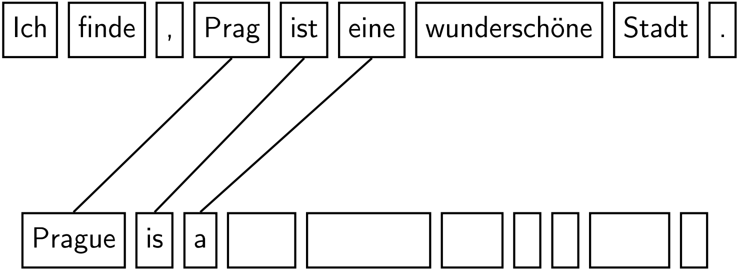
A Simple Word-to-word Translation Model



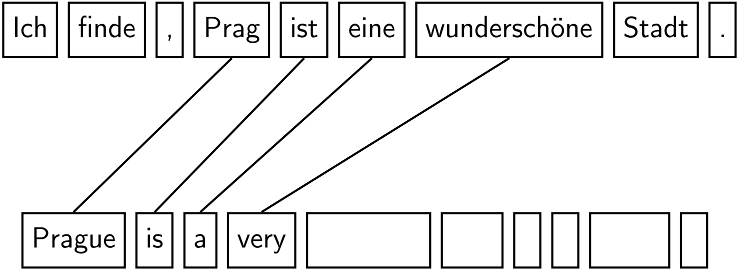
A Simple Word-to-word Translation Model



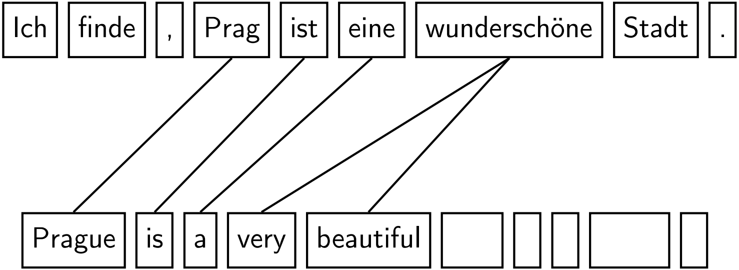
A Simple Word-to-word Translation Model



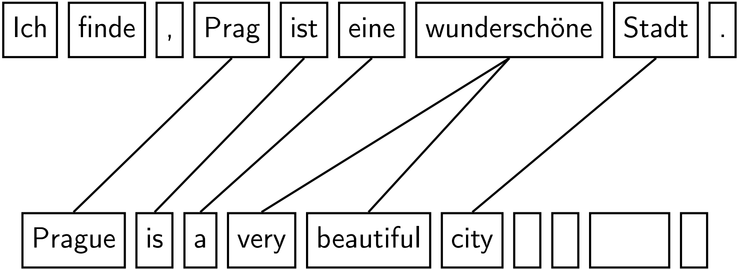
A Simple Word-to-word Translation Model



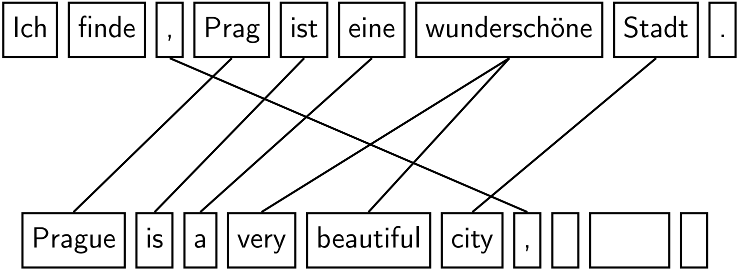
A Simple Word-to-word Translation Model



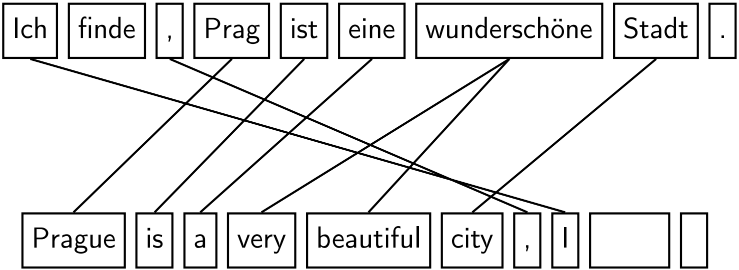
A Simple Word-to-word Translation Model



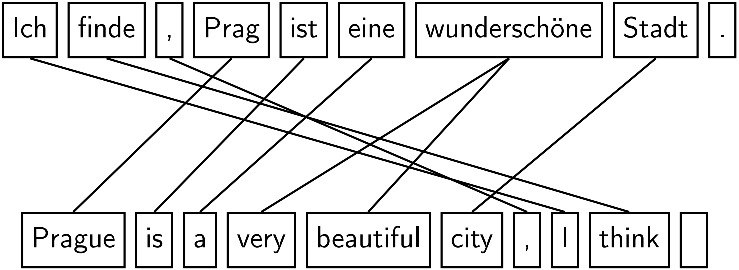
A Simple Word-to-word Translation Model



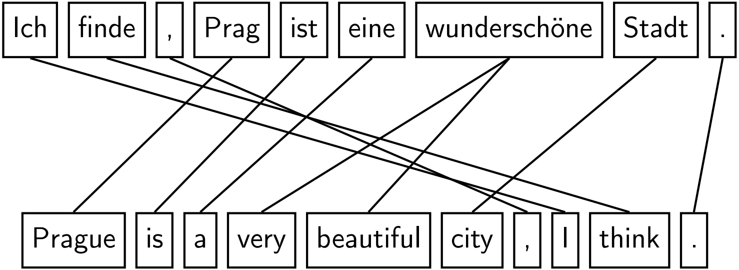
A Simple Word-to-word Translation Model



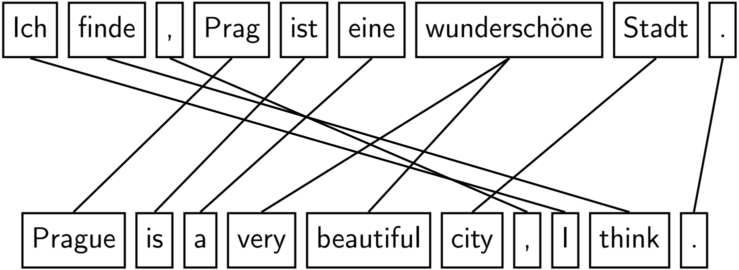
A Simple Word-to-word Translation Model



A Simple Word-to-word Translation Model

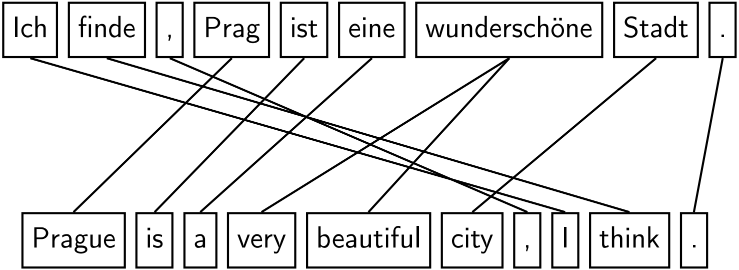


A Simple Word-to-word Translation Model



(sentence length model)

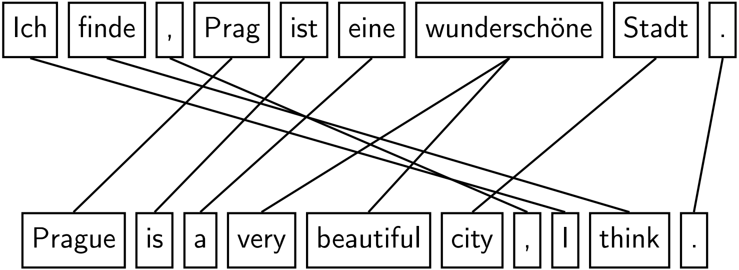
A Simple Word-to-word Translation Model



(sentence length model)

(distortion probability)

A Simple Word-to-word Translation Model

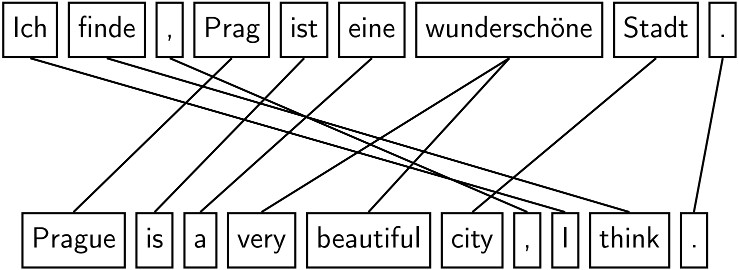


(sentence length model)

(distortion probability)

(word) translation probability

A Simple Word-to-word Translation Model



(sentence length model)

(distortion probability)

(word) translation probability

(language model probability)

Model Estimation (for now, only translation probabilities)

Word translation probabilities are easy to estimate from word alignment links:

$$t\,(e\,|\,f) = \frac{count\,(f \rightarrow e)}{\sum_{\hat{e}} count\,(f \rightarrow \hat{e})}$$

Model Estimation (for now, only translation probabilities)

Word translation probabilities are easy to estimate from word alignment links:

$$t\left(e \mid f\right) = \sum_{\vec{e},\vec{f} \in Corpus} \sum_{i=1; e_i = e}^{|\vec{e}|} \sum_{k=0; f_k = f}^{|\vec{k}|} p\left(a_i = k \mid \vec{e}, \vec{f}\right)$$

Model Estimation (for now, only translation probabilities)

Word translation probabilities are easy to estimate from word alignment links:

$$t\left(e \mid f\right) = \sum_{\vec{e},\vec{f} \in Corpus} \sum_{i=1; e_i = e}^{|\vec{e}|} \sum_{k=0; f_k = f}^{|\vec{k}|} p\left(a_i = k \mid \vec{e}, \vec{f}\right)$$

Word alignment links can be inferred from word translation probabilities:

$$p\left(a_i = k\right) = \frac{t\left(e_i \mid f_k\right)}{\sum_{\hat{k}} t\left(e_i \mid f_{\hat{k}}\right)}$$

Model Estimation (for now, only translation probabilities)

Word translation probabilities are easy to estimate from word alignment links:

$$t\left(e \mid f\right) = \sum_{\vec{e},\vec{f} \in Corpus} \sum_{i=1; e_i = e}^{|\vec{e}|} \sum_{k=0; f_k = f}^{|\vec{k}|} p\left(a_i = k \mid \vec{e}, \vec{f}\right)$$

Word alignment links can be inferred from word translation probabilities:

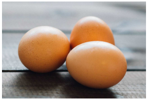
$$p\left(a_i = k\right) = \frac{t\left(e_i \mid f_k\right)}{\sum_{\hat{k}} t\left(e_i \mid f_{\hat{k}}\right)}$$

Model Estimation (for now, only translation probabilities)

Word translation probabilities are easy to estimate from word alignment links:

$$t(e \mid f) = \sum_{\vec{e},\vec{f} \in Corpus} \sum_{i=1; e_i = e}^{|\vec{e}|} \sum_{k=0; f_k = f}^{|\vec{k}|} p(a_i = k \mid \vec{e}, \vec{f})$$

?



Word alignment links can be inferred from word translation probabilities:

$$p(a_i = k) = \frac{t(e_i \mid f_k)}{\sum_{\hat{k}} t(e_i \mid f_{\hat{k}})}$$

The Expectation Maximization Algorithm

E-Step instead of counting, guess (partial counts for each event, based on probability)

M-Step update probability estimates based on these partial counts

- repeat until likelihood of training data stops increasing (convergence)

IBM Models 1 and 2

Model 1

- uniform sentence length probability
- uniform distortion probabilty
- $p(\vec{e} \mid \vec{f}) = \epsilon \sum_{\vec{a}} \prod_{i=1}^{|\vec{e}|} t(e_i \mid f_{a_i})$

Model 2

- uniform sentence length probability
- **distortion probability** based on absolute positions within the sentence $d(k \mid i)$.
- word translation probabilities as in Model 1

IBM Models 3 to 5

New generative story

- *for each source word* f_k *pick a fertility* n_k *with probability* $p(n_k \mid f)$.
- *copy* f_k n_k *times*
- *translate each copy according to* $t(e_{k:j} \mid f_k)$
- *place translations into target sentence*

- **Model 3:** distortion probabilities based on absolute positions
- **Model 4:** distortion probabilities based on positions relative to the target positions of previously placed word(s)
- **Model 5:** eliminates a deficiency of Models 3 and 4; not used in practice.

Nota bene

From Model 3, on, individual word translations are not independent of one another any more (because of fertility, relative distortions)!

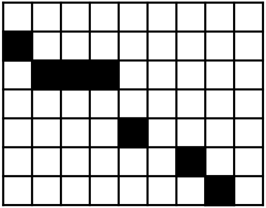
- full marginalization $\sum_{\vec{a}} p(\vec{e}, \vec{a} \mid \vec{f})$ is too expensive
- initialize *Viterbi Alignment* alignment from lower Model, consider only neighboring alignments during training

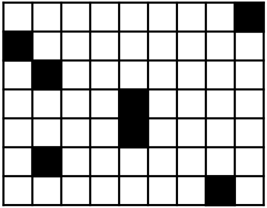
Hidden Markov Models for Alignment

- source words f are hidden states
- emit target words according to $t(e \mid f)$
- distortion modeled via transition probabilities between states of Hidden Markov Model
- replaces Model 2 in the standard Giza++ setup

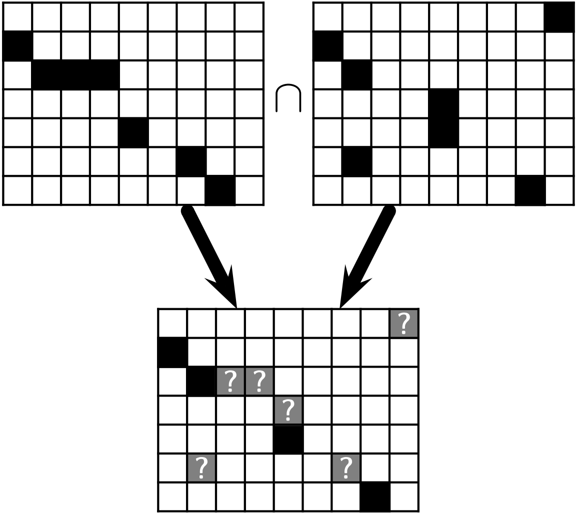
Alignment Symmetrisation

grow-diag + final-and





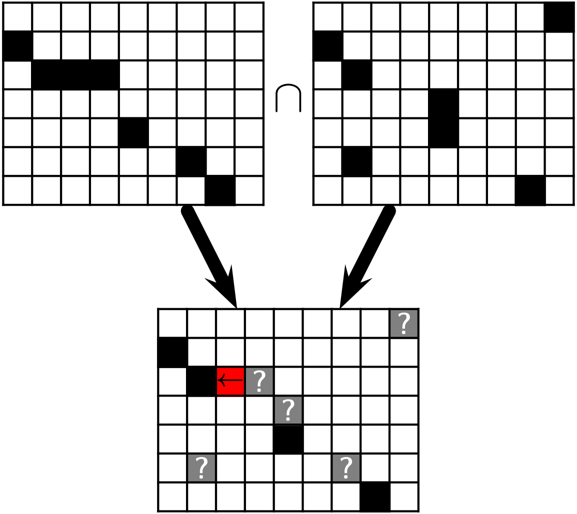
Step 1: Intersect the two alignments:



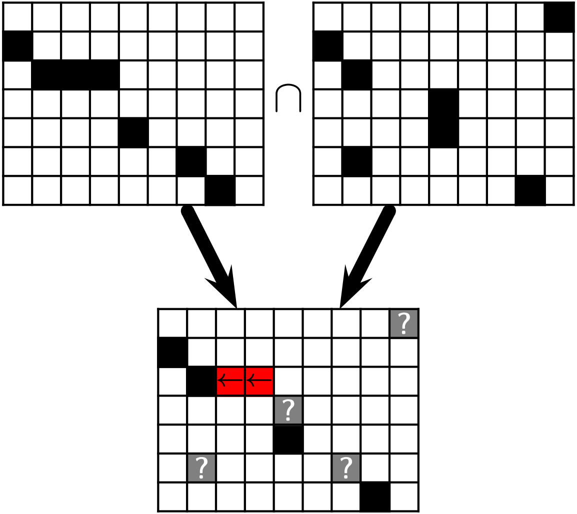
Alignment Symmetrisation

grow-diag + final-and

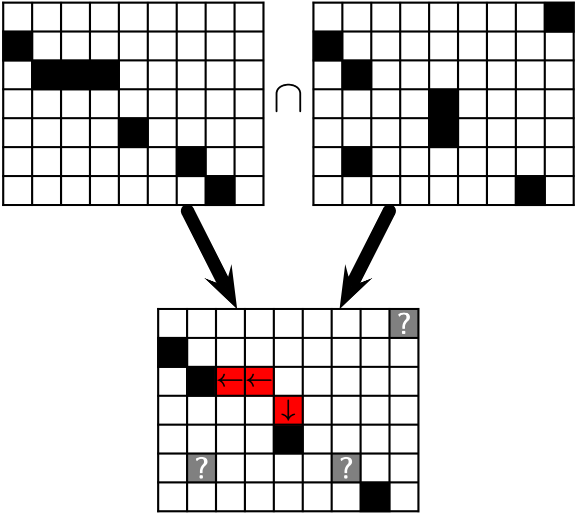
Step 1: Intersect the two alignments:



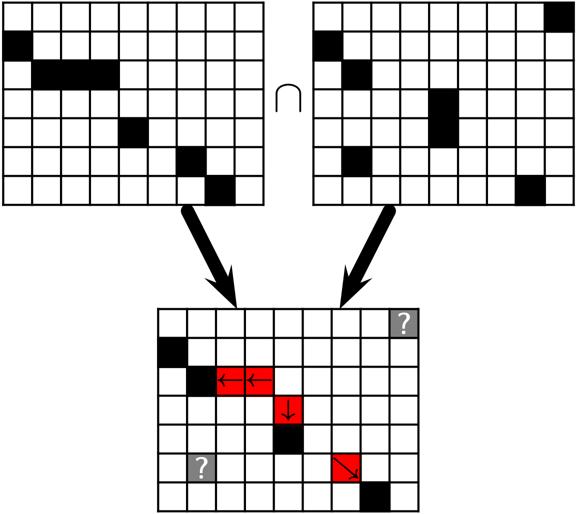
Step 1: Intersect the two alignments:



Step 1: Intersect the two alignments:



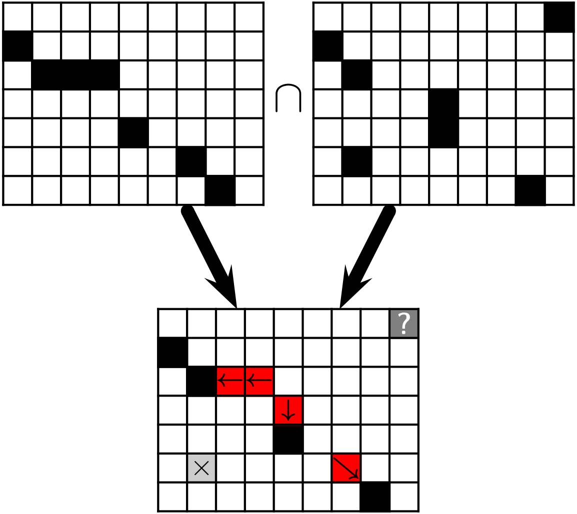
Step 1: Intersect the two alignments:



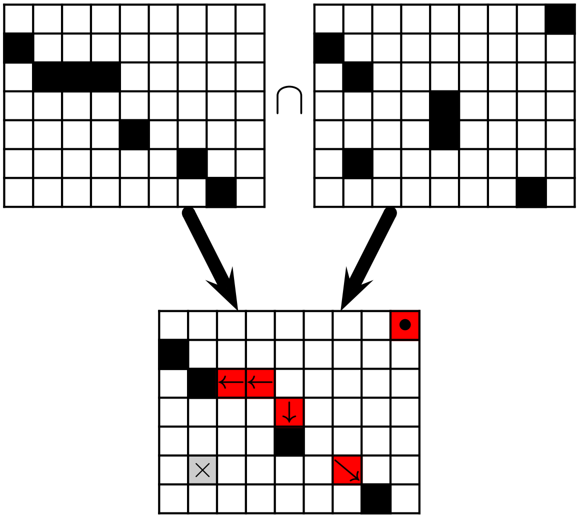
Alignment Symmetrisation

grow-diag + final-and

Step 1: Intersect the two alignments:

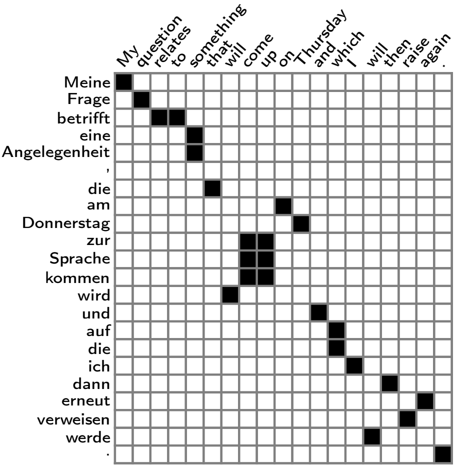


Step 1: Intersect the two alignments:



Building Phrase Tables

phrase extraction

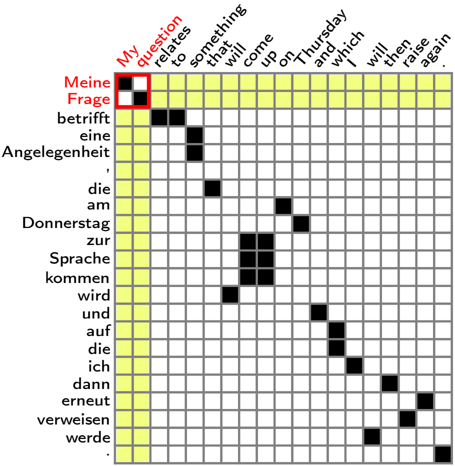


Phrase Table

meine	⇔	my
Frage	⇔	question

Building Phrase Tables

phrase extraction

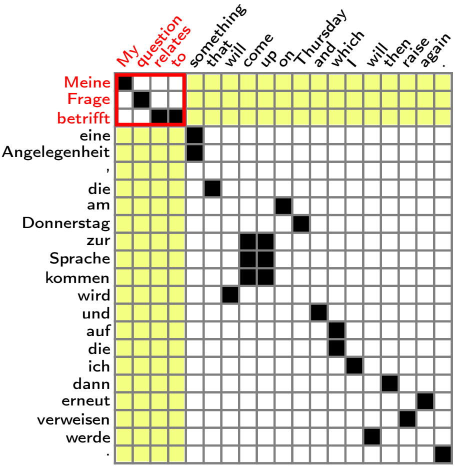


Phrase Table

meine	⇔	my
meine Frage	⇔	my question

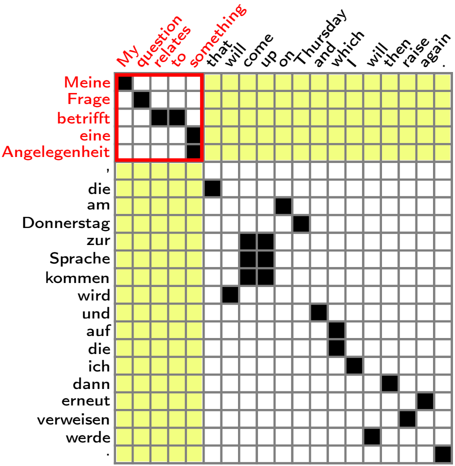
Building Phrase Tables

phrase extraction



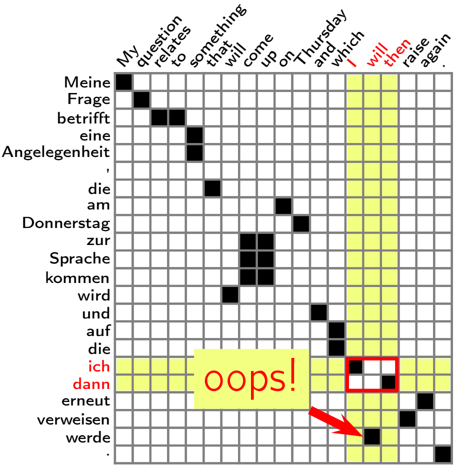
Phrase Table

meine	⇔	my
meine Frage	⇔	my question
meine Frage betrifft	⇔	my question relates to



Phrase Table

meine	⇔	my
meine Frage	⇔	my question
meine Frage betrifft	⇔	my question relates to
meine Frage betrifft eine Angelegenheit	⇔	my question relates to something



Phrase Table

meine	⇔	my
meine Frage	⇔	my question
meine Frage betrifft	⇔	my question relates to
meine Frage betrifft eine Angelegenheit	⇔	my question relates to something
⋮		⋮
Frage	⇔	question
Frage betrifft	⇔	question relates to
⋮		⋮
~~ich dann~~	⇔	~~I will then~~
⋮		⋮

Scoring Phrase Table Entries

weighted linear combination of features:

$$P_{TM}(t \mid s) = exp\left(\sum_j \alpha_j f_j(s, t)\right)$$

Scoring Phrase Table Entries: Feature Functions

- log of smoothed **forward** cond. prob.:
  $$smooth\left(\frac{count\,(target\ phrase)}{count\,(source\ phrase)}\right)$$
- log of smoothed **backward** cond. prob.:
  $$smooth\left(\frac{count\,(source\ phrase)}{count\,(target\ phrase)}\right)$$
- "lexically smoothed" (Zens&Ney) **forward** probability
  $$\sum_t \log P\,(t \mid source\ phrase[, alignment])$$
- "lexically smoothed" **backward** probability
  $$\sum_s \log P\,(s \mid target\ phrase[, alignment])$$
- length of target phrase ("word penalty")
- 1 ("phrase penalty")

Scoring Translation Hypotheses

Log-linear combination of:

Translation Model	assesses the quality of phrase-level translations.
Distortion Model	evaluates jumps between source phrases.
Language Model	evaluates the fluency of the translation hypothesis

$$P\,(\mathit{translation} \mid \mathit{source}) = \exp\left(\begin{array}{cl} & \alpha_{TM} \log P_{TM}(\mathit{translation} \mid \mathit{source}) \\ + & \alpha_{DM} \log P_{DM}(\mathit{translation} \mid \mathit{source}) \\ + & \alpha_{LM} \log P_{LM}(\mathit{translation} \mid \mathit{source}) \end{array}\right)$$

Scoring Translation Hypotheses

November inflation rates were higher than expected in the 13 countries of the eurozone .

Scoring Translation Hypotheses

November [inflation rates] were higher than expected in the 13 countries of the euro. . .

Teuerungsraten

⟨s⟩ Inflationsraten

. . .

$$p(t \mid i, \mathcal{M}_{tr}, \mathcal{M}_{lm}, \mathcal{M}_{d}) =$$

$$\exp\left(\alpha_{tr} \cdot \log p_{tr}(\text{Inflationsraten} \mid \text{inflation rates}) + \alpha_{lm} \cdot \log p_{lm}(\text{Inflationsraten} \mid \langle s \rangle) \right)$$

Scoring Translation Hypotheses

November | inflation rates | were higher than expected in the 13 countries of the euro. . .

⟨s⟩ Inflationsraten

$$p(t \mid i, \mathcal{M}_{tr}, \mathcal{M}_{lm}, \mathcal{M}_{d}) =$$

$$\exp \left(\begin{array}{l} \alpha_{tr} \cdot \log p_{tr}(\text{Inflationsraten} \mid \text{inflation rates}) \quad + \alpha_{lm} \cdot \log p_{lm}(\text{Inflationsraten} \mid \langle s \rangle) \\ + \alpha_{d} \cdot \log p_{d}(-2) \end{array} \right)$$

Scoring Translation Hypotheses

November inflation rates were higher than expected in the 13 countries of the euro. . .

⟨s⟩ Inflationsraten im November

$$p(t \mid i, \mathcal{M}_{tr}, \mathcal{M}_{lm}, \mathcal{M}_{d}) =$$

$$\exp\left(\begin{array}{ll} \alpha_{tr} \cdot \log p_{tr}(\text{Inflationsraten} \mid \text{inflation rates}) & + \alpha_{lm} \cdot \log p_{lm}(\text{Inflationsraten} \mid \langle s \rangle) \\ + \alpha_{d} \cdot \log p_{d}(-2) & \\ + \alpha_{tr} \cdot \log p_{tr}(\text{im November} \mid \text{November}) & + \alpha_{lm} \cdot \log p_{lm}(\text{im} \mid \ldots \text{ Inflationsraten}) \\ & + \alpha_{lm} \cdot \log p_{lm}(\text{November} \mid \ldots \text{ in}) \end{array} \right.$$

Scoring Translation Hypotheses

November inflation rates **were higher than** expected in the 13 countries of the euro. . .

⟨s⟩ Inflationsraten im November waren höher als

$$p(t \mid i, \mathcal{M}_{tr}, \mathcal{M}_{lm}, \mathcal{M}_{d}) =$$

$$\exp \left(\begin{array}{ll} \alpha_{tr} \cdot \log p_{tr}(\text{Inflationsraten} \mid \text{inflation rates}) & + \alpha_{lm} \cdot \log p_{lm}(\text{Inflationsraten} \mid \langle s \rangle) \\ + \alpha_{d} \cdot \log p_{d}(-2) & \\ + \alpha_{tr} \cdot \log p_{tr}(\text{im November} \mid \text{November}) & + \alpha_{lm} \cdot \log p_{lm}(\text{im} \mid \ldots \text{ Inflationsraten}) \\ & + \alpha_{lm} \cdot \log p_{lm}(\text{November} \mid \ldots \text{ in}) \\ + \alpha_{d} \cdot \log p_{d}(+3) & \\ + \alpha_{tr} \cdot \log p_{tr}(\text{waren} \ldots \text{ als} \mid \text{were} \ldots \text{ than}) & + \alpha_{lm} \cdot \log p_{lm}(\text{waren} \mid \ldots \text{ November}) \\ & + \alpha_{lm} \cdot \log p_{lm}(\text{höher} \mid \ldots \text{ waren}) \\ & + \alpha_{lm} \cdot \log p_{lm}(\text{als} \mid \ldots \text{ höher}) \end{array} \right.$$

Scoring Translation Hypotheses

November inflation rates | were higher than | expected in the | 13 countries of the euro...

⟨s⟩ Inflationsraten im November waren höher als erwartet in den

$$p(t \mid i, \mathcal{M}_{tr}, \mathcal{M}_{lm}, \mathcal{M}_{d}) =$$

$$\exp\left(\begin{array}{ll}
\alpha_{tr} \cdot \log p_{tr}(\text{Inflationsraten} \mid \text{inflation rates}) & + \alpha_{lm} \cdot \log p_{lm}(\text{Inflationsraten} \mid \langle s \rangle) \\
+ \alpha_{d} \cdot \log p_{d}(-2) & \\
+ \alpha_{tr} \cdot \log p_{tr}(\text{im November} \mid \text{November}) & + \alpha_{lm} \cdot \log p_{lm}(\text{im} \mid \ldots \text{ Inflationsraten}) \\
 & + \alpha_{lm} \cdot \log p_{lm}(\text{November} \mid \ldots \text{ in}) \\
+ \alpha_{d} \cdot \log p_{d}(+3) & \\
+ \alpha_{tr} \cdot \log p_{tr}(\text{waren} \ldots \text{als} \mid \text{were} \ldots \text{than}) & + \alpha_{lm} \cdot \log p_{lm}(\text{waren} \mid \ldots \text{ November}) \\
 & + \alpha_{lm} \cdot \log p_{lm}(\text{höher} \mid \ldots \text{ waren}) \\
 & + \alpha_{lm} \cdot \log p_{lm}(\text{als} \mid \ldots \text{ höher}) \\
\ldots &
\end{array}\right.$$

Scoring Translation Hypotheses

November inflation rates were higher than | expected in the | | 13 countries | of the euro. . .

⟨s⟩ Inflationsraten im November waren höher als erwartet in den 13 Ländern

$$
p(t \mid i, \mathcal{M}_{tr}, \mathcal{M}_{lm}, \mathcal{M}_{d}) =
$$

$$
\exp \left(\begin{array}{ll}
\alpha_{tr} \cdot \log p_{tr}\,(\text{Inflationsraten} \mid \text{inflation rates}) & + \alpha_{lm} \cdot \log p_{lm}\,(\text{Inflationsraten} \mid \langle s \rangle) \\
+ \alpha_{d} \cdot \log p_{d}(-2) & \\
+ \alpha_{tr} \cdot \log p_{tr}\,(\text{im November} \mid \text{November}) & + \alpha_{lm} \cdot \log p_{lm}\,(\text{im} \mid \ldots \text{ Inflationsraten}) \\
 & + \alpha_{lm} \cdot \log p_{lm}\,(\text{November} \mid \ldots \text{ in}) \\
+ \alpha_{d} \cdot \log p_{d}(+3) & \\
+ \alpha_{tr} \cdot \log p_{tr}\,(\text{waren} \ldots \text{ als} \mid \text{were} \ldots \text{ than}) & + \alpha_{lm} \cdot \log p_{lm}\,(\text{waren} \mid \ldots \text{ November}) \\
 & + \alpha_{lm} \cdot \log p_{lm}\,(\text{höher} \mid \ldots \text{ waren}) \\
 & + \alpha_{lm} \cdot \log p_{lm}\,(\text{als} \mid \ldots \text{ höher}) \\
\ldots &
\end{array} \right.
$$

Scoring Translation Hypotheses

... inflation rates were higher than expected in the 13 countries of the eurozone .

⟨s⟩ Inflationsraten ... waren höher als erwartet in den 13 Ländern der Eurozone .

$$p(t \mid i, \mathcal{M}_{tr}, \mathcal{M}_{lm}, \mathcal{M}_{d}) =$$

$$\exp\left(\begin{array}{ll}
\alpha_{tr} \cdot \log p_{tr}(\text{Inflationsraten} \mid \text{inflation rates}) & + \alpha_{lm} \cdot \log p_{lm}(\text{Inflationsraten} \mid \langle s \rangle) \\
+ \alpha_{d} \cdot \log p_{d}(-2) & \\
+ \alpha_{tr} \cdot \log p_{tr}(\text{im November} \mid \text{November}) & + \alpha_{lm} \cdot \log p_{lm}(\text{im} \mid \ldots \text{ Inflationsraten}) \\
 & + \alpha_{lm} \cdot \log p_{lm}(\text{November} \mid \ldots \text{ in}) \\
+ \alpha_{d} \cdot \log p_{d}(+3) & \\
+ \alpha_{tr} \cdot \log p_{tr}(\text{waren ... als} \mid \text{were ... than}) & + \alpha_{lm} \cdot \log p_{lm}(\text{waren} \mid \ldots \text{ November}) \\
 & + \alpha_{lm} \cdot \log p_{lm}(\text{höher} \mid \ldots \text{ waren}) \\
 & + \alpha_{lm} \cdot \log p_{lm}(\text{als} \mid \ldots \text{ höher}) \\
\ldots &
\end{array}\right.$$

Distortion Modeling

- Exponential probability decay over distance:

$$p_d(x) = \gamma^{\text{abs}(x)}$$

- Lexicalized discrete model (Koehn et al., 2005)
  - Estimated seperately for each phrase.
  - Three types of *type*(j) of jumps:

mono	phrase immediately follows the previously translated phrase
swap	phrase swaps positions with the previously translated phrase
other	anything else

- ...

Decoding

Given the model, find the best translation

$$\mathbf{e}_{\text{best}} = \text{argmax}_{\mathbf{e}}\, p(\mathbf{e} \,|\, \mathbf{f})$$

We use the "Viterbi approximation"

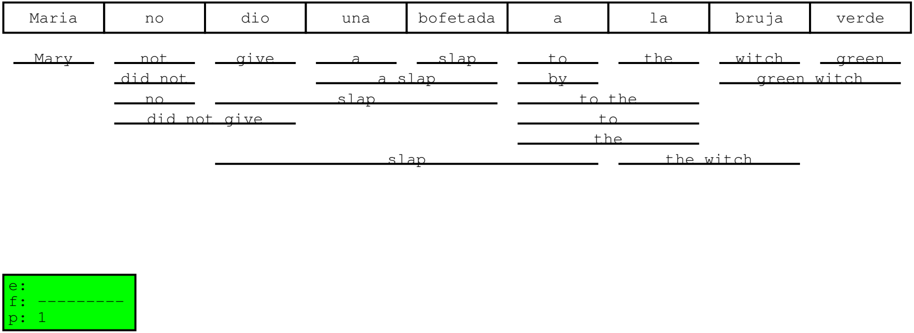
$$(a, \mathbf{e})_{\text{best}} = \text{argmax}_{(a,\mathbf{e})}\, p(a, \mathbf{e} \,|\, \mathbf{f})$$

- This is a search problem - a big one.
  - Dynamic programming
  - Approximation (beam search)
  - Model restrictions (reordering)

Decoding

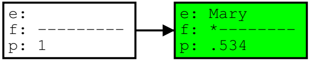
- many different ways to *segment* the input sentence into phrases
- many different ways to *translate* each phrase

Hypothesis Expansion



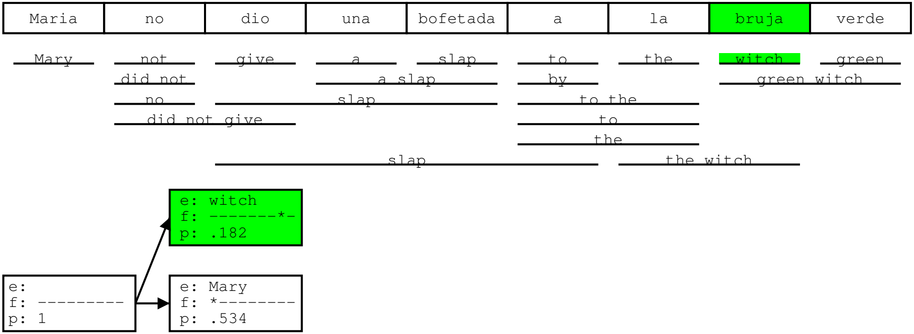
- Start with empty hypothesis
  - e: no English words
  - f: no foreign words covered
  - p: probability 1

Hypothesis Expansion



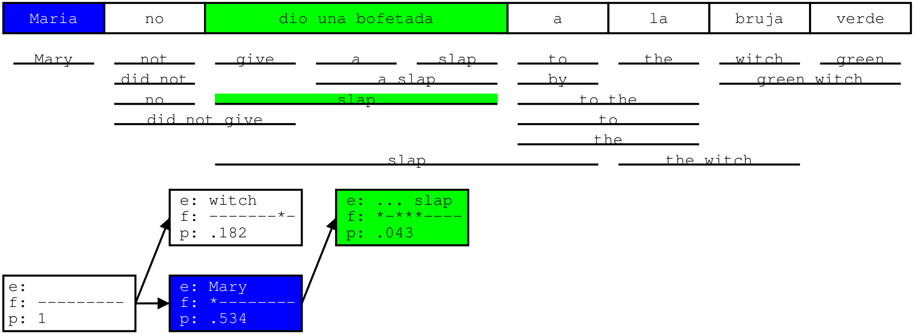
- Pick *translation option*
- Create *hypothesis*
  - e: add English phrase `Mary`
  - f: first foreign word covered
  - p: probability 0.534

Hypothesis Expansion



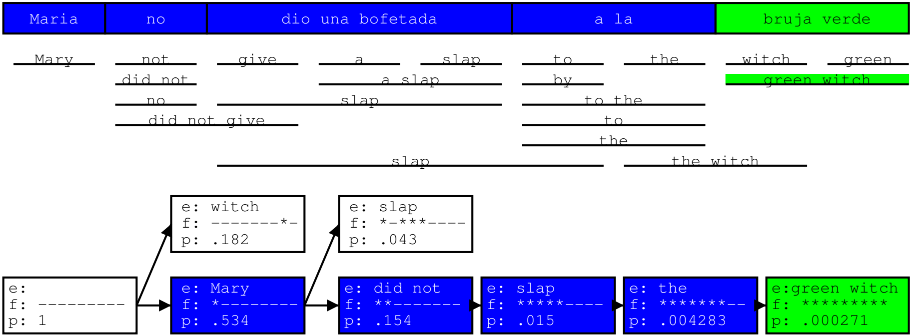
- Add another *hypothesis*

Hypothesis Expansion



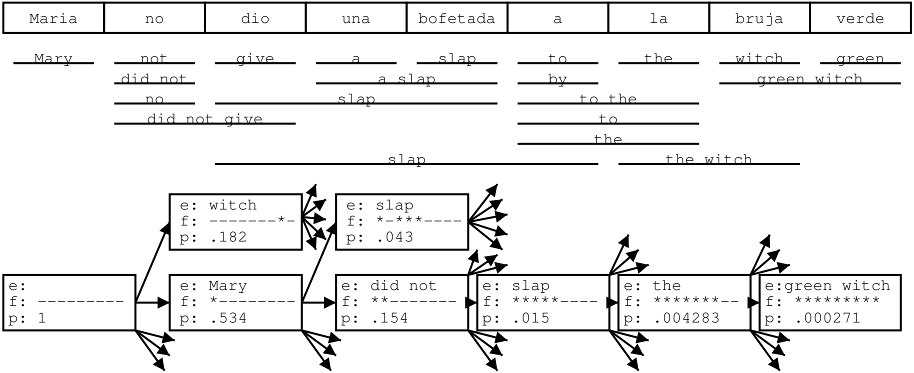
- Further *hypothesis expansion*

Hypothesis Expansion



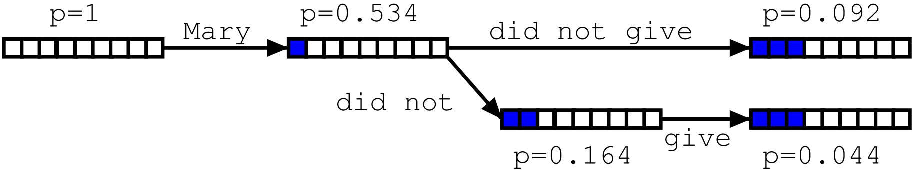
- . . . until all foreign words *covered*
  - find *best hypothesis* that covers all foreign words
  - *backtrack* to read off translation

Hypothesis Expansion



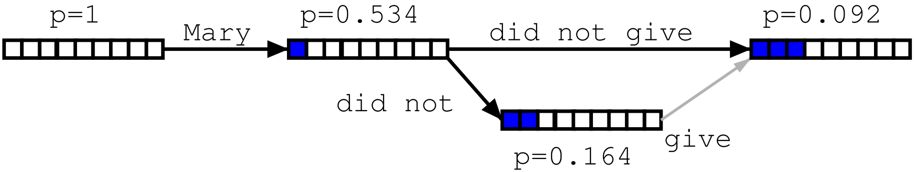
Adding more hypothesis ⇒ *Explosion* of search space

Hypothesis Recombination



- Different paths to the *same* partial translation

Hypothesis Recombination

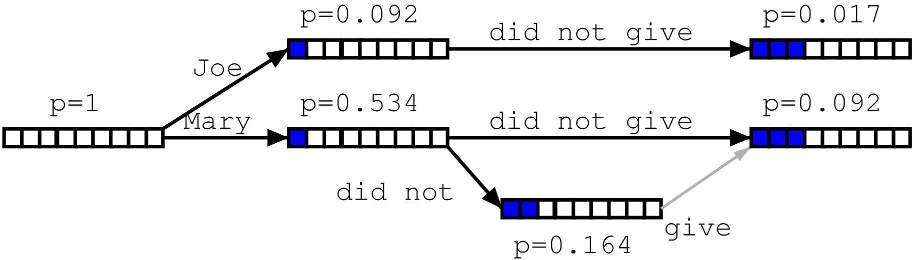


- Different paths to the same partial translation

⇒ *Combine paths*

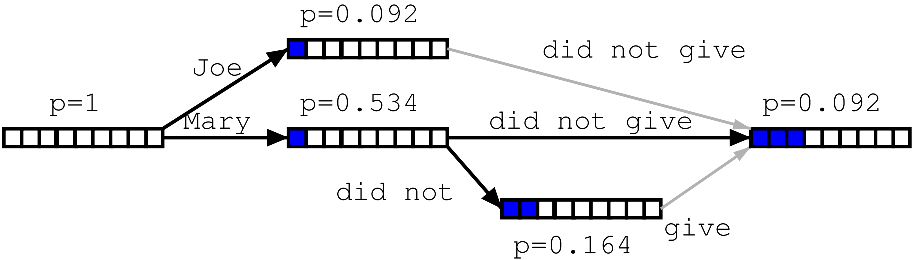
  - *drop weaker* path
  - keep pointer from weaker path (for lattice generation)

Hypothesis Recombination



- Recombined hypotheses do *not* have to *match completely*
- No matter what is added, weaker path can be dropped, if:
  - *last $n-1$ English words* match (matters for language model)
  - *foreign word coverage* vectors match (affects future path)

Hypothesis Recombination



- Recombined hypotheses do not have to match completely
- No matter what is added, weaker path can be dropped, if:
    - last $n-1$ English words match (matters for language model)
    - foreign word coverage vectors match (effects future path)

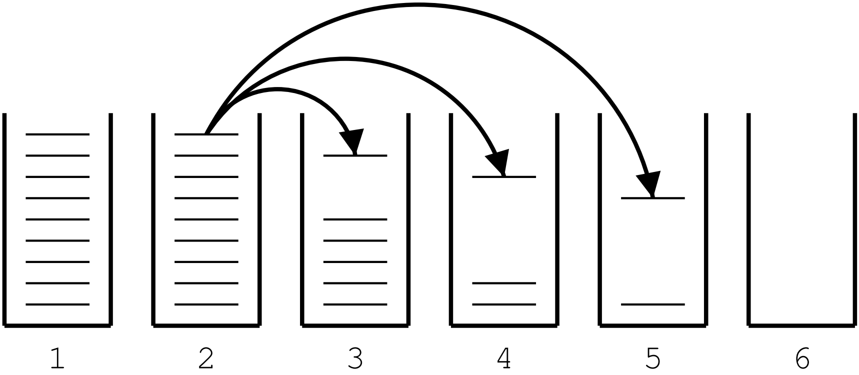
⇒ *Combine paths*

Beam Search

heuristically *discard* weak hypotheses early

- it is better to organize hypotheses in stacks (actually: priority queues), e.g. by
  - *same* foreign words covered
  - *same number* of foreign words covered
- compare hypotheses in stacks, discard bad ones
  - histogram pruning: keep top k hypotheses in each stack (e.g., k=100)
  - threshold pruning: keep hypotheses that are at most α times the cost of best hypothesis in stack (e.g., $\alpha = 0.001$)

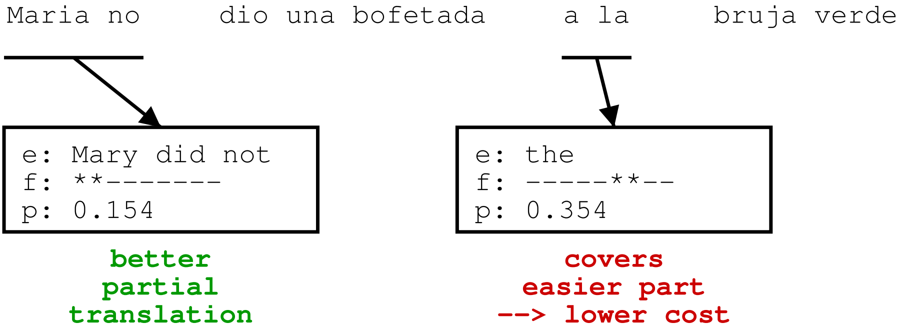
Hypothesis Stacks



- Organization of hypotheses into stacks
  - here: based on *number of foreign words* translated
  - during translation all hypotheses from one stack are expanded
  - expanded hypotheses are placed into stacks

Comparing Hypotheses

- Comparing hypotheses with *same number of foreign words* covered

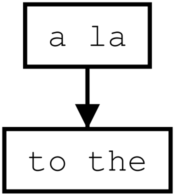


- Hypothesis that covers *easy part* of sentence is preferred

⇒ Need to consider future cost of uncovered parts

Future Cost Estimation

Step 1: estimate future cost for each *translation option*

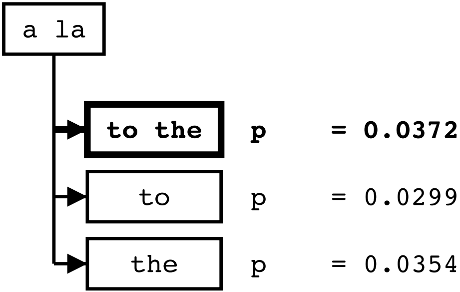


- look up translation model cost
- estimate language model cost (no prior context)
- ignore reordering model cost

⇒ LM * TM = p(to) * p(the|to) * p(to the|a la)

Future Cost Estimation

Step 2: find *cheapest cost* (highest probability) among translation options



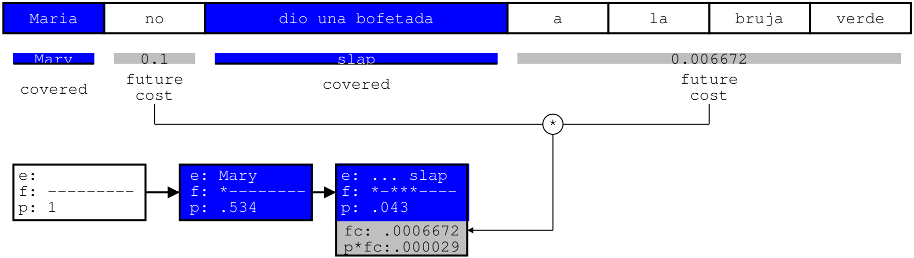
Future Cost Estimation

Step 3: Find *lowest future cost* for each possible span

- Cost of translation option for that span, *or*
- Sum of costs of covering subspans

⇒ Pre-compute future costs, bottom up., via dynamic programming.

Future Cost Estimation: Application

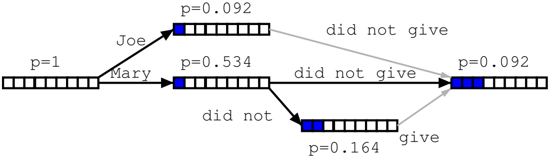


- Use future cost estimates when *pruning* hypotheses
- For each *uncovered continuous span*:
  - look up *future costs* for each maximal contiguous uncovered span
  - *add* to actually accumulated cost for translation option for pruning

Limits on Reordering

- Reordering may be limited
  - Monotone translation: No reordering at all
  - Only phrase movements of at most d words
- Reordering limits *speed* up search (polynomial instead of exponential)
- Current reordering models are weak, so limits *improve* translation quality

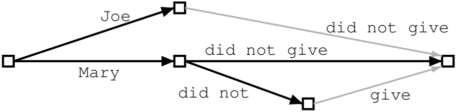
Word Lattice Generation



- Search graph can be easily converted into a word lattice
  - can be further mined for N-best lists

  ⇒ enables reranking approaches

  ⇒ enables discriminative training



Sample N-Best List

- Simple N-best list:

Translation	Reordering	LM	TM	WordPenalty	Score
this is a small house	0	-27.0908	-1.83258	-5	-28.9234
this is a little house	0	-28.1791	-1.83258	-5	-30.0117
it is a small house	0	-27.108	-3.21888	-5	-30.3268
it is a little house	0	-28.1963	-3.21888	-5	-31.4152
this is an small house	0	-31.7294	-1.83258	-5	-33.562
it is an small house	0	-32.3094	-3.21888	-5	-35.5283
this is an little house	0	-33.7639	-1.83258	-5	-35.5965
this is a house small	-3	-31.4851	-1.83258	-5	-36.3176
this is a house little	-3	-31.5689	-1.83258	-5	-36.4015
it is an little house	0	-34.3439	-3.21888	-5	-37.5628
it is a house small	-3	-31.5022	-3.21888	-5	-37.7211
this is an house small	-3	-32.8999	-1.83258	-5	-37.7325
it is a house little	-3	-31.586	-3.21888	-5	-37.8049
this is an house little	-3	-32.9837	-1.83258	-5	-37.8163
the house is a little	-7	-28.5107	-2.52573	-5	-38.0364
the is a small house	0	-35.6899	-2.52573	-5	-38.2156
is it a little house	-4	-30.3603	-3.91202	-5	-38.2723
the house is a small	-7	-28.7683	-2.52573	-5	-38.294
it 's a small house	0	-34.8557	-3.91202	-5	-38.7677
this house is a little	-7	-28.0443	-3.91202	-5	-38.9563
it 's a little house	0	-35.1446	-3.91202	-5	-39.0566
this house is a small	-7	-28.3018	-3.91202	-5	-39.2139

Summary

- Left-to-right decoding as search
- Hypothesis recombination
- Pruning
- Future cost estimation
- Word lattices and *n*-best lists

So long and thanks for all the fish?

A few personal musings

- PBSMT lead the field for more than a decade.
- Until very recently the most successful approach.
- Widely used in commercial systems.
- Despite recent developments, still a strong contender for jobs with high overlap with existing data.
- But let's face it:
  - Philosophically, not really an attractive model of the translation process to begin with.
  - "hard" distortion limit makes correct translations impossible to reach in certain cases (e.g., long subordinate clauses in German).
  - standard PBSMT doesn't allow for gappy phrases
    Er nahm aus Krankeitsgründen doch nicht teil.