Statistical Machine Translation
Part IV – Decoding

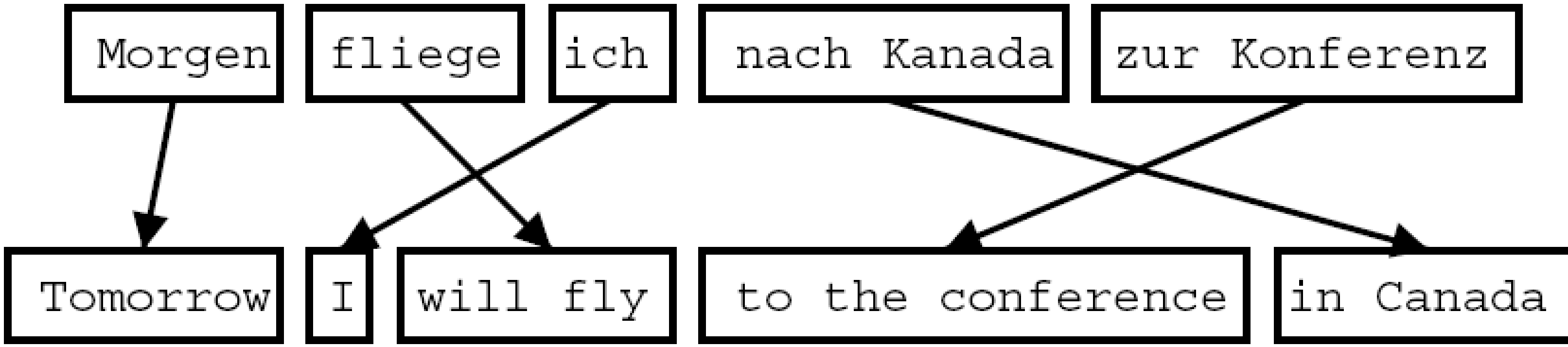
Alexander Fraser
CIS, LMU München

2014.11.04   WSD and MT

Outline

- *Phrase-based translation model*
- Decoding
  - Basic phrase-based decoding
  - Dealing with complexity
    - Recombination
    - Pruning
    - Future cost estimation

Phrase-based translation



- Foreign input is segmented in phrases
  - any sequence of words, not necessarily linguistically motivated
- Each phrase is translated into English
- Phrases are reordered

Slide from Koehn 2008

Phrase-based translation model

- Major components of phrase-based model
  - **phrase translation model** $\phi(\mathbf{f}|\mathbf{e})$
  - **reordering model** d
  - **language model** $p_{\text{LM}}(\mathbf{e})$
- Bayes rule

$$\begin{aligned}\text{argmax}_{\mathbf{e}} p(\mathbf{e}|\mathbf{f}) &= \text{argmax}_{\mathbf{e}} p(\mathbf{f}|\mathbf{e}) p(\mathbf{e}) \\ &= \text{argmax}_{\mathbf{e}} \phi(\mathbf{f}|\mathbf{e}) p_{\text{LM}}(\mathbf{e}) \omega^{\text{length}(\mathbf{e})}\end{aligned}$$

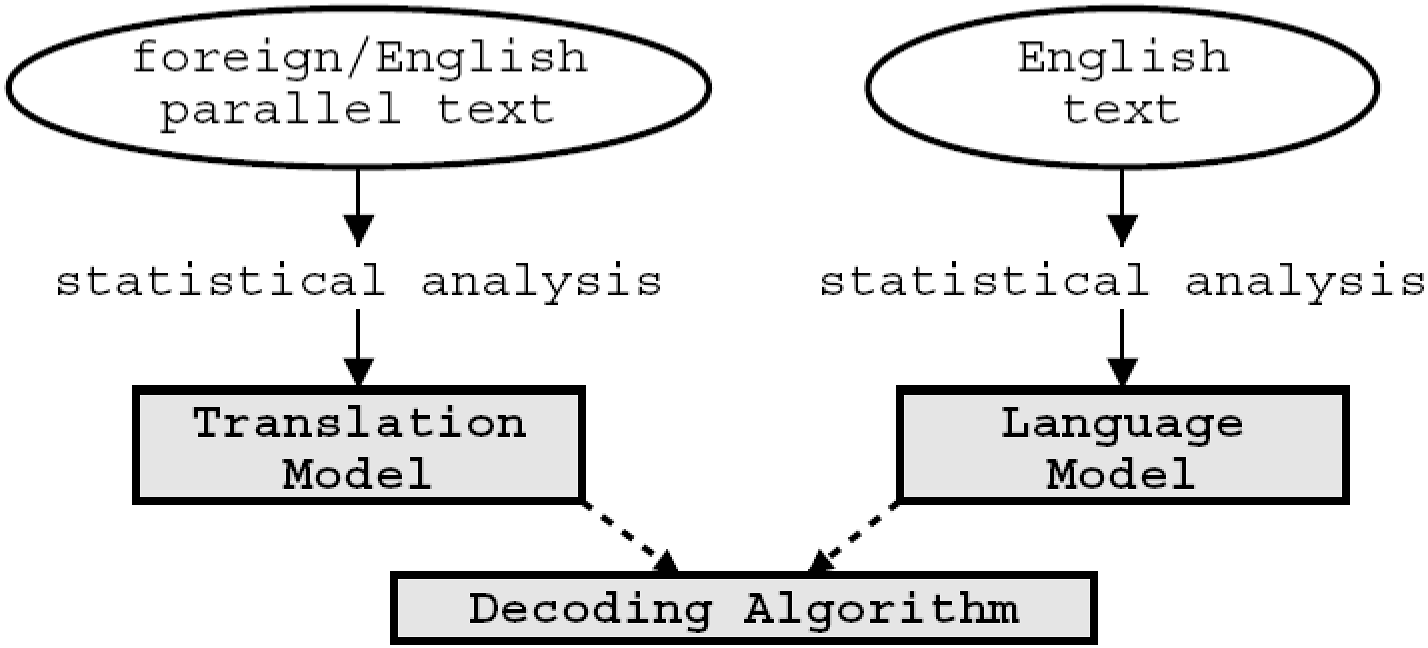
- Sentence $\mathbf{f}$ is decomposed into I phrases $\bar{f}_1^I = \bar{f}_1, ..., \bar{f}_I$
- Decomposition of $\phi(\mathbf{f}|\mathbf{e})$

$$\phi(\bar{f}_1^I|\bar{e}_1^I) = \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i) d(a_i - b_{i-1})$$

Modified from Koehn 2008

Statistical Machine Translation

- Components: Translation model, language model, decoder



Slide from Koehn 2008

Decoding

- Goal: find the best target translation of a source sentence
- Involves search
  - Find maximum probability path in a dynamically generated search graph
- Generate English string, from left to right, by covering parts of Foreign string
  - Generating English string left to right allows scoring with the n-gram language model
- Here is an example of one path

Decoding Process

Maria	no	dio	una	bofetada	a	la	bruja	verde

- Build translation left to right
  - *select foreign* words to be translated

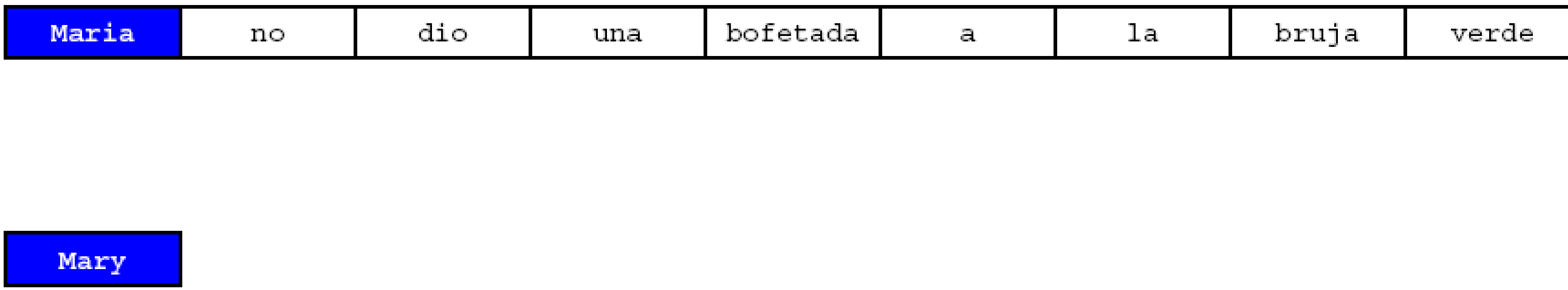
Slide from Koehn 2008

Decoding Process

- Build translation *left to right*
  - select foreign words to be translated
  - *find English* phrase translation
  - *add English* phrase to end of partial translation

Slide from Koehn 2008

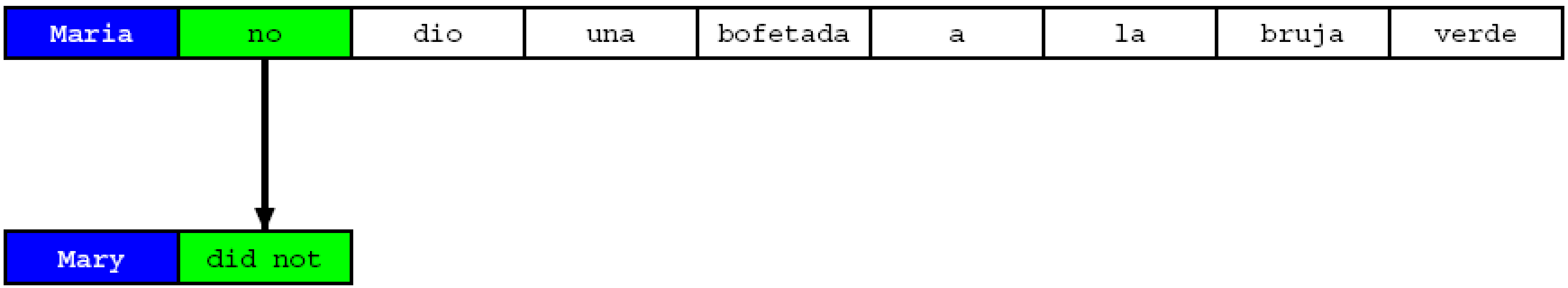
Decoding Process



- Build translation left to right
  - select foreign words to be translated
  - find English phrase translation
  - add English phrase to end of partial translation
  - *mark foreign* words as translated

Slide from Koehn 2008

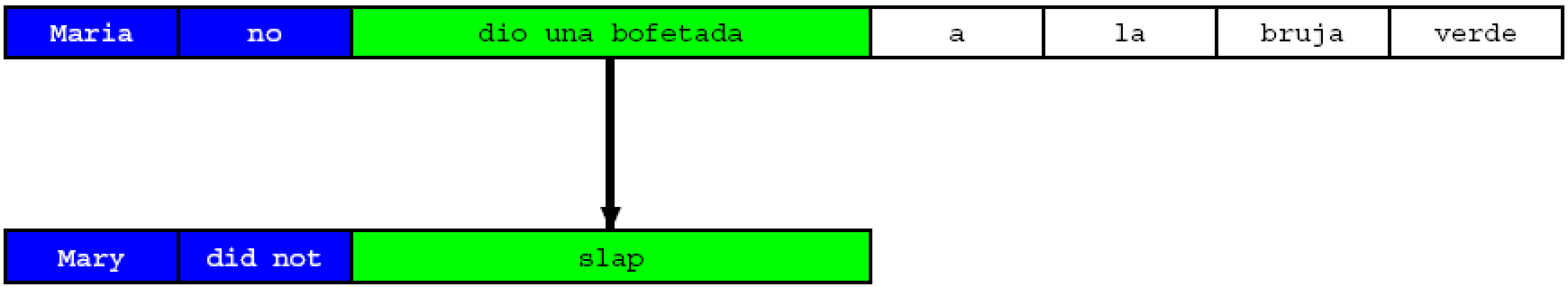
Decoding Process



- *One to many* translation

Slide from Koehn 2008

Decoding Process



- Many to one translation

Slide from Koehn 2008

Decoding Process

- *Many to one* translation

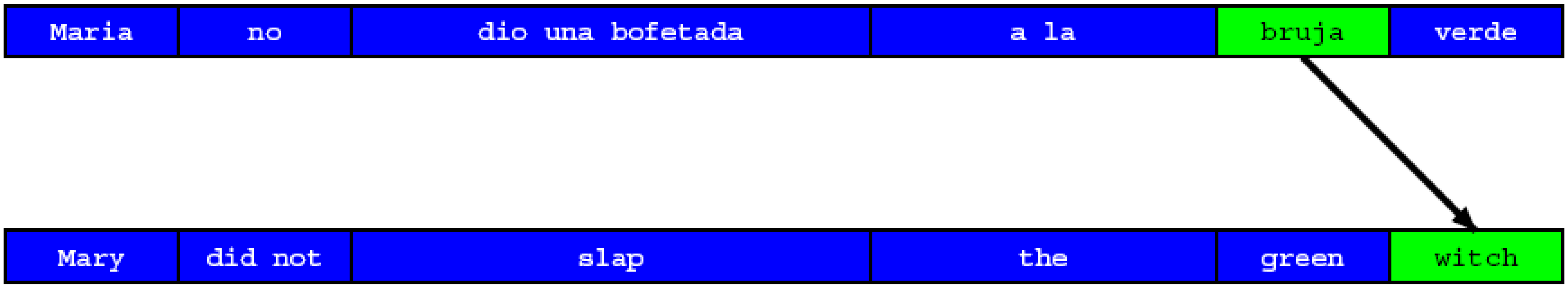
Slide from Koehn 2008

Decoding Process

- *Reordering*

Slide from Koehn 2008

Decoding Process



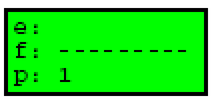
- Translation *finished*

Slide from Koehn 2008

Translation Options

- Look up *possible phrase translations*
  - many different ways to *segment* words into phrases
  - many different ways to *translate* each phrase

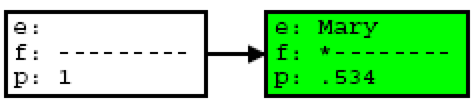
Hypothesis Expansion



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

Slide from Koehn 2008

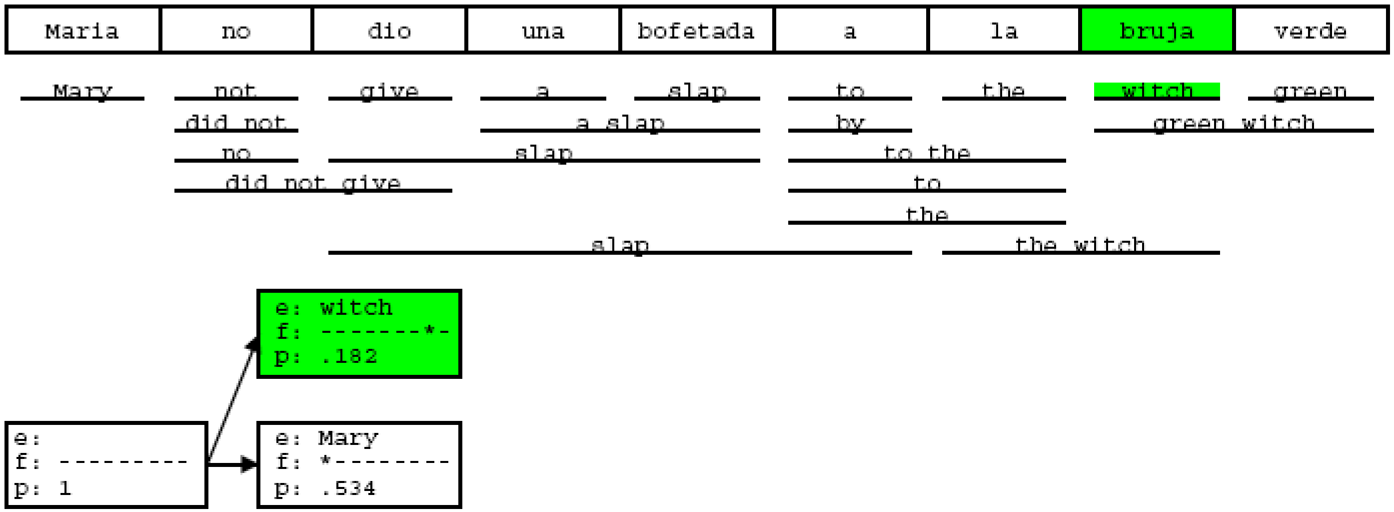
Hypothesis Expansion



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

Slide from Koehn 2008

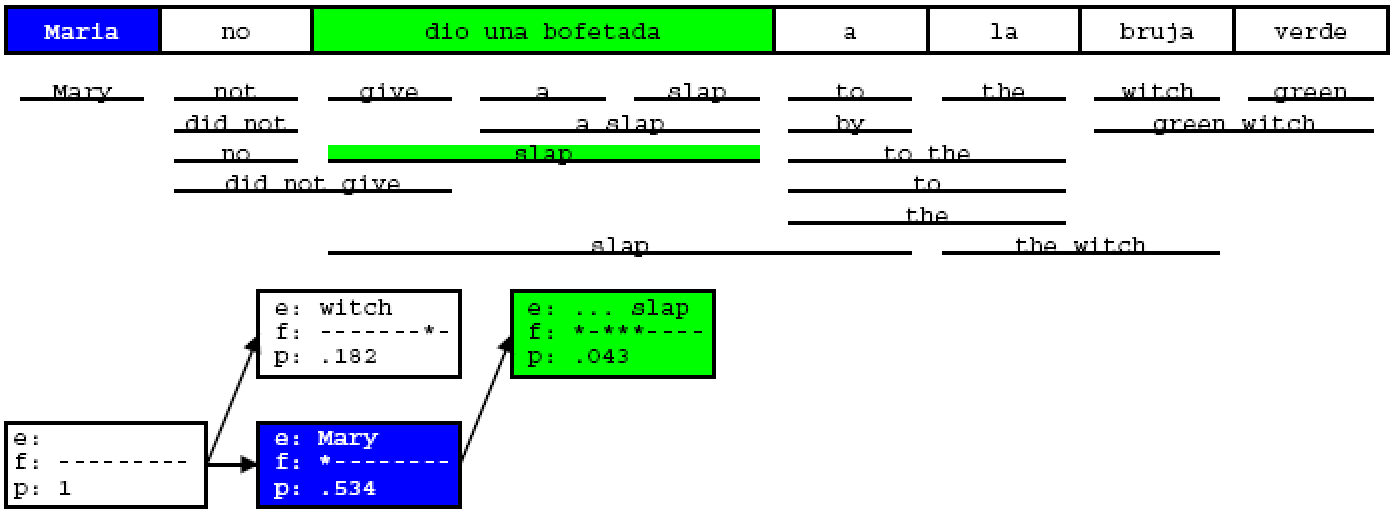
Hypothesis Expansion



- Add another *hypothesis*

Slide from Koehn 2008

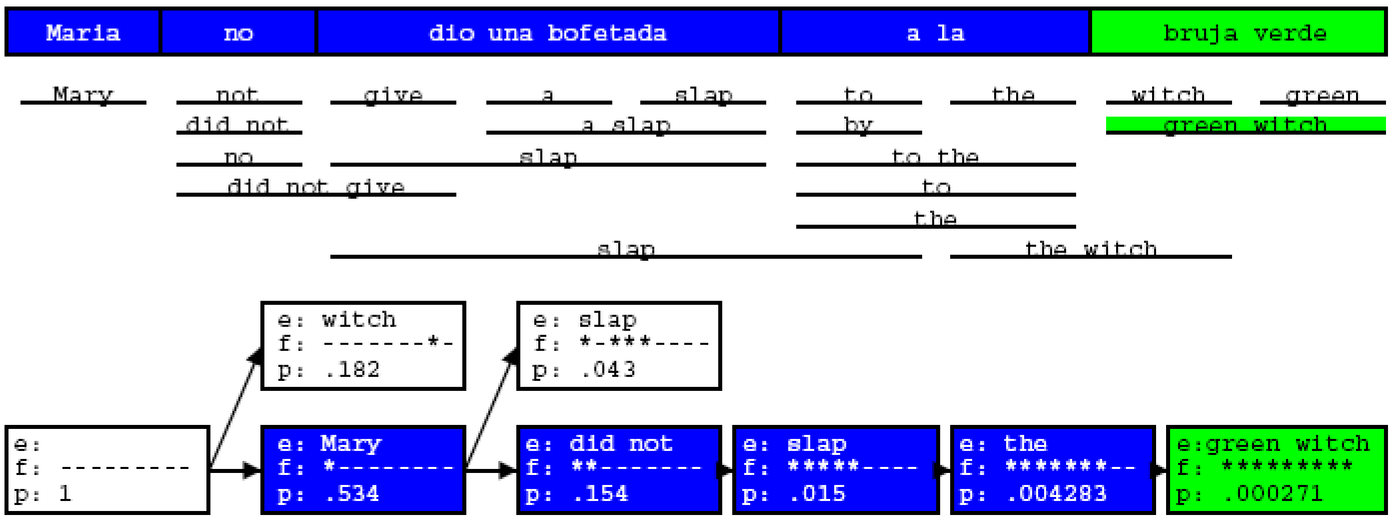
Hypothesis Expansion



- Further *hypothesis expansion*

Slide from Koehn 2008

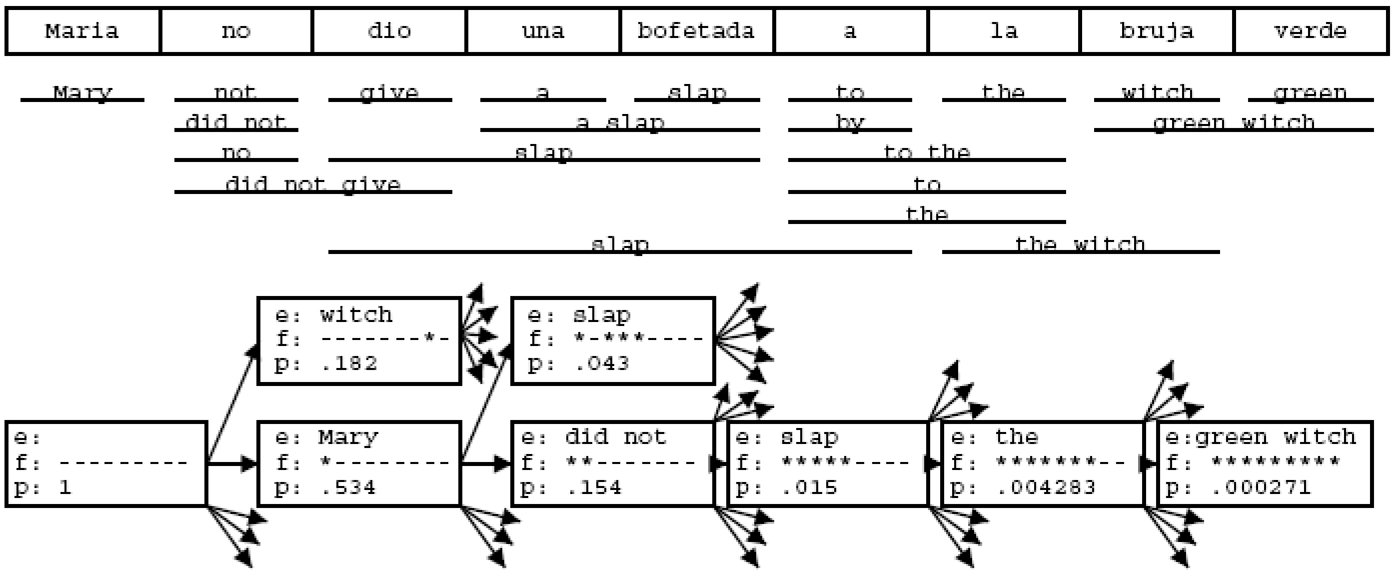
Hypothesis Expansion



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

Slide from Koehn 2008

Hypothesis Expansion



- Adding more hypothesis

⇒ *Explosion* of search space

Slide from Koehn 2008

Explosion of Search Space

- Number of hypotheses is *exponential* with respect to sentence length

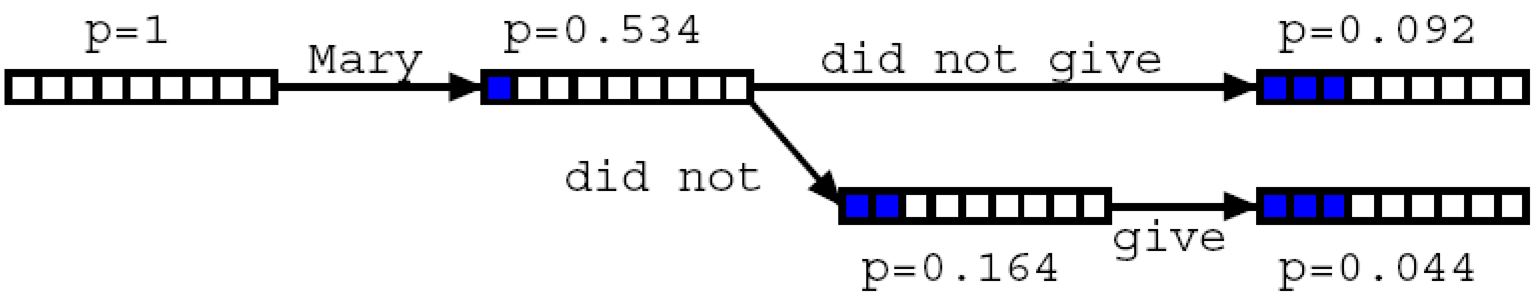
⇒ Decoding is NP-complete [Knight, 1999]

⇒ Need to *reduce search space*

- risk free: hypothesis **recombination**
- risky: **histogram/threshold pruning**

Slide from Koehn 2008

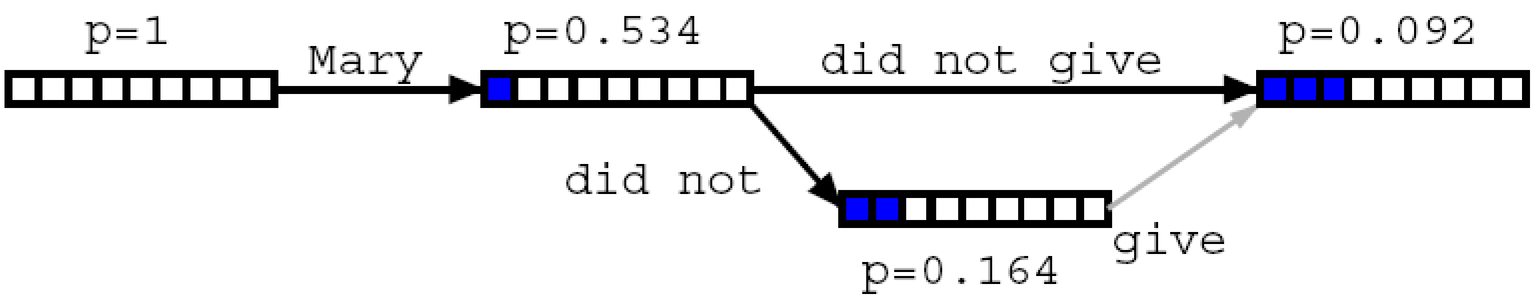
Hypothesis Recombination



- Different paths to the *same* partial translation

Slide from Koehn 2008

Hypothesis Recombination

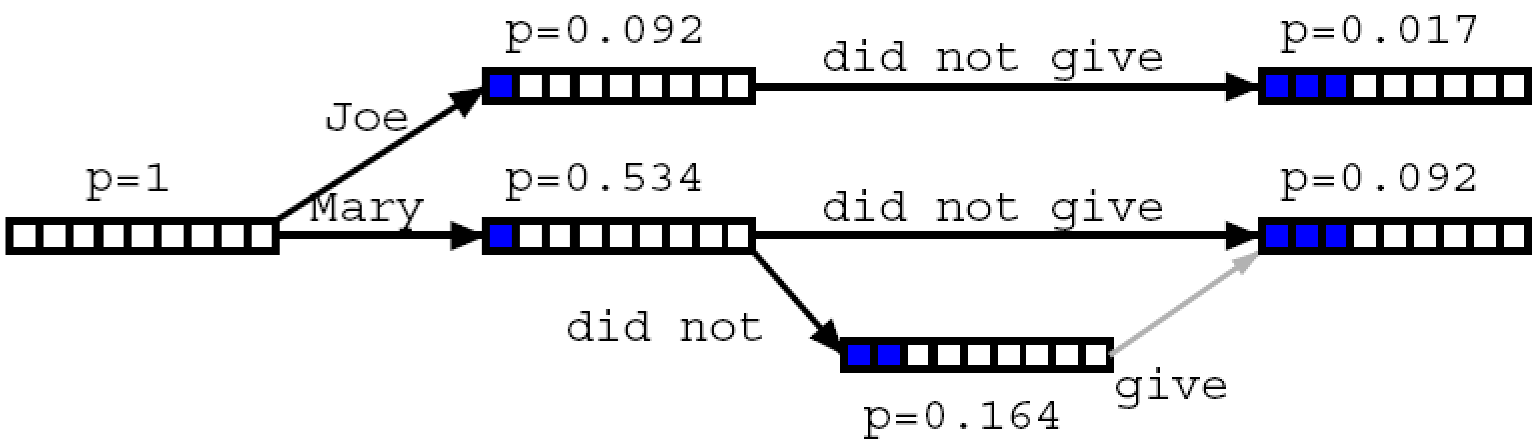


- Different paths to the same partial translation

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

Slide from Koehn 2008

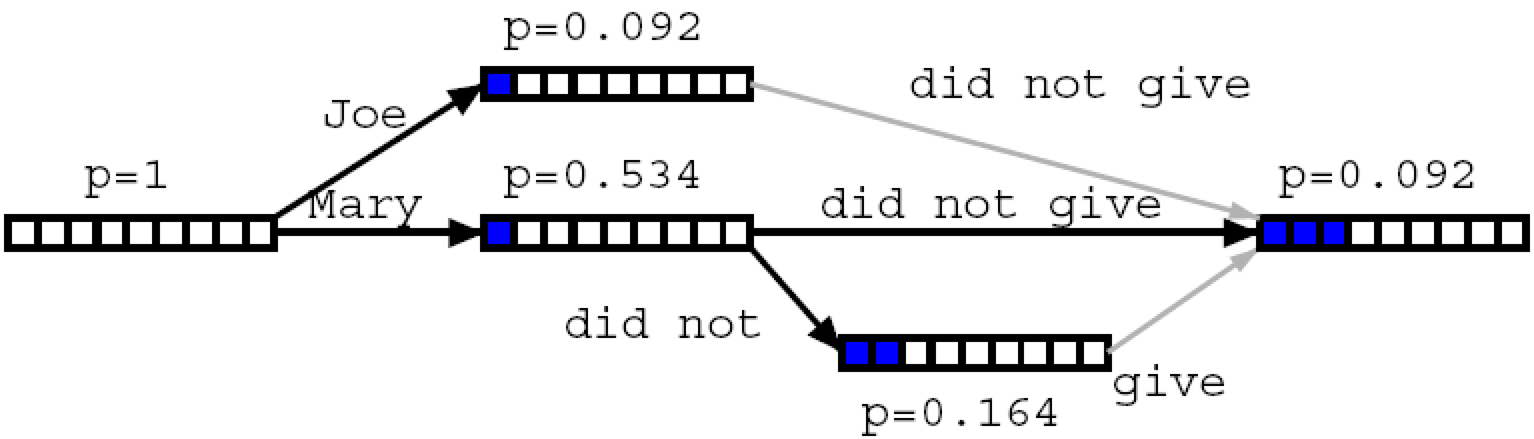
Hypothesis Recombination



- Recombined hypotheses do *not* have to *match completely*
- No matter what is added, weaker path can be dropped, if:
  - *last two English words* match (matters for language model)
  - *foreign word coverage* vectors match (possible future paths are the same)

Modified from Koehn 2008

Hypothesis Recombination



- Recombined hypotheses do not have to match completely
- No matter what is added, weaker path can be dropped, if:
  - last two English words match (matters for language model)
  - foreign word coverage vectors match (possible future paths are the same)

⇒ *Combine paths*

Modified from Koehn 2008

Pruning

- Hypothesis recombination is *not sufficient*

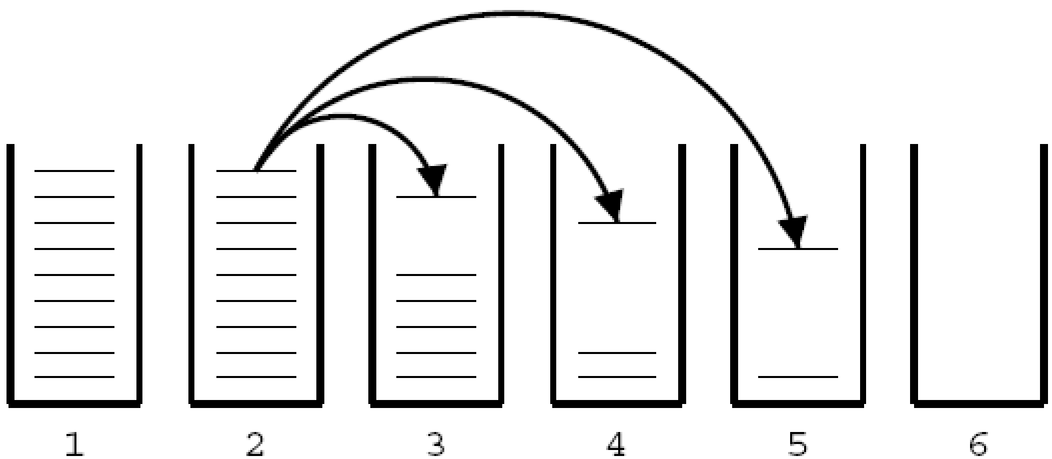
⇒ Heuristically *discard* weak hypotheses early

- Organize Hypothesis in **stacks**, e.g. by
  - *same* foreign words covered
  - *same number* of foreign words covered
  - *same number* of English words produced

- Compare hypotheses in stacks, discard bad ones
  - **histogram pruning**: keep top n hypotheses in each stack (e.g., n=100)
  - **threshold pruning**: keep hypotheses that are at most α times the cost of best hypothesis in stack (e.g., $\alpha = 0.001$)

Slide from Koehn 2008

Hypothesis Stacks

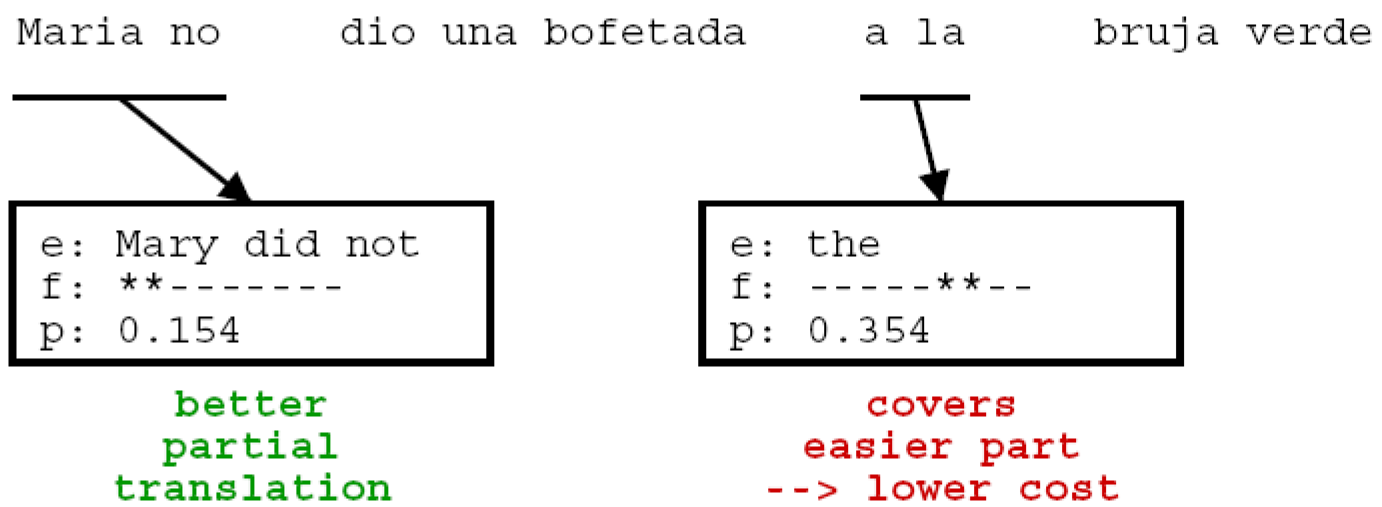


- Organization of hypothesis into stacks
  - here: based on *number of foreign words* translated
  - during translation all hypotheses from one stack are expanded
  - expanded Hypotheses are placed into stacks

Slide from Koehn 2008

Comparing Hypotheses

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

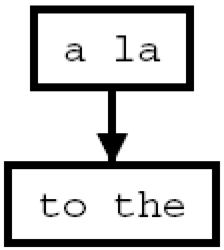


- Hypothesis that covers *easy part* of sentence is preferred

$\Rightarrow$ Need to consider **future cost** of uncovered parts

Slide from Koehn 2008

Future Cost Estimation



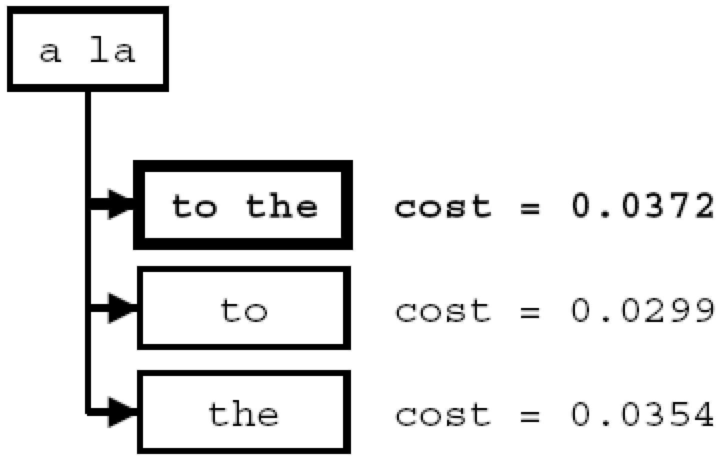
- *Estimate cost* to translate remaining part of input

- 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)

Slide from Koehn 2008

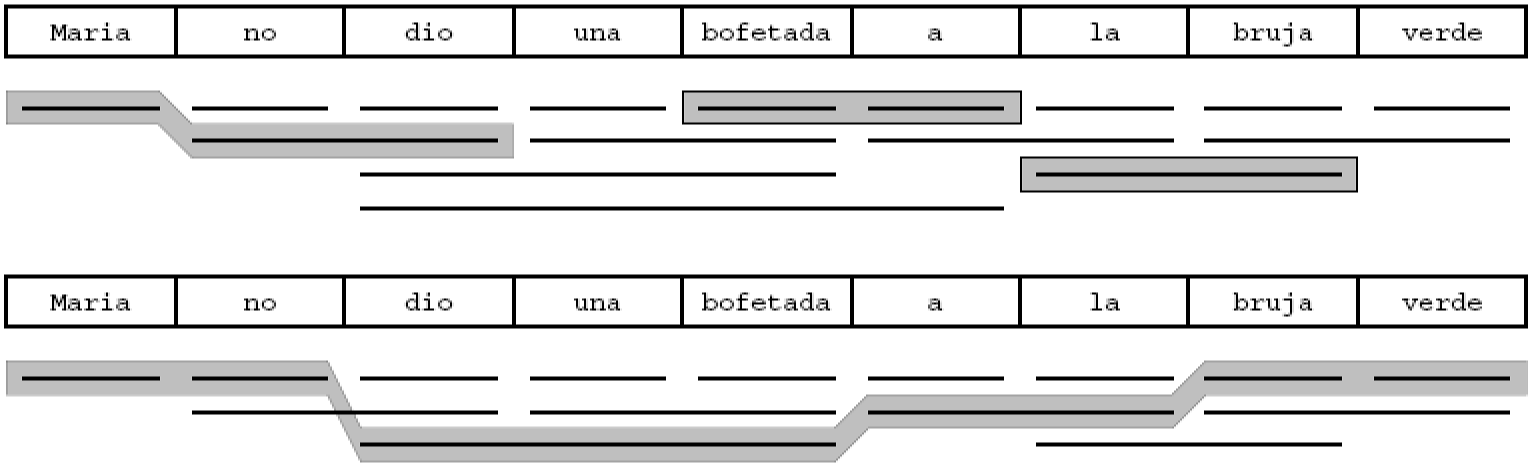
Future Cost Estimation: Step 2



- Step 2: find *cheapest cost* among translation options

Slide from Koehn 2008

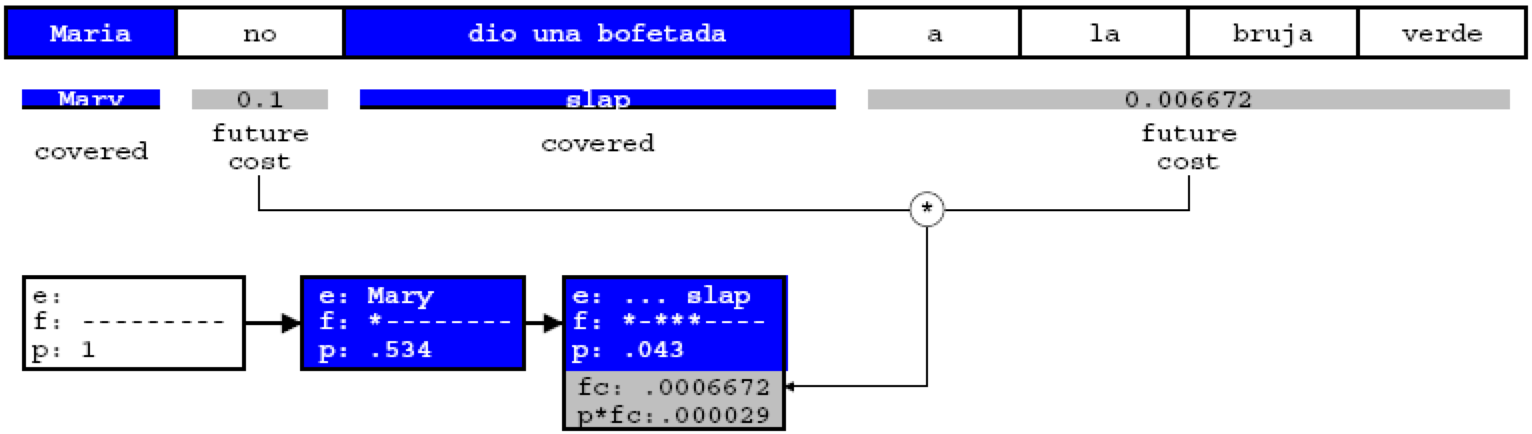
Future Cost Estimation: Step 3



- Step 3: find *cheapest future cost path* for each span
  - can be done *efficiently* by dynamic programming
  - future cost for every span can be *pre-computed*

Slide from Koehn 2008

Future Cost Estimation: Application



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

Slide from Koehn 2008

A* search

- Pruning might drop hypothesis that lead to the best path (**search error**)
- **A* search**: safe pruning
  - future cost estimates have to be accurate or underestimates
  - **lower bound** for probability is established early by **depth first search**: compute cost for one complete translation
  - if cost-so-far and future cost are worse than *lower bound*, hypothesis can be safely discarded
- Not commonly done, since not aggressive enough

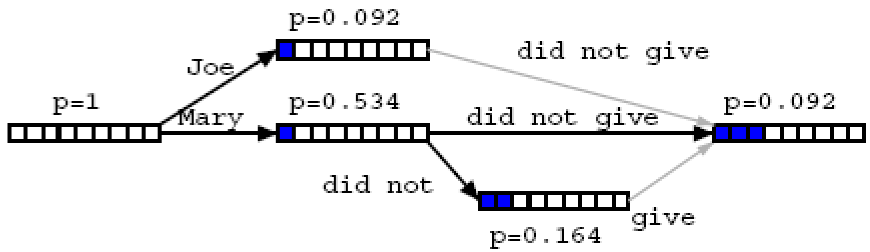
Slide from Koehn 2008

Limits on Reordering

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

Slide from Koehn 2008

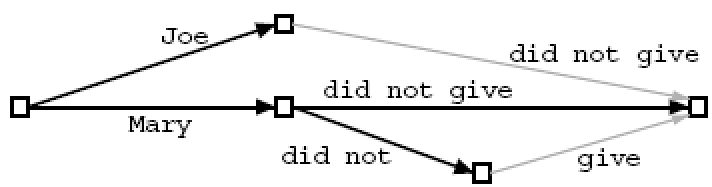
Word Lattice Generation



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

  → enables **reranking** approaches

  → enables **discriminative training**



Slide from Koehn 2008