Introduction to Statistical Machine Translation

Philipp Koehn

28 November 2008

Topics

- Introduction
- Word-based models and the EM algorithm
- Decoding
- Phrase-based models
- Open source: Moses
- Syntax-based statistical MT
- Factored models
- Large-Scale discriminative training

Machine translation

- Task: translate this into English

毒品

本册子爲家長們提供實際和有用的關于毒品
的信息，包括如何減少使用非法毒品的危險。
它有助於您和您的家人討論有關毒品的問題。
這本小册子的主要內容已錄在磁帶上。如果您
想索取一盒免費的磁帶(中文)，請在下面的

- One of the oldest problems in Artificial Intelligence

- AI-hard: reasoning and world knowledge required

The Rosetta stone

- Egyptian language was a mystery for centuries
- 1799 a stone with Egyptian text and its translation into Greek was found

⇒ Humans *could learn* how to translated Egyptian

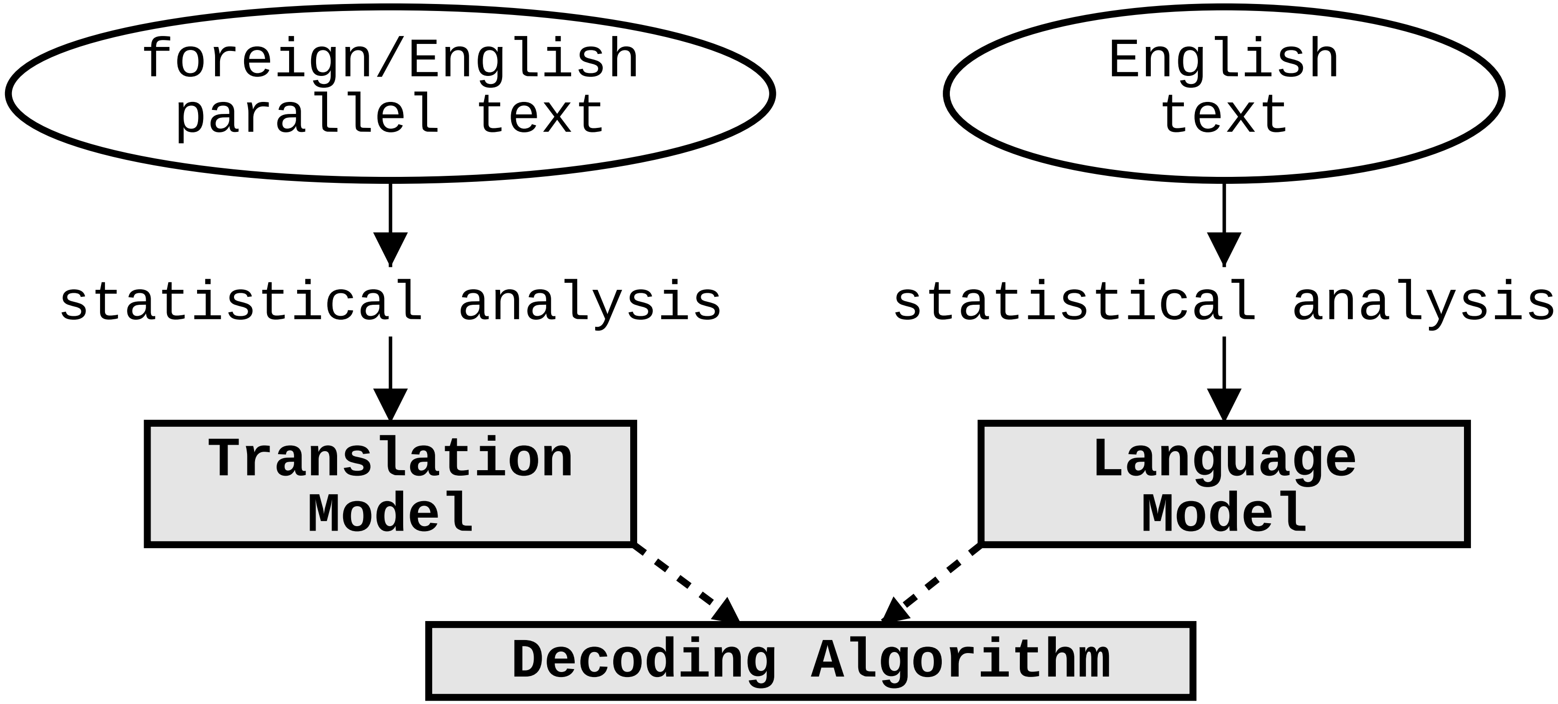
Parallel data

- Lots of translated text available: 100s of million words of translated text for some language pairs
  - a book has a few 100,000s words
  - an educated person may read 10,000 words a day
  - $\rightarrow$ 3.5 million words a year
  - $\rightarrow$ *300 million a lifetime*
  - $\rightarrow$ soon computers will be able to see more translated text than humans read in a lifetime

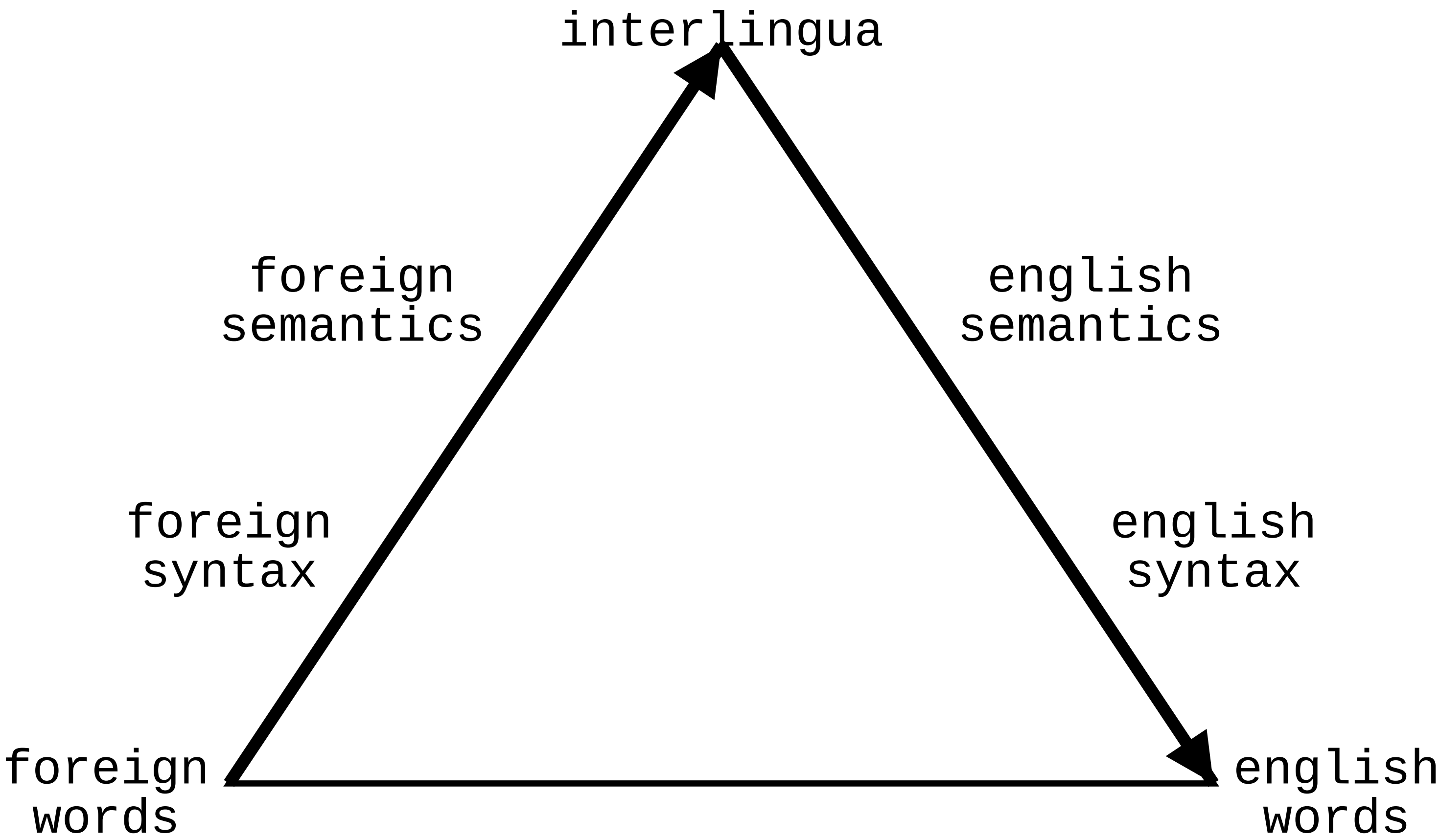
$\Rightarrow$ Machine *can learn* how to translated foreign languages

Statistical machine translation

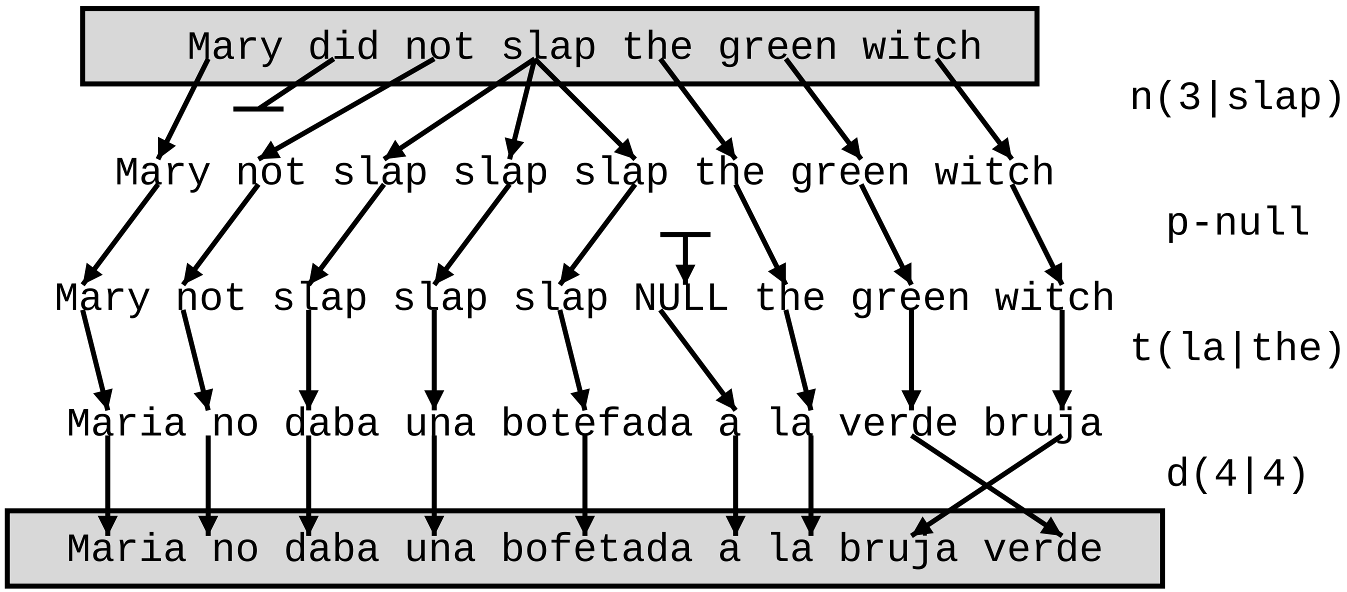
- Components: **Translation model**, **language model**, **decoder**



The machine translation pyramid



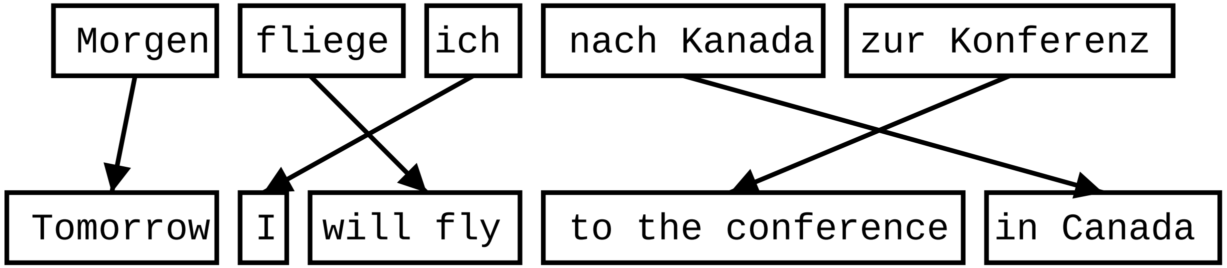
Word-based models



[from Knight, 1997]

- Translation process is *decomposed into smaller steps*, each is tied to words
- Original models for statistical machine translation [Brown et al., 1993]

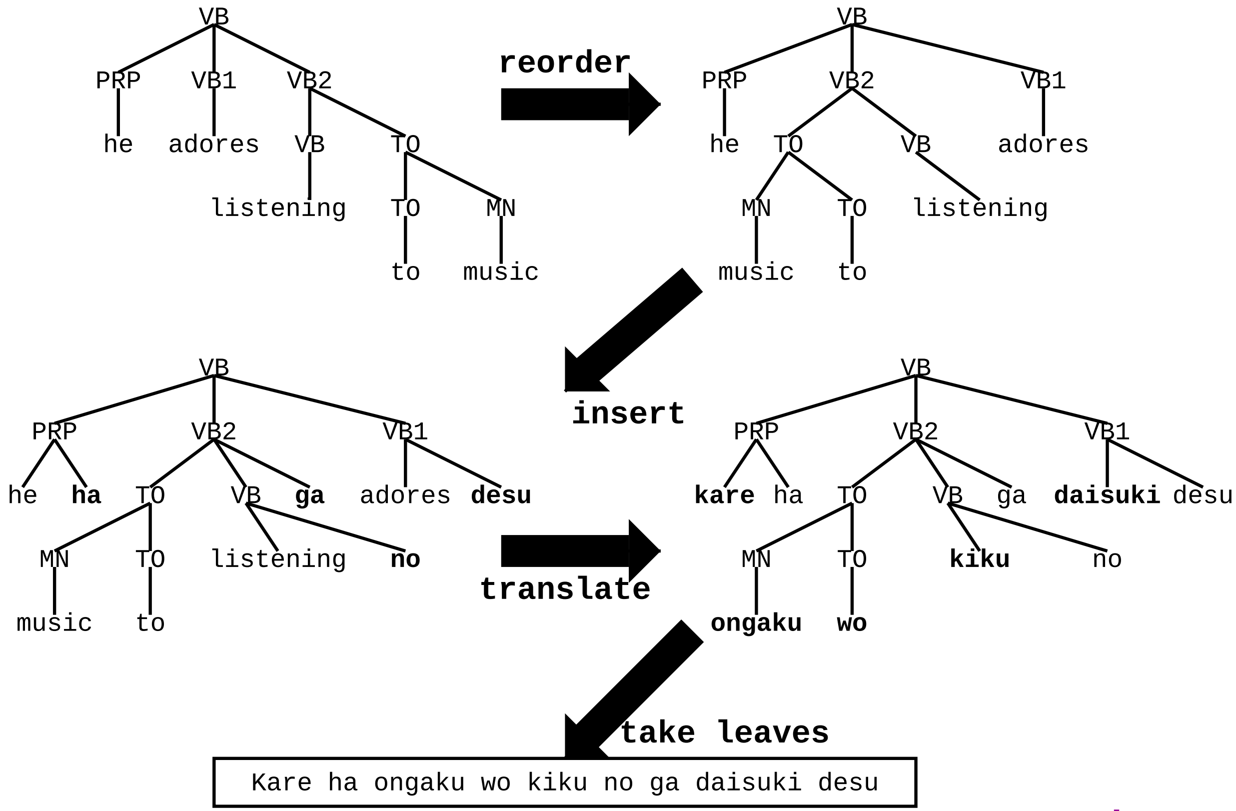
Phrase-based models



[from Koehn et al., 2003, NAACL]

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

Syntax-based models



[from Yamada and Knight, 2001]

Automatic evaluation

- Why **automatic evaluation** metrics?
  - Manual evaluation is *too slow*
  - Evaluation on large test sets *reveals minor improvements*
  - **Automatic tuning** to improve machine translation performance

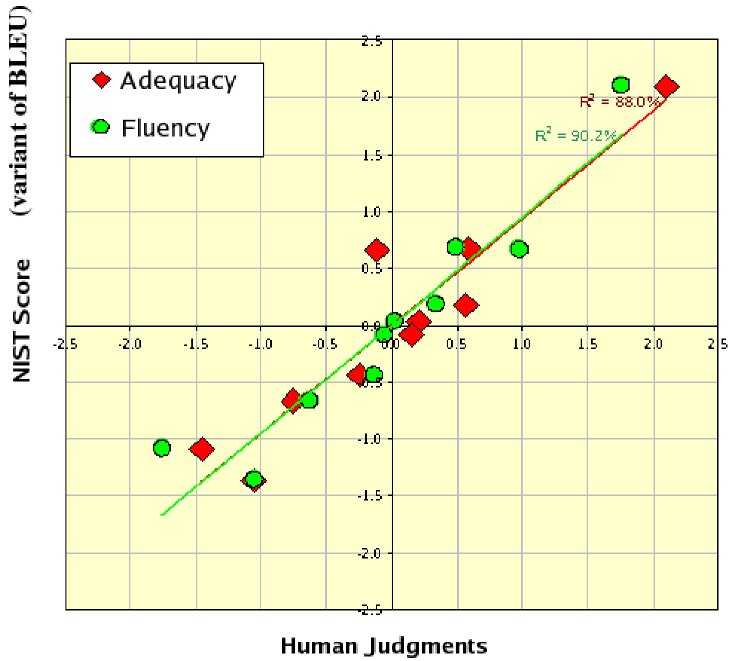
- History
  - Word Error Rate
  - **BLEU** since 2002

- BLEU in short: *Overlap with reference* translations

Automatic evaluation

- Reference Translation
  - the gunman was shot to death by the police .
- System Translations
  - **the gunman was police kill .**
  - **wounded police jaya of**
  - **the gunman was shot dead by the police .**
  - **the gunman arrested by police kill .**
  - **the gunmen were killed .**
  - **the gunman was shot to death by the police .**
  - **gunmen were killed by police ?SUB>0 ?SUB>0**
  - **al by the police .**
  - **the ringer is killed by the police .**
  - **police killed the gunman .**
- Matches
  - green = 4 gram match (good!)
  - red = word not matched (bad!)

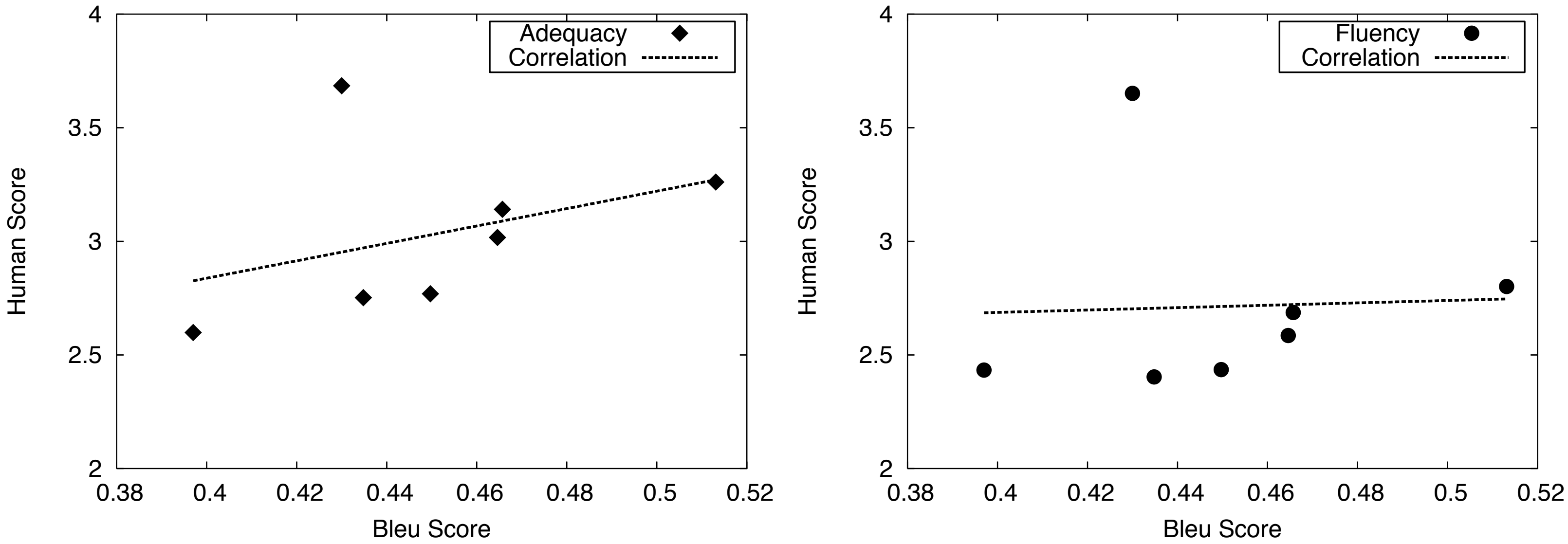
Automatic evaluation



[from George Doddington, NIST]

- BLEU **correlates** with human judgement
  - **multiple reference translations** may be used

Correlation? [Callison-Burch et al., 2006]

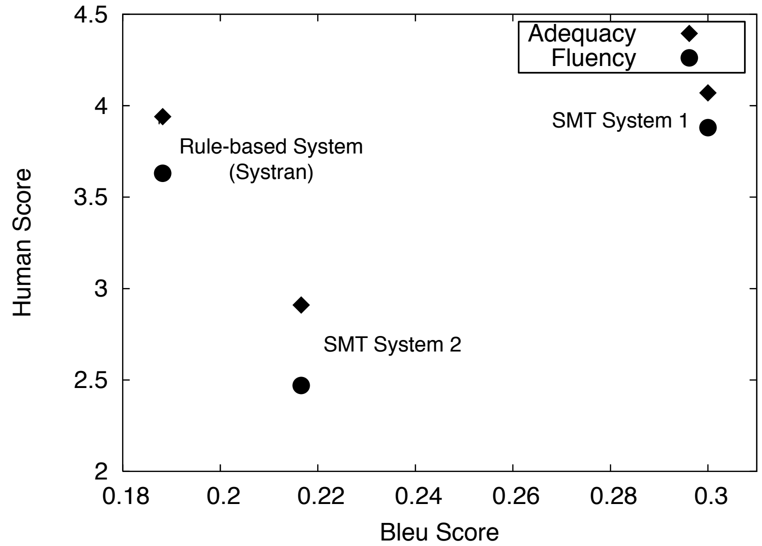


[from Callison-Burch et al., 2006, EACL]

- DARPA/NIST MT Eval 2005
  - Mostly statistical systems (all but one in graphs)
  - One submission **manual post-edit** of statistical system's output

→ Good adequacy/fluency scores *not reflected* by BLEU

Correlation? [Callison-Burch et al., 2006]



[from Callison-Burch et al., 2006, EACL]

- Comparison of
  - *good statistical* system: high BLEU, high adequacy/fluency
  - *bad statistical* sys. (trained on less data): low BLEU, low adequacy/fluency
  - *Systran*: lowest BLEU score, but high adequacy/fluency

Automatic evaluation: outlook

- Research questions
  - why does BLEU *fail* Systran and manual post-edits?
  - how can this *overcome* with novel evaluation metrics?

- Future of automatic methods
  - automatic metrics too *useful* to be abandoned
  - evidence still supports that during *system development*, a better BLEU indicates a better system
  - *final assessment* has to be human judgement

Competitions

- Progress driven by **MT Competitions**
  - **NIST/DARPA**: Yearly campaigns for Arabic-English, Chinese-English, newstexts, since 2001
  - **IWSLT**: Yearly competitions for Asian languages and Arabic into English, speech travel domain, since 2003
  - **WPT/WMT**: Yearly competitions for European languages, European Parliament proceedings, since 2005

- Increasing number of statistical MT groups participate

Euromatrix

- Proceedings of the European Parliament
  - translated into *11 official languages*
  - entry of new members in May 2004: more to come...

- Europarl corpus
  - collected 20-30 million words per language

  → *110 language pairs*

- 110 Translation systems
  - 3 weeks on 16-node cluster computer

  → *110 translation systems*

Quality of translation systems

- *Scores* for all 110 systems `http://www.statmt.org/matrix/`

	da	de	el	en	es	fr	fi	it	nl	pt	sv
da	-	**18.4**	**21.1**	**28.5**	**26.4**	**28.7**	**14.2**	**22.2**	**21.4**	**24.3**	**28.3**
de	**22.3**	-	**20.7**	**25.3**	**25.4**	**27.7**	**11.8**	**21.3**	**23.4**	**23.2**	**20.5**
el	**22.7**	**17.4**	-	**27.2**	**31.2**	**32.1**	**11.4**	**26.8**	**20.0**	**27.6**	**21.2**
en	**25.2**	**17.6**	**23.2**	-	**30.1**	**31.1**	**13.0**	**25.3**	**21.0**	**27.1**	**24.8**
es	**24.1**	**18.2**	**28.3**	**30.5**	-	**40.2**	**12.5**	**32.3**	**21.4**	**35.9**	**23.9**
fr	**23.7**	**18.5**	**26.1**	**30.0**	**38.4**	-	**12.6**	**32.4**	**21.1**	**35.3**	**22.6**
fi	**20.0**	**14.5**	**18.2**	**21.8**	**21.1**	**22.4**	-	**18.3**	**17.0**	**19.1**	**18.8**
it	**21.4**	**16.9**	**24.8**	**27.8**	**34.0**	**36.0**	**11.0**	-	**20.0**	**31.2**	**20.2**
nl	**20.5**	**18.3**	**17.4**	**23.0**	**22.9**	**24.6**	**10.3**	**20.0**	-	**20.7**	**19.0**
pt	**23.2**	**18.2**	**26.4**	**30.1**	**37.9**	**39.0**	**11.9**	**32.0**	**20.2**	-	**21.9**
sv	**30.3**	**18.9**	**22.8**	**30.2**	**28.6**	**29.7**	**15.3**	**23.9**	**21.9**	**25.9**	-

[from Koehn, 2005: Europarl]

What makes MT difficult?

- Some language pairs more difficult than others

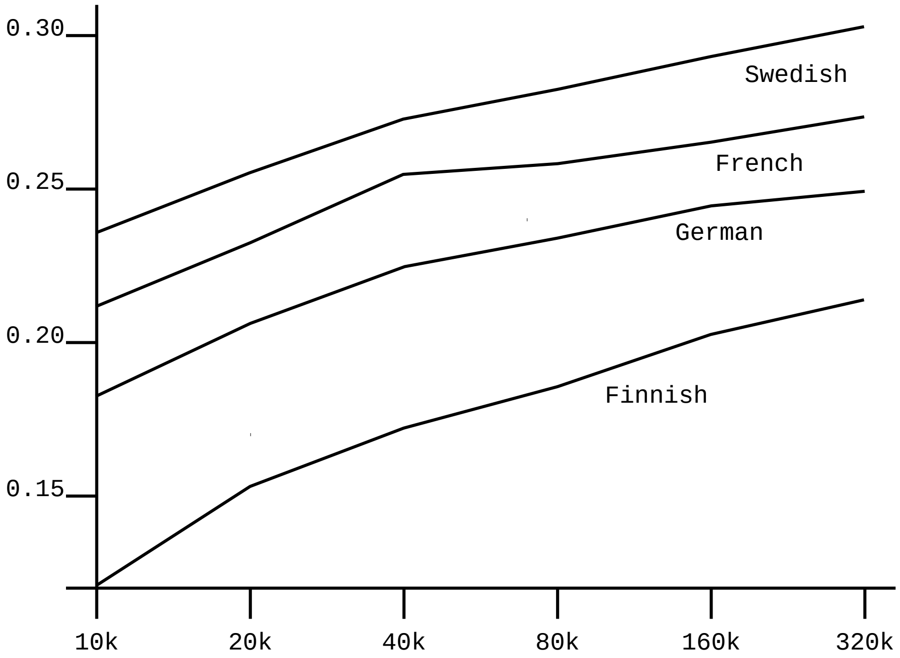
- Birch et al [EMNLP 2008] showed 75% of the differences in BLEU scores due to
  - morphology on target side (vocabulary size)
  - historic distance of languages (cognate ratio)
  - degree of reordering requited

- Not a factor: morphology on source
  - note: Arabic–English fairly good, despite rich morphology in Arabic

Available data

- Available *parallel text*
  - **Europarl**: *40 million words* in 11 languages http://www.statmt.org/europarl/
  - **Acquis Communitaire**: *8-50 million words* in 20 EU languages
  - **Canadian Hansards**: *20 million words* from Ulrich Germann, ISI
  - Chinese/Arabic to English: *over 100 million words* from **LDC**
  - lots more French/English, Spanish/French/English from **LDC**
- Available monolingual text (for language modeling)
  - *2.8 billion words* of English from **LDC**
  - *trillions of words* on the web

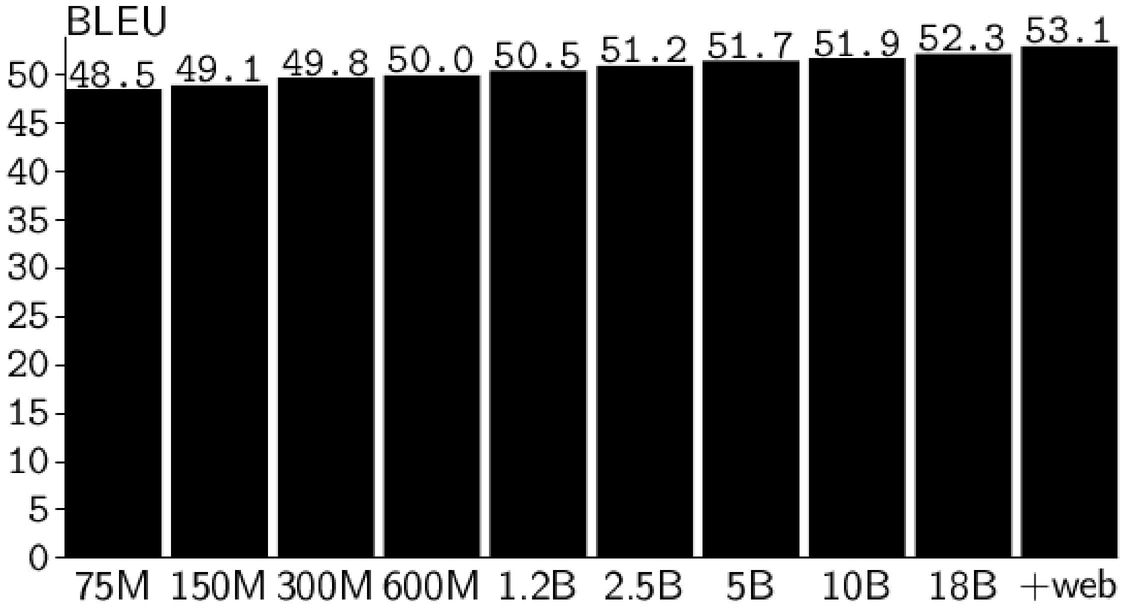
More data, better translations



[from Koehn, 2003: Europarl]

- **Log-scale improvements** on BLEU:
  Doubling the training data gives constant improvement *(+1 %BLEU)*

More LM data, better translations



[from Och, 2005: MT Eval presentation]

- Also **log-scale improvements** on BLEU:
  doubling the training data gives constant improvement *(+0.5 %BLEU)*
  (last addition is 218 billion words out-of-domain web data)

Word-based models and the EM algorithm

Lexical translation

- How to translate a word → look up in dictionary

  Haus — *house, building, home, household, shell.*

- *Multiple translations*
  - some more frequent than others
  - for instance: *house*, and *building* most common
  - special cases: *Haus* of a *snail* is its *shell*

- Note: During all the lectures, we will translate from a foreign language into English

Collect statistics

- Look at a *parallel corpus* (German text along with English translation)

Translation of *Haus*	**Count**
house	8,000
building	1,600
home	200
household	150
shell	50

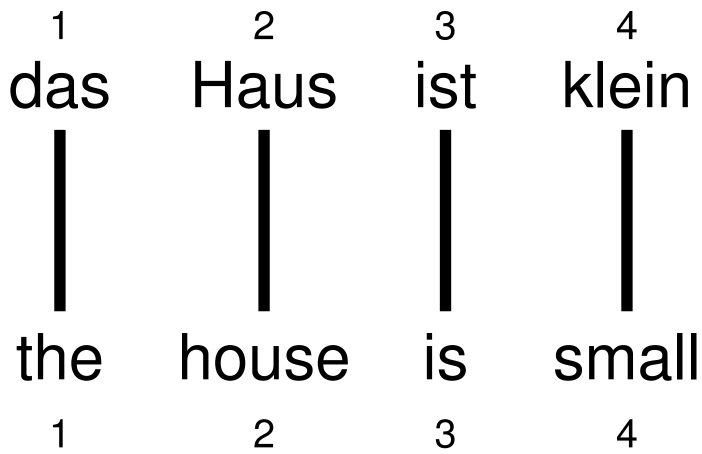
Estimate translation probabilities

- *Maximum likelihood estimation*

$$p_f(e) = \begin{cases} 0.8 & \text{if } e = \textit{house}, \\ 0.16 & \text{if } e = \textit{building}, \\ 0.02 & \text{if } e = \textit{home}, \\ 0.015 & \text{if } e = \textit{household}, \\ 0.005 & \text{if } e = \textit{shell}. \end{cases}$$

Alignment

- In a parallel text (or when we translate), we **align** words in one language with the words in the other



- Word *positions* are numbered 1–4

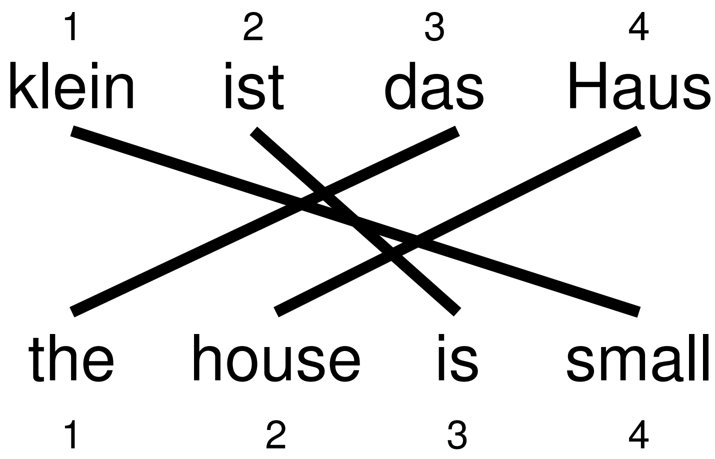
Alignment function

- Formalizing *alignment* with an **alignment function**
- Mapping an English target word at position i to a German source word at position j with a function $a : i \rightarrow j$
- Example

$$a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 4\}$$

Reordering

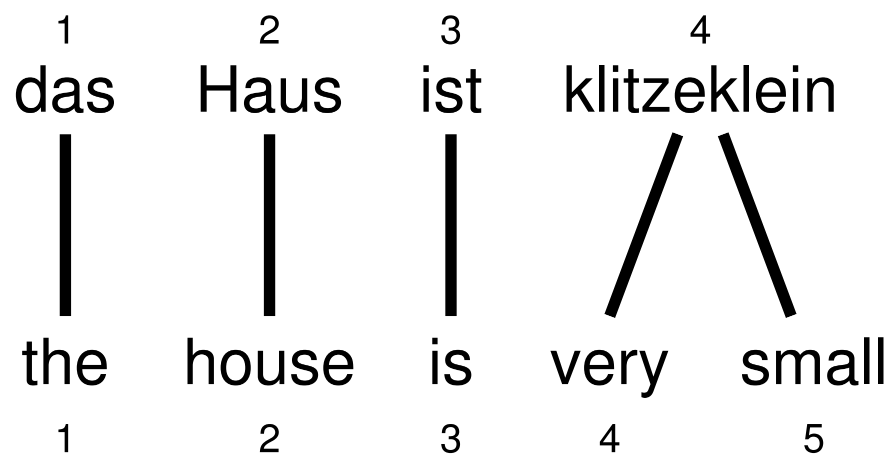
- Words may be **reordered** during translation



$$a : \{1 \rightarrow 3, 2 \rightarrow 4, 3 \rightarrow 2, 4 \rightarrow 1\}$$

One-to-many translation

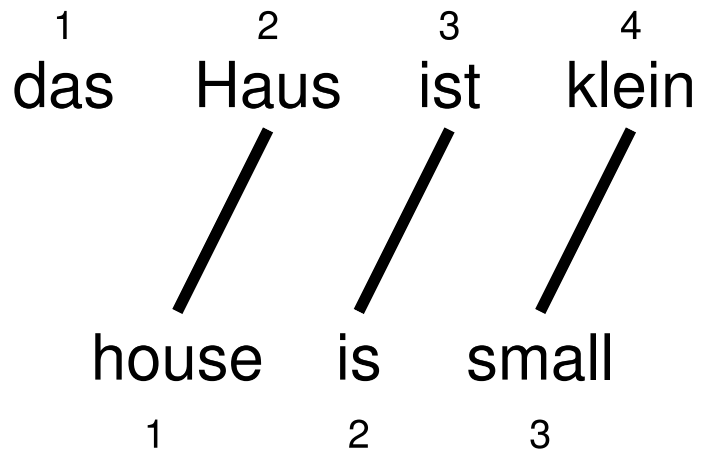
- A source word may translate into **multiple** target words



$$a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 4, 5 \rightarrow 4\}$$

Dropping words

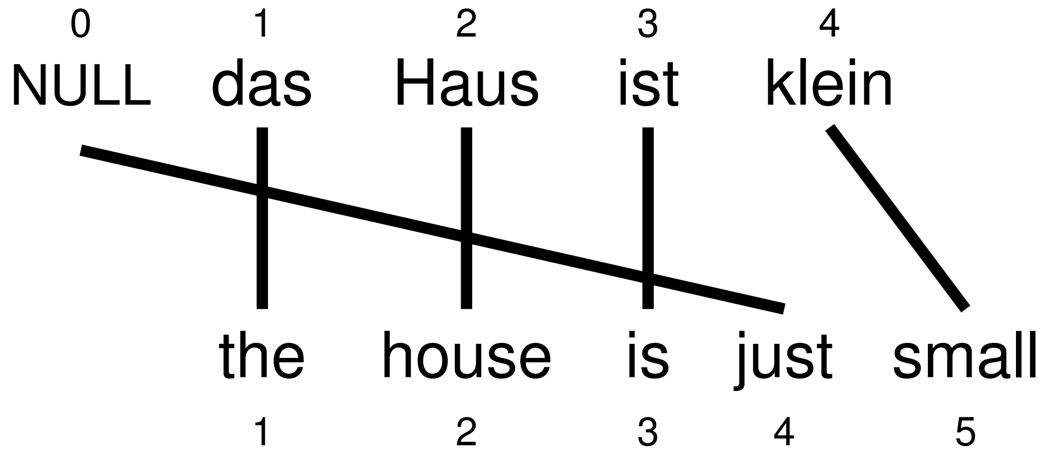
- Words may be **dropped** when translated
  - The German article *das* is dropped



$$a : \{1 \rightarrow 2, 2 \rightarrow 3, 3 \rightarrow 4\}$$

Inserting words

- Words may be **added** during translation
  - The English *just* does not have an equivalent in German
  - We still need to map it to something: special NULL token



$$a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 0, 5 \rightarrow 4\}$$

IBM Model 1

- *Generative model*: break up translation process into smaller steps
  - **IBM Model 1** only uses *lexical translation*

- Translation probability
  - for a foreign sentence $\mathbf{f} = (f_1, ..., f_{l_f})$ of length l_f
  - to an English sentence $\mathbf{e} = (e_1, ..., e_{l_e})$ of length l_e
  - with an alignment of each English word e_j to a foreign word f_i according to the alignment function $a : j \to i$

$$p(\mathbf{e}, a|\mathbf{f}) = \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|f_{a(j)})$$

  - parameter ϵ is a *normalization constant*

Example

das

e	$t(e\|f)$
the	0.7
that	0.15
which	0.075
who	0.05
this	0.025

Haus

e	$t(e\|f)$
house	0.8
building	0.16
home	0.02
household	0.015
shell	0.005

ist

e	$t(e\|f)$
is	0.8
's	0.16
exists	0.02
has	0.015
are	0.005

klein

e	$t(e\|f)$
small	0.4
little	0.4
short	0.1
minor	0.06
petty	0.04

$$\begin{aligned} p(e,a|f) &= \frac{\epsilon}{4^3} \times t(\text{the}|\text{das}) \times t(\text{house}|\text{Haus}) \times t(\text{is}|\text{ist}) \times t(\text{small}|\text{klein}) \\ &= \frac{\epsilon}{4^3} \times 0.7 \times 0.8 \times 0.8 \times 0.4 \\ &= 0.0028\epsilon \end{aligned}$$

Learning lexical translation models

- We would like to *estimate* the lexical translation probabilities $t(e|f)$ from a parallel corpus

- ... but we do not have the alignments

- **Chicken and egg problem**
  - if we had the *alignments*,
    $\rightarrow$ we could estimate the *parameters* of our generative model
  - if we had the *parameters*,
    $\rightarrow$ we could estimate the *alignments*

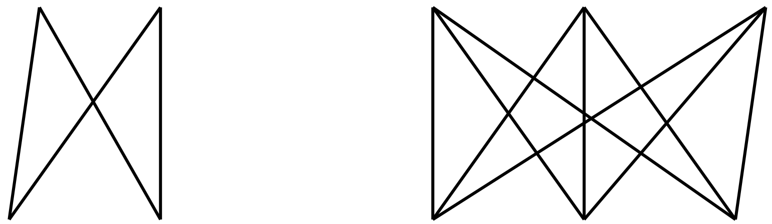
EM algorithm

- **Incomplete data**
  - if we had *complete data*, would could estimate *model*
  - if we had *model*, we could fill in the *gaps in the data*

- **Expectation Maximization (EM)** in a nutshell
  - initialize model parameters (e.g. uniform)
  - assign probabilities to the missing data
  - estimate model parameters from completed data
  - iterate

EM algorithm

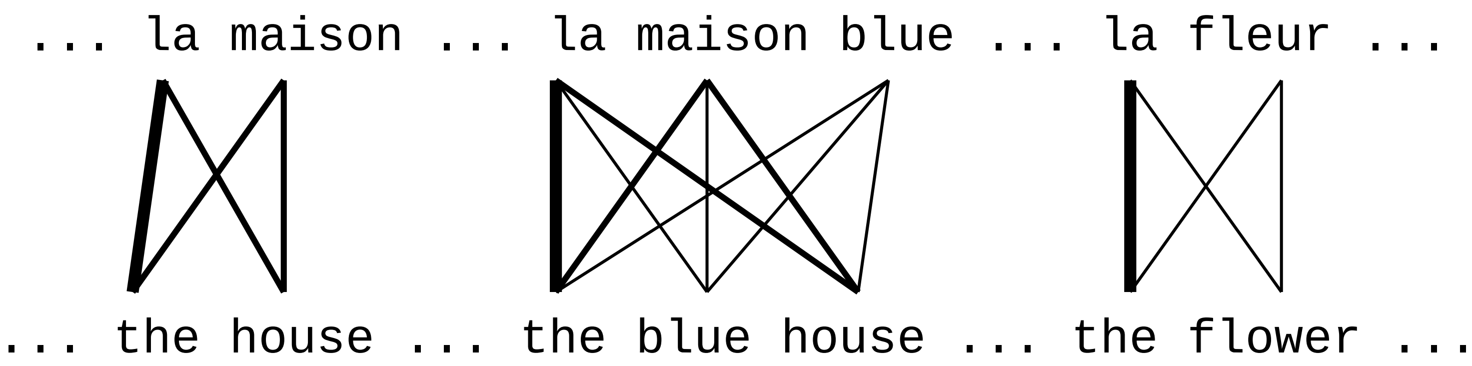
... la maison ... la maison blue ... la fleur ...



... the house ... the blue house ... the flower ...

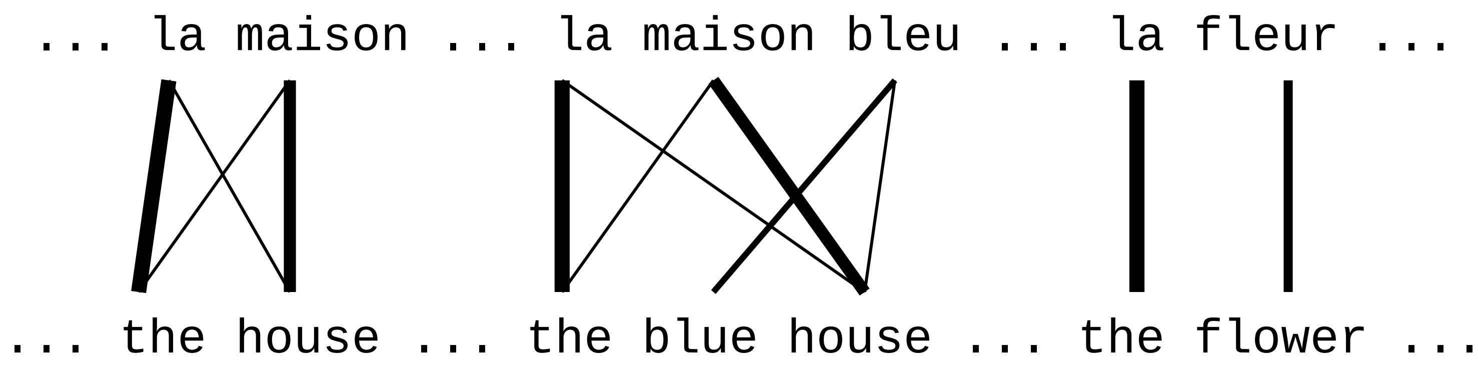
- Initial step: all alignments equally likely
- Model learns that, e.g., *la* is often aligned with *the*

EM algorithm



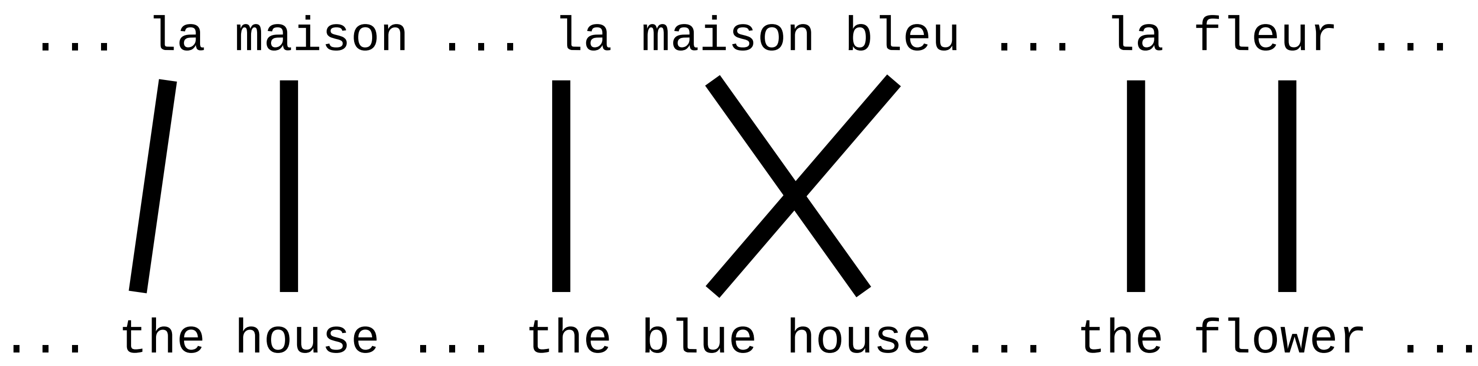
- After one iteration
- Alignments, e.g., between *la* and *the* are more likely

EM algorithm



- After another iteration
- It becomes apparent that alignments, e.g., between *fleur* and *flower* are more likely (**pigeon hole principle**)

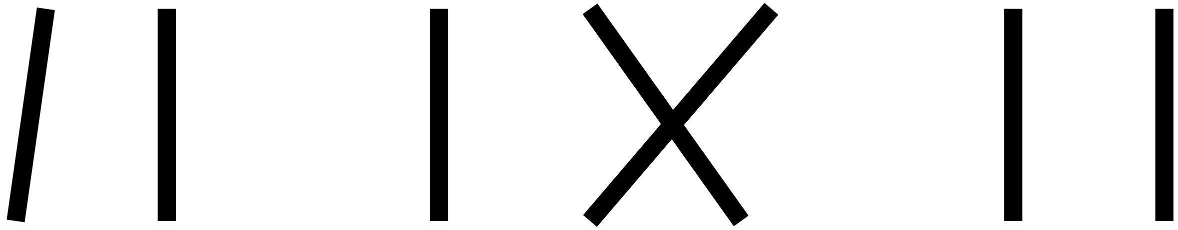
EM algorithm



- Convergence
- Inherent hidden structure revealed by EM

EM algorithm

... la maison ... la maison bleu ... la fleur ...



... the house ... the blue house ... the flower ...

p(la|the) = 0.453
p(le|the) = 0.334
p(maison|house) = 0.876
p(bleu|blue) = 0.563
...

- Parameter estimation from the aligned corpus

IBM Model 1 and EM

- EM Algorithm consists of two steps
- **Expectation-Step**: Apply model to the data
  - parts of the model are hidden (here: alignments)
  - using the model, assign probabilities to possible values
- **Maximization-Step**: Estimate model from data
  - take assign values as fact
  - collect counts (weighted by probabilities)
  - estimate model from counts
- Iterate these steps until **convergence**

IBM Model 1 and EM

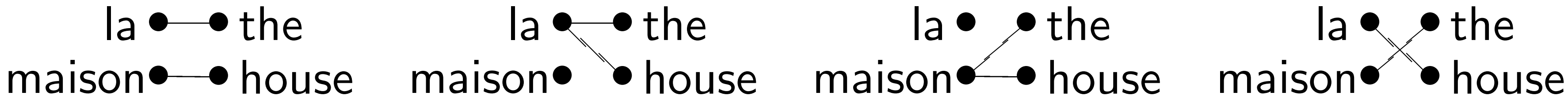
- We need to be able to compute:
  - Expectation-Step: probability of alignments
  - Maximization-Step: count collection

IBM Model 1 and EM

- **Probabilities**

$$p(\text{the}|\text{la}) = 0.7 \qquad p(\text{house}|\text{la}) = 0.05$$
$$p(\text{the}|\text{maison}) = 0.1 \qquad p(\text{house}|\text{maison}) = 0.8$$

- **Alignments**



$$p(\mathbf{e}, a|\mathbf{f}) = 0.56 \qquad p(\mathbf{e}, a|\mathbf{f}) = 0.035 \qquad p(\mathbf{e}, a|\mathbf{f}) = 0.08 \qquad p(\mathbf{e}, a|\mathbf{f}) = 0.005$$

$$p(a|\mathbf{e}, \mathbf{f}) = 0.824 \qquad p(a|\mathbf{e}, \mathbf{f}) = 0.052 \qquad p(a|\mathbf{e}, \mathbf{f}) = 0.118 \qquad p(a|\mathbf{e}, \mathbf{f}) = 0.007$$

- **Counts**

$$c(\text{the}|\text{la}) = 0.824 + 0.052 \qquad c(\text{house}|\text{la}) = 0.052 + 0.007$$
$$c(\text{the}|\text{maison}) = 0.118 + 0.007 \qquad c(\text{house}|\text{maison}) = 0.824 + 0.118$$

Higher IBM Models

IBM Model 1	lexical translation
IBM Model 2	adds absolute **reordering model**
IBM Model 3	adds **fertility model**
IBM Model 4	relative reordering model
IBM Model 5	fixes **deficiency**

- Only IBM Model 1 has *global maximum*
  - training of a higher IBM model builds on previous model
- Compuationally biggest change in Model 3
  - trick to simplify estimation does not work anymore
  - $\rightarrow$ *exhaustive* count collection becomes computationally too expensive
  - **sampling** over high probability alignments is used instead

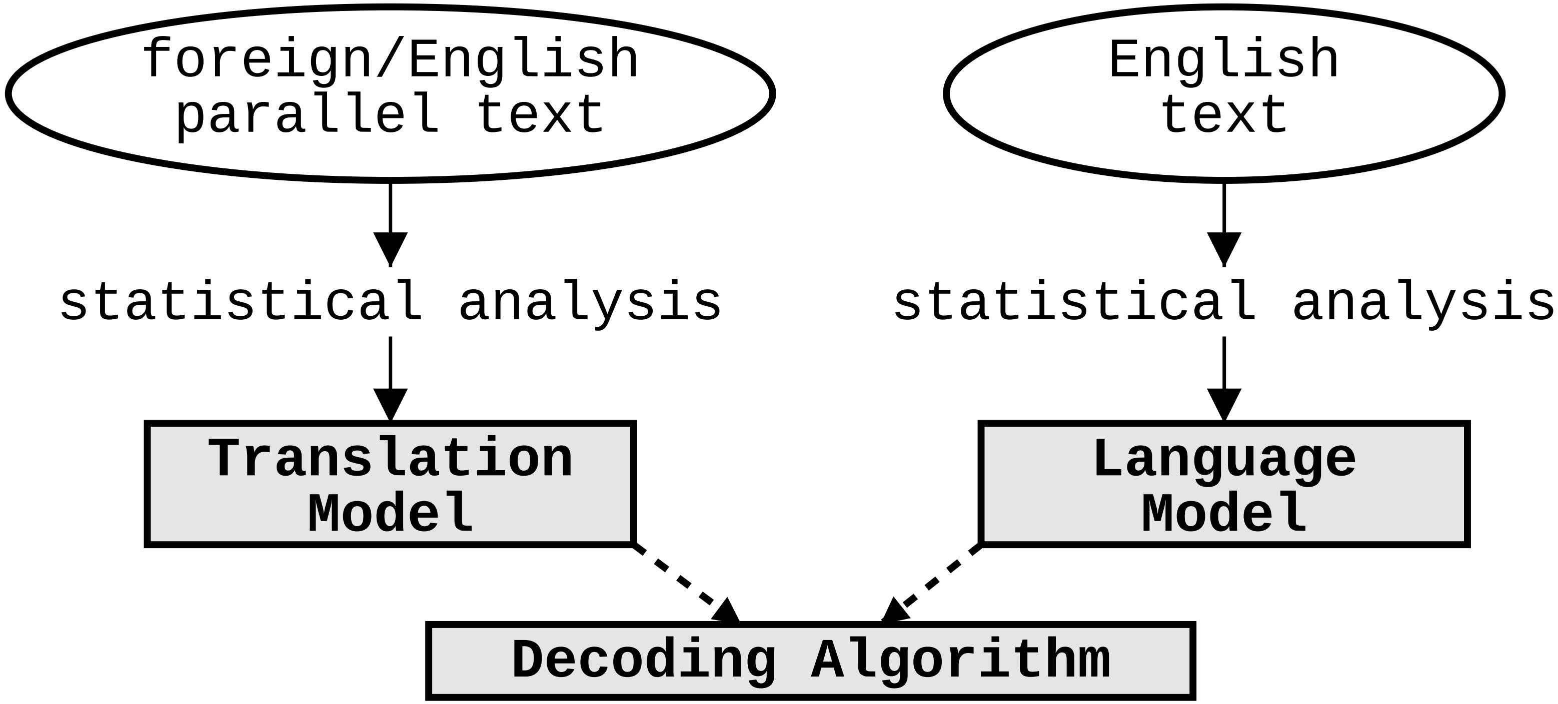
IBM Model 4


Mary did not slap the green witch
n(3|slap)
Mary not slap slap slap the green witch
p-null
Mary not slap slap slap NULL the green witch
t(la|the)
Maria no daba una botefada a la verde bruja
d(4|4)
Maria no daba una bofetada a la bruja verde

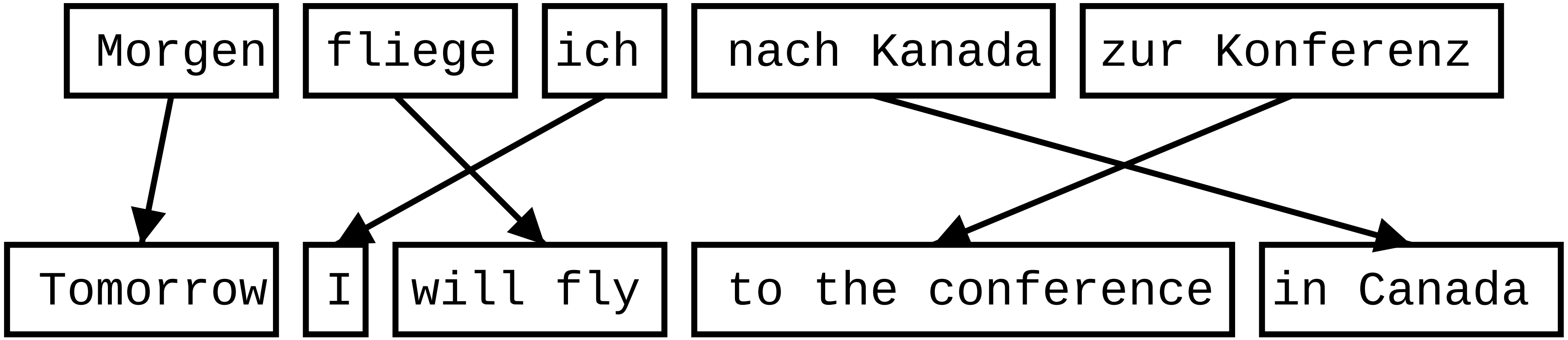
Decoding

Statistical Machine Translation

- Components: Translation model, language model, decoder



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

Phrase Translation Table

- Phrase Translations for "den Vorschlag":

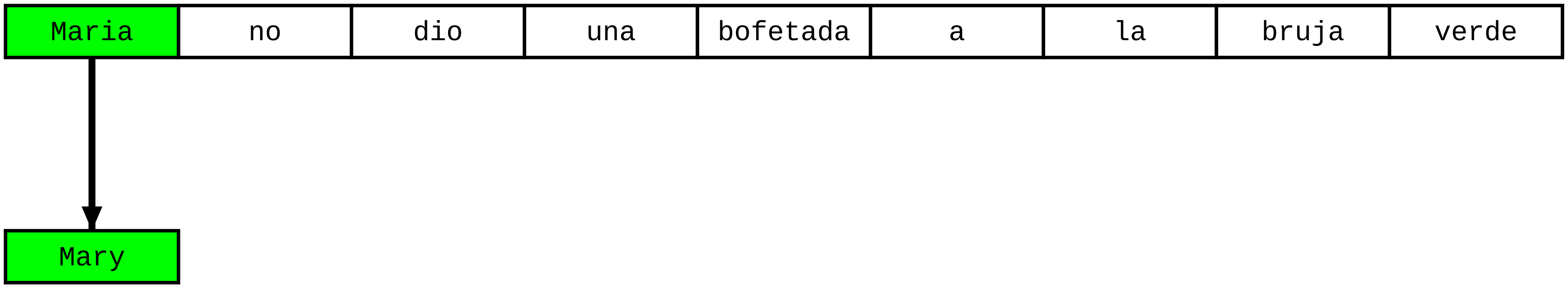
English	$\phi(e\|f)$	English	$\phi(e\|f)$
the proposal	0.6227	the suggestions	0.0114
's proposal	0.1068	the proposed	0.0114
a proposal	0.0341	the motion	0.0091
the idea	0.0250	the idea of	0.0091
this proposal	0.0227	the proposal ,	0.0068
proposal	0.0205	its proposal	0.0068
of the proposal	0.0159	it	0.0068
the proposals	0.0159	...	...

Decoding Process

Maria	no	dio	una	bofetada	a	la	bruja	verde

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

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

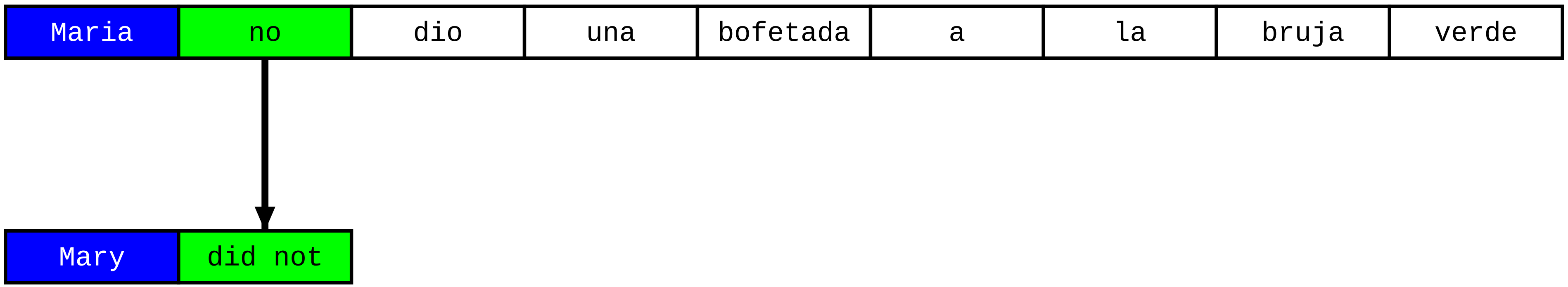
Decoding Process

Maria	no	dio	una	bofetada	a	la	bruja	verde

Mary

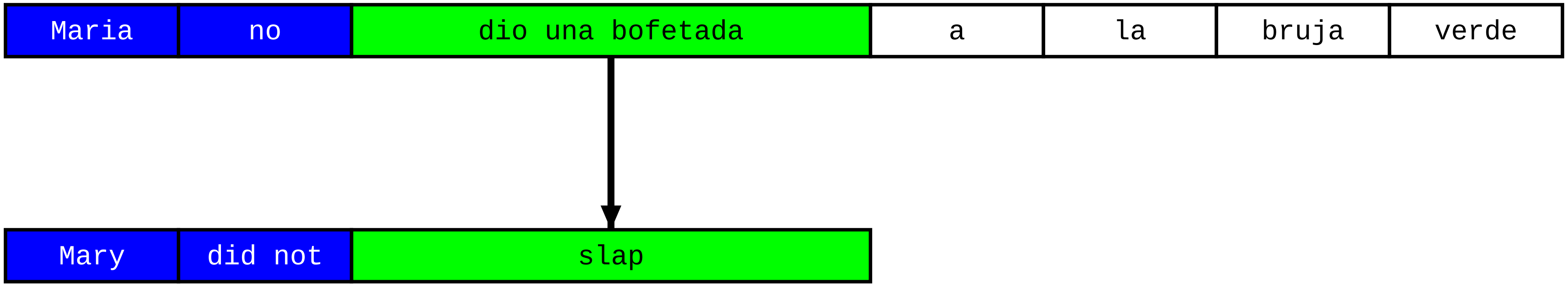
- Build translation left to right
  - select foreign words to be translated
  - find English phrase translation
  - add English phrase to end of partial translation
  - *mark foreign* words as translated

Decoding Process



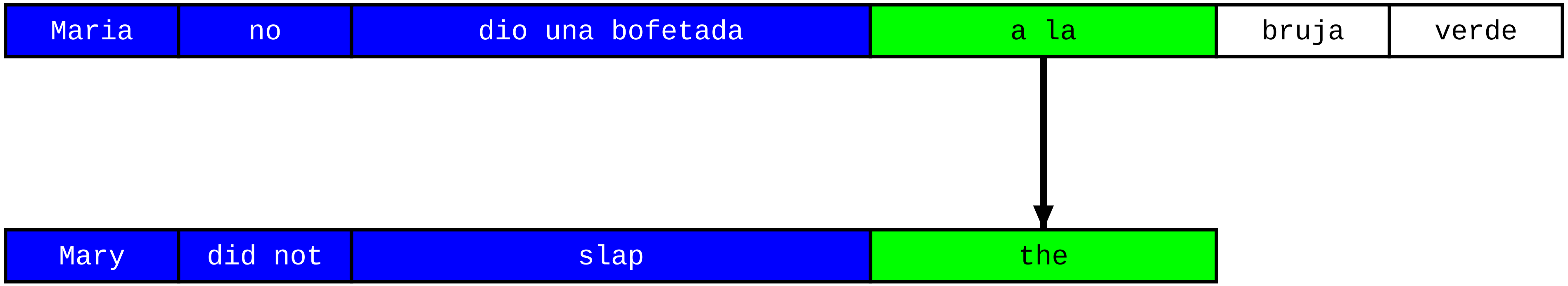
- *One to many* translation

Decoding Process



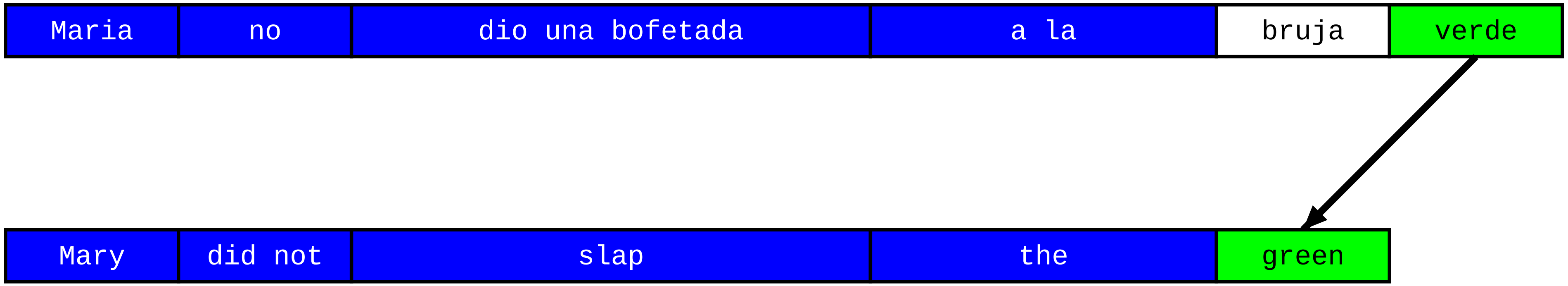
- Many to one translation

Decoding Process



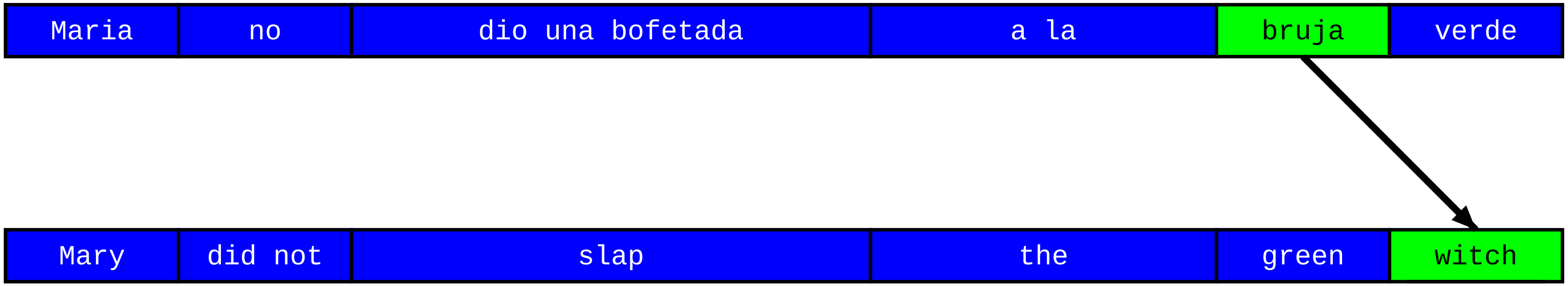
- *Many to one* translation

Decoding Process



- *Reordering*

Decoding Process



- Translation *finished*

Translation Options

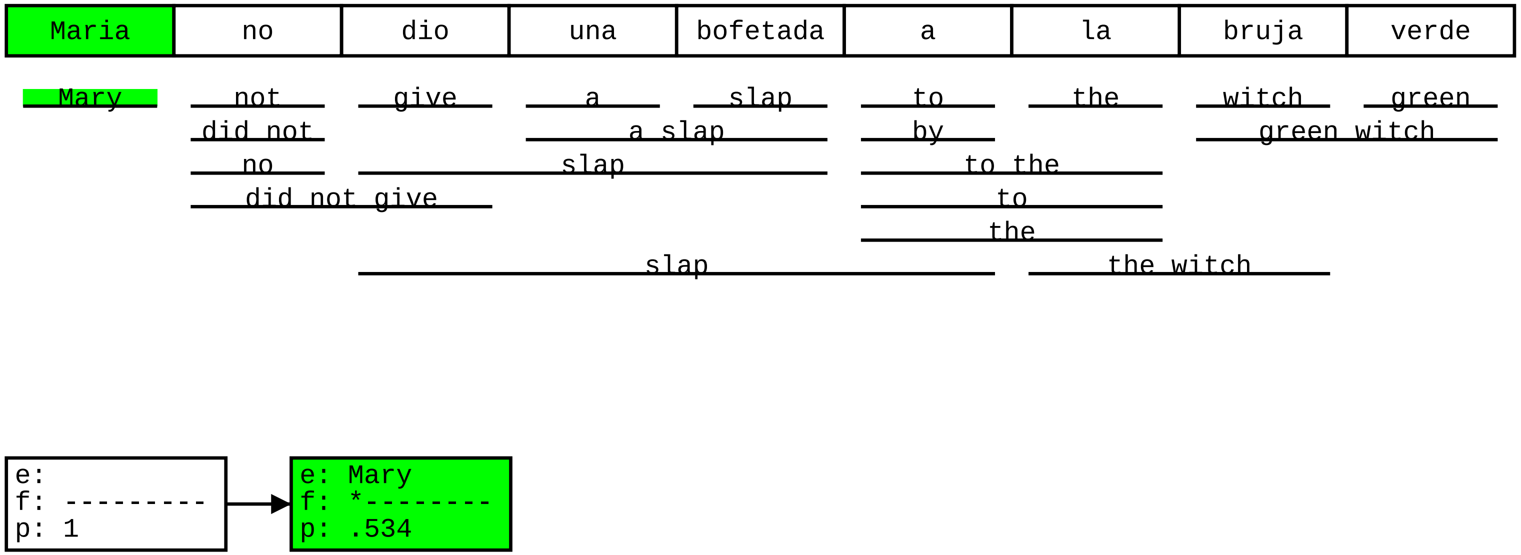
- Look up *possible phrase translations*
  - many different ways to *segment* words into phrases
  - many different ways to *translate* each phrase

Hypothesis Expansion


```
e:
f: ---------
p: 1
```

- 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

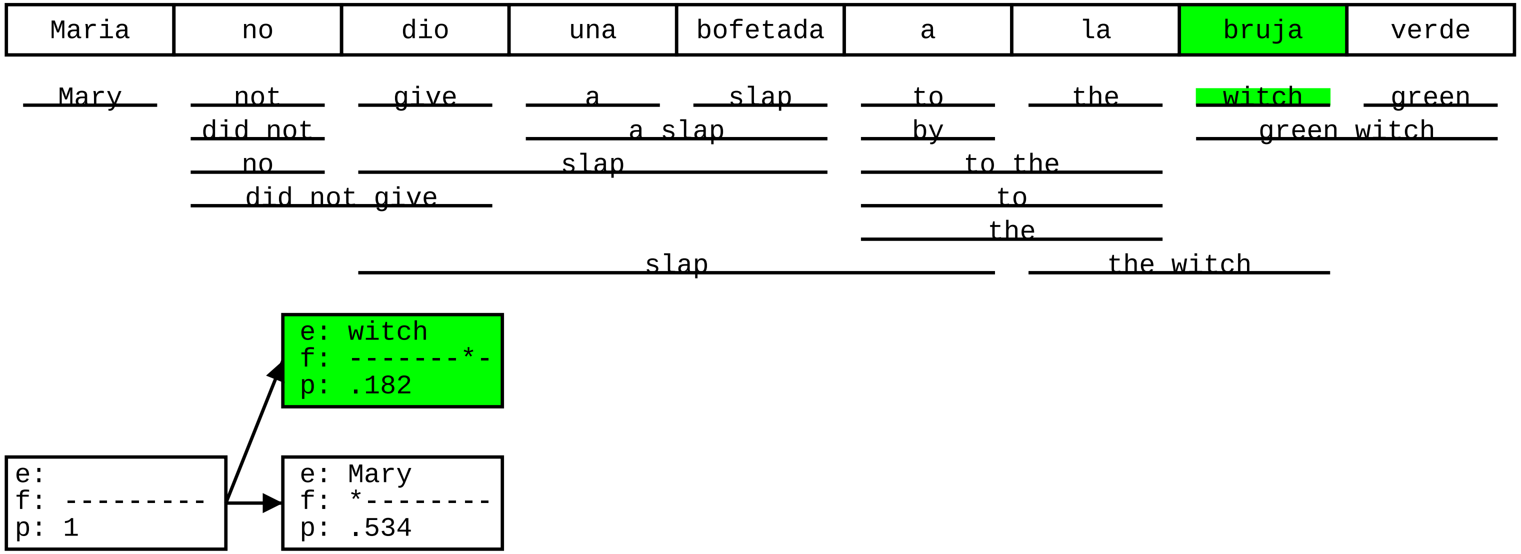
A Quick Word on Probabilities

- Not going into detail here, but...

- *Translation Model*
  - phrase translation probability p(Mary|Maria)
  - reordering costs
  - phrase/word count costs
  - ...

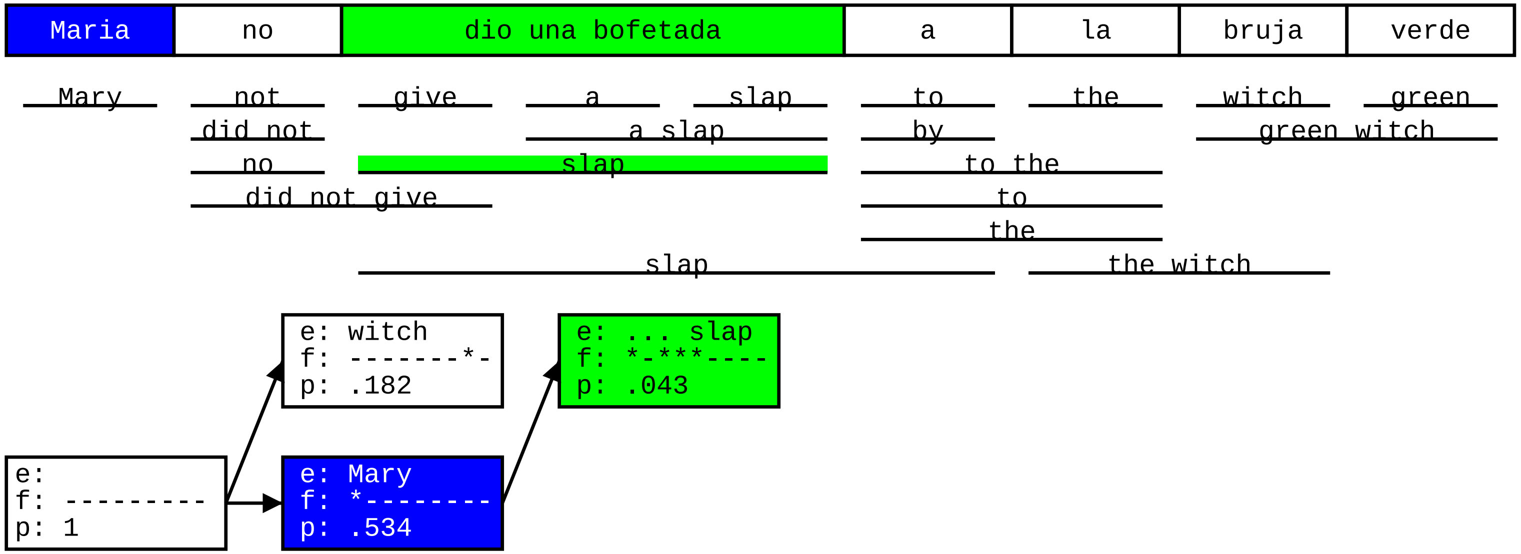
- *Language Model*
  - uses trigrams:
  - p(Mary did not) $=$
    p(Mary|START) $\times p$(did|Mary,START) $\times$ p(not|Mary did)

Hypothesis Expansion



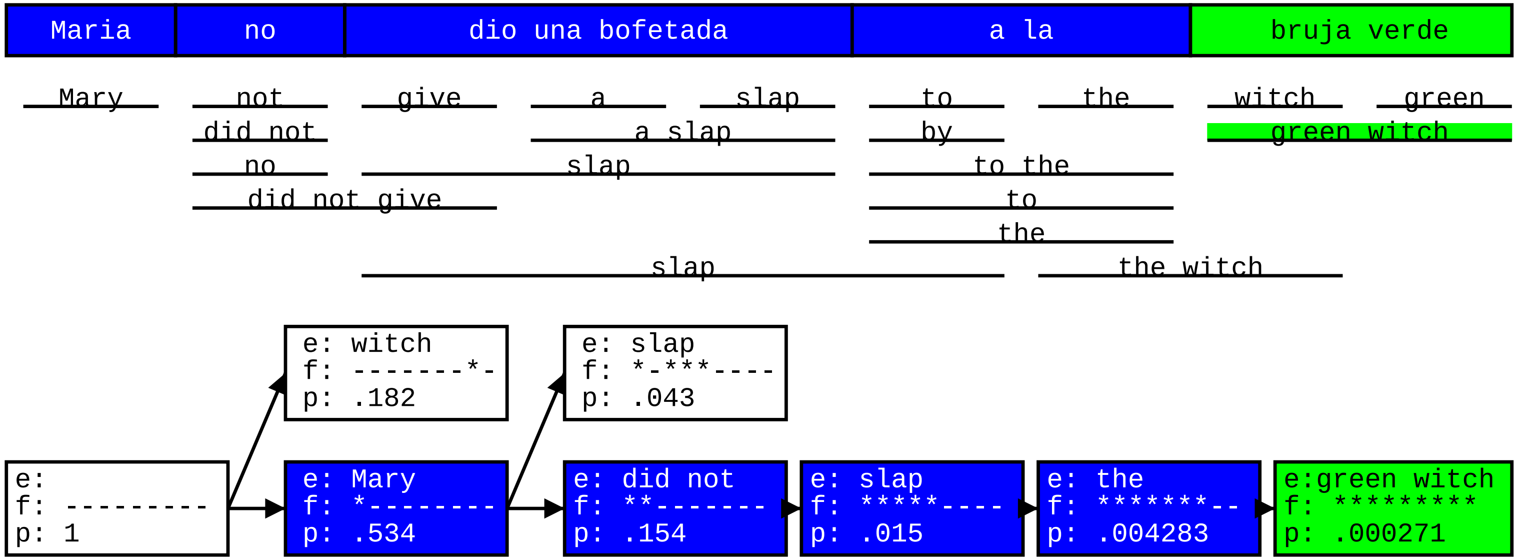
- Add another *hypothesis*

Hypothesis Expansion



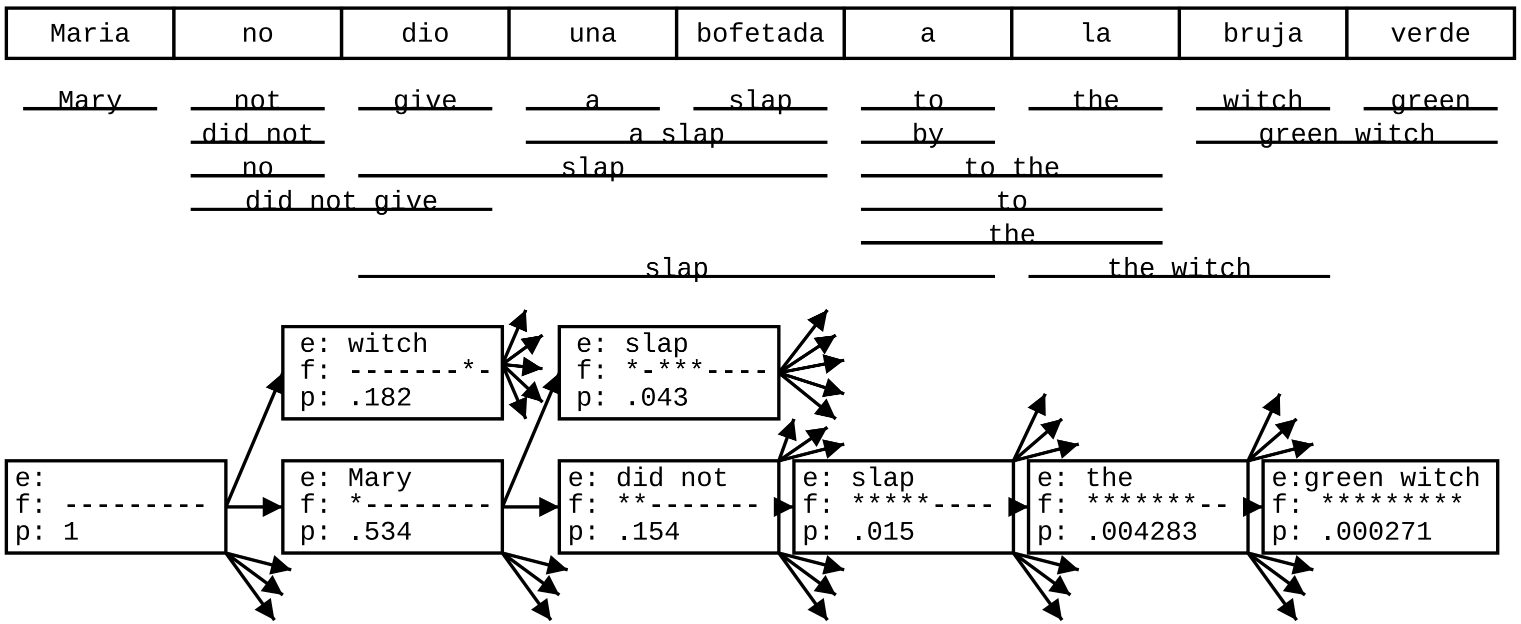
- Further *hypothesis expansion*

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

Explosion of Search Space

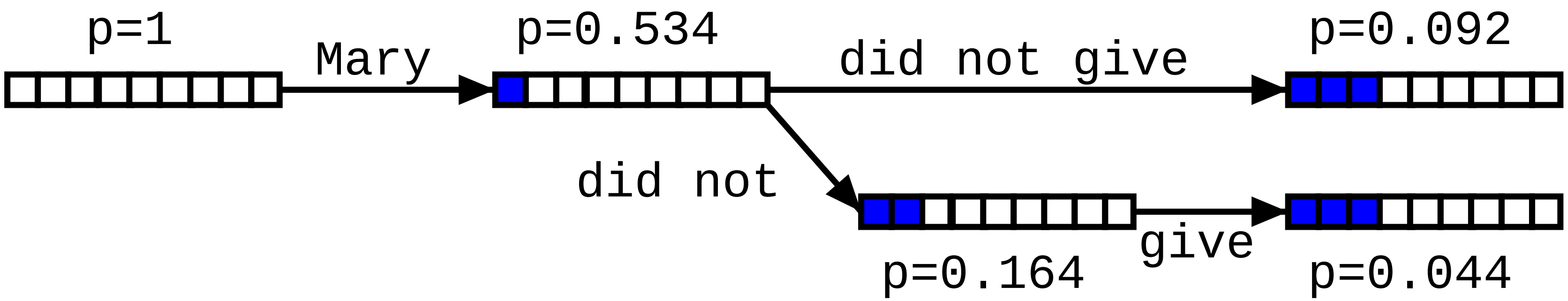
- Number of hypotheses is *exponential* with respect to sentence length

⇒ Decoding is NP-complete [Knight, 1999]

⇒ Need to *reduce search space*

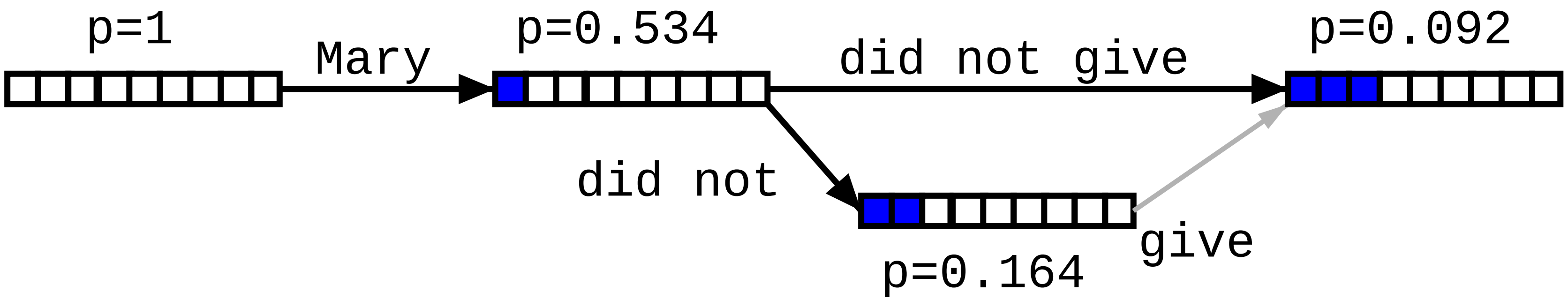
  - risk free: hypothesis **recombination**
  - risky: **histogram/threshold pruning**

Hypothesis Recombination



- Different paths to the *same* partial translation

Hypothesis Recombination

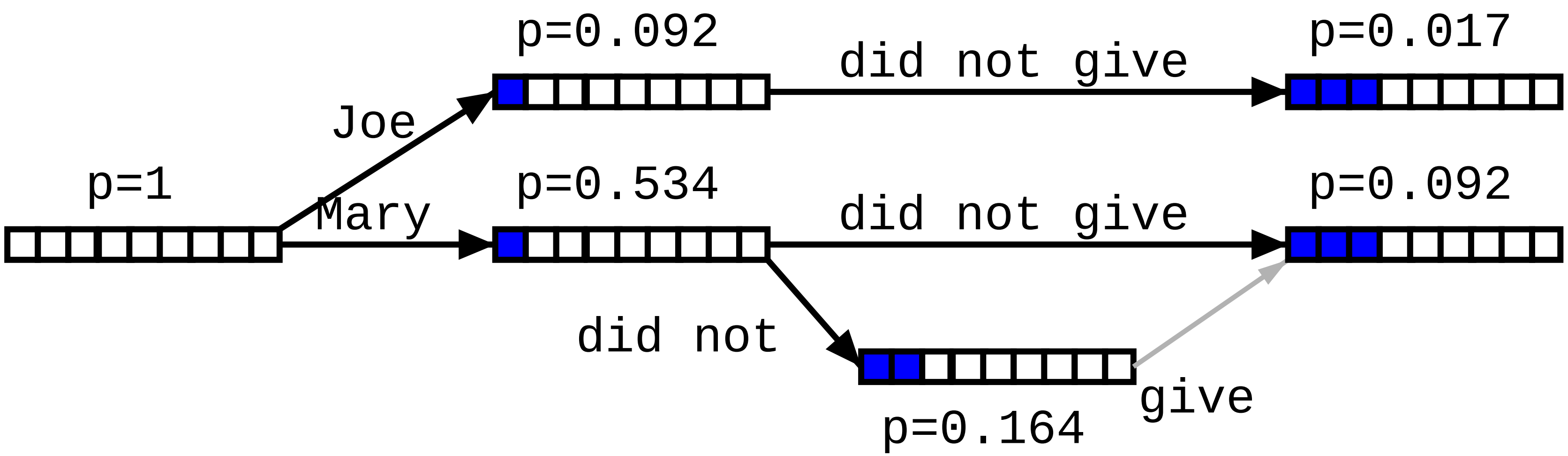


- Different paths to the same partial translation

⇒ *Combine paths*

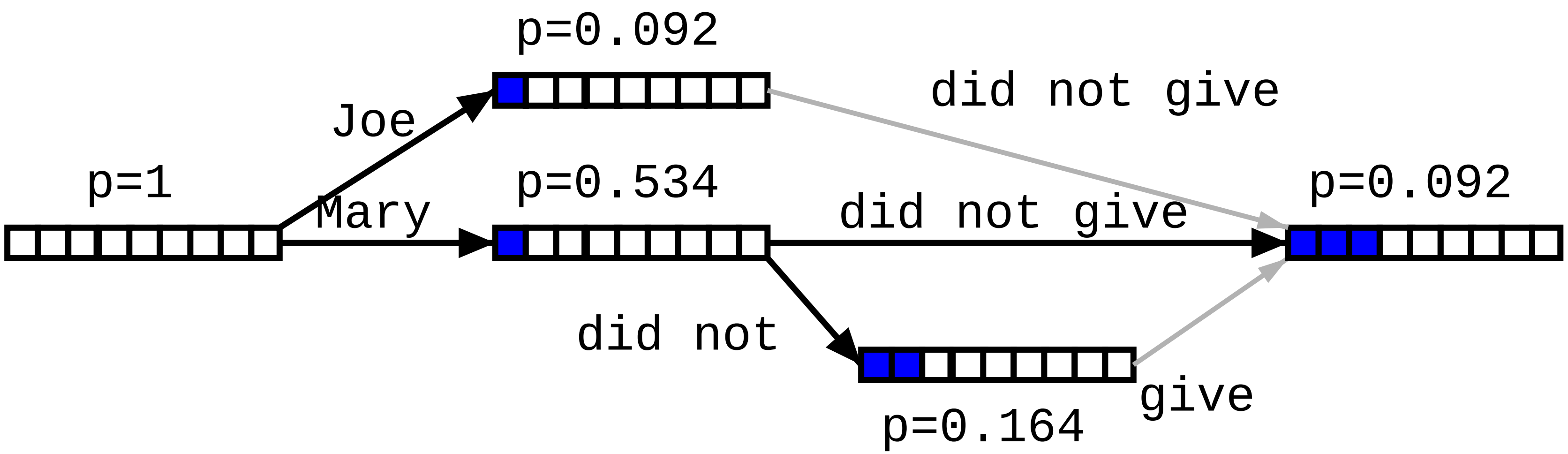
- *drop weaker* path
- keep pointer from weaker path (for lattice generation)

Hypothesis Recombination



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

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 (effects future path)

⇒ *Combine paths*

Pruning

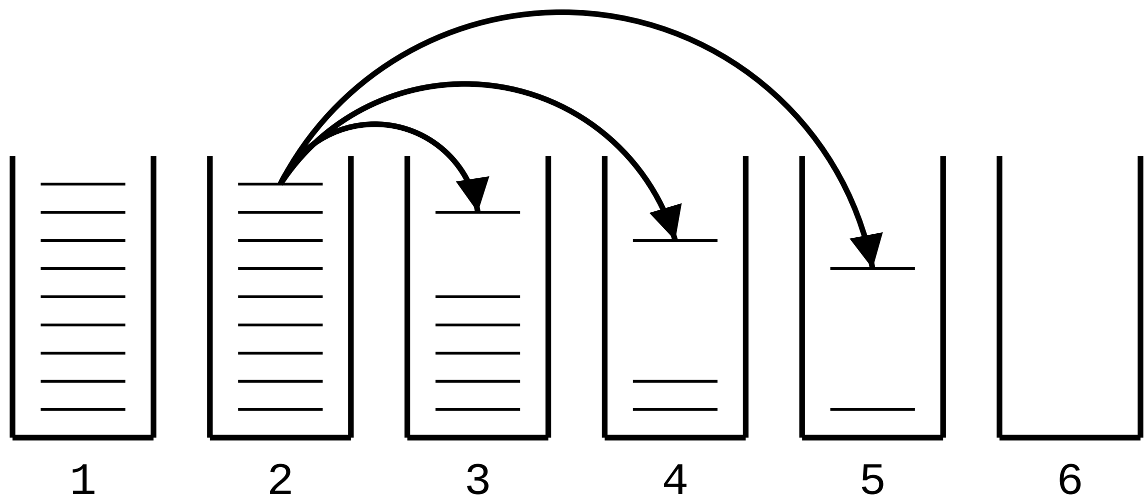
- Hypothesis recombination is *not sufficient*

⇒ 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., α = 0.001)

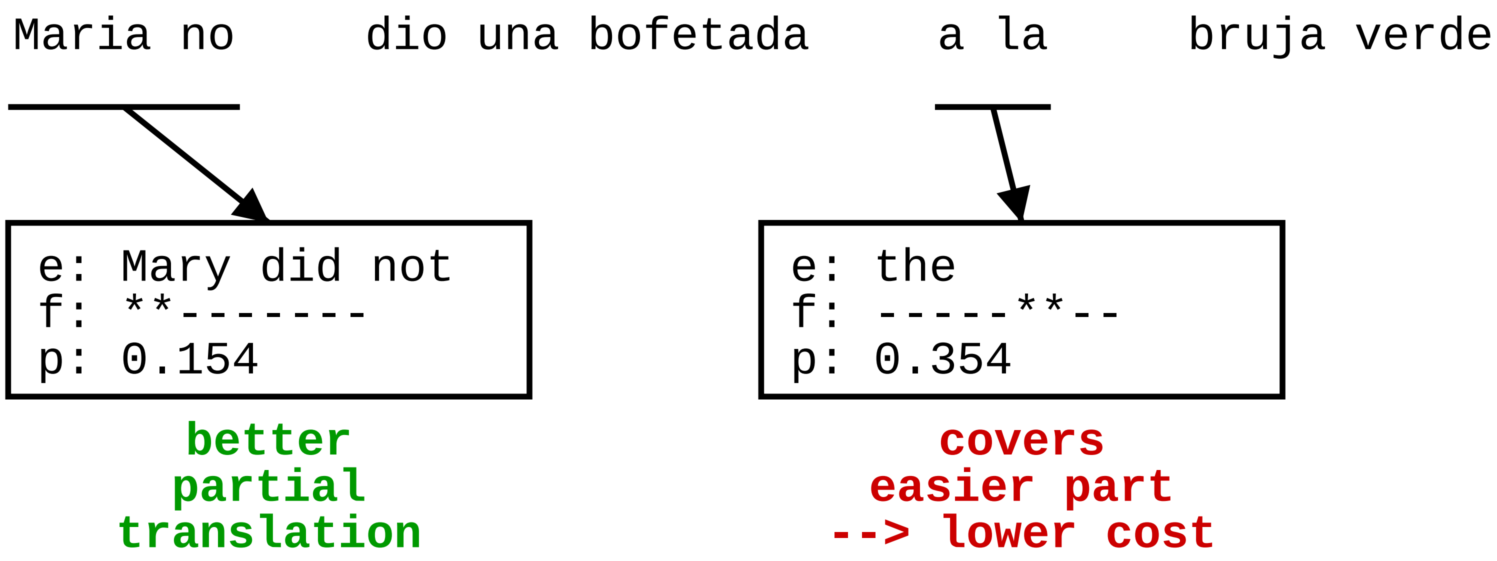
Hypothesis Stacks



- 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

Comparing Hypotheses

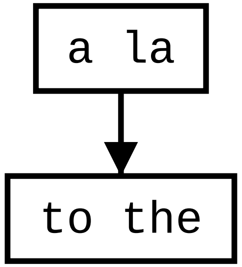
- Comparing hypotheses with *same number of foreign words* covered



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

⇒ Need to consider **future cost** of uncovered parts

Future Cost Estimation

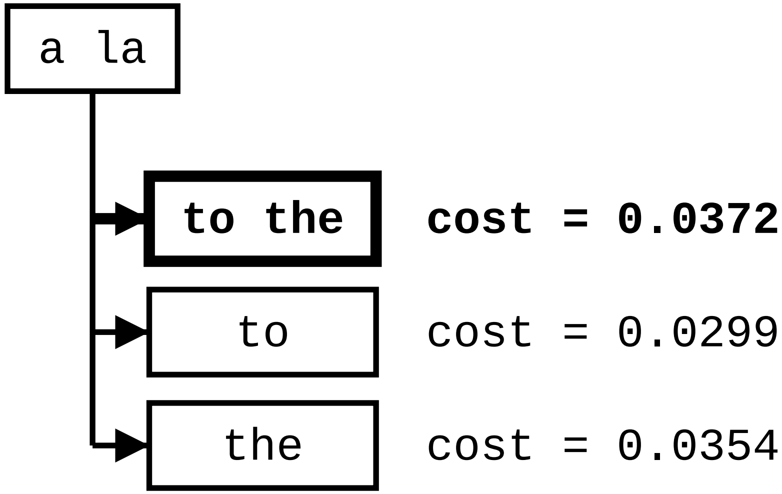


- *Estimate cost* to translate remaining part of input

- 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

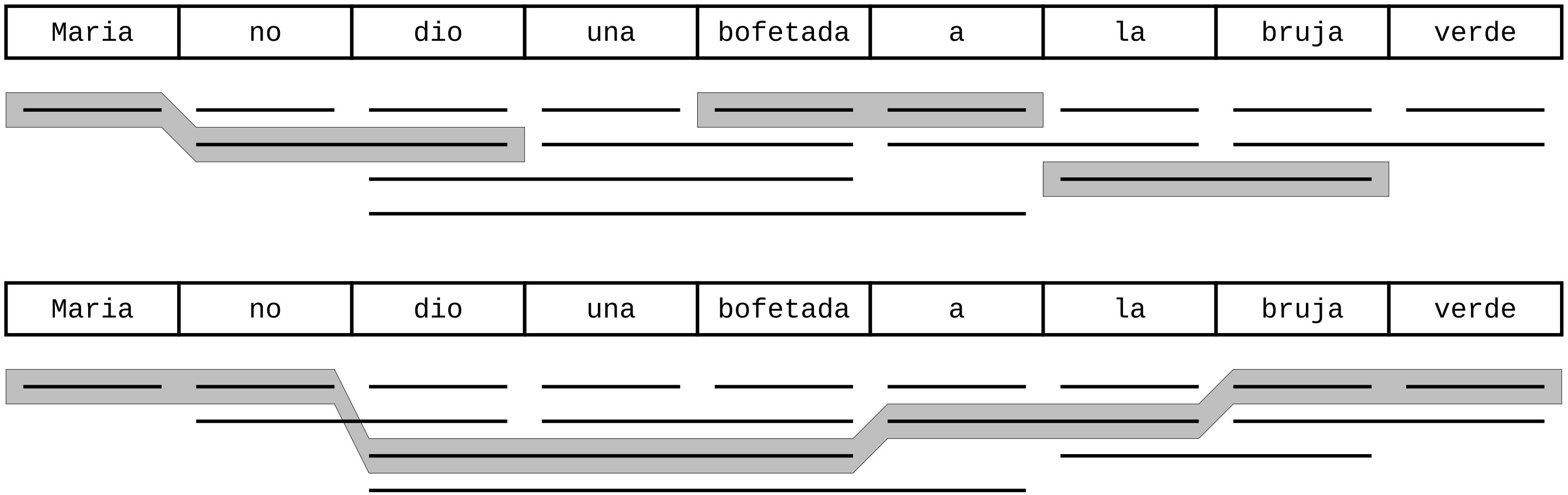
  $\rightarrow$ LM * TM = p(to) * p(the|to) * p(to the|a la)

Future Cost Estimation: Step 2



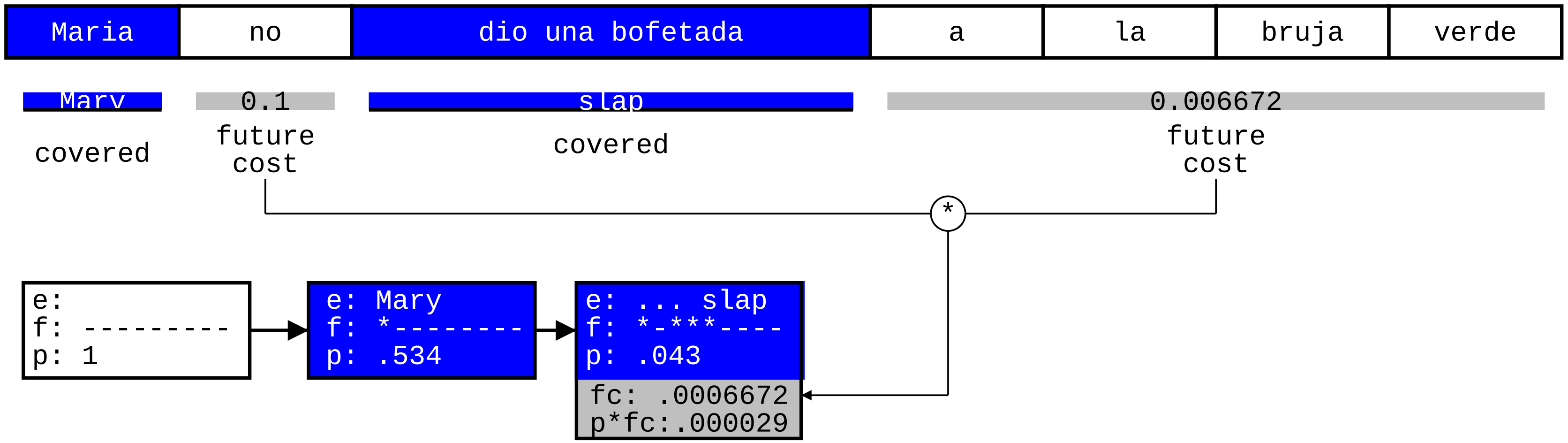
- Step 2: find *cheapest cost* among translation options

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*

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

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

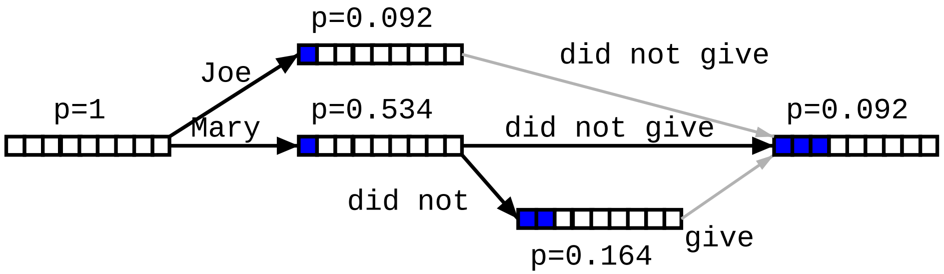
Limits on Reordering

- 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

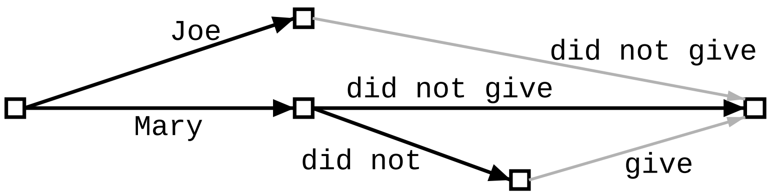
Word Lattice Generation



- **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**:

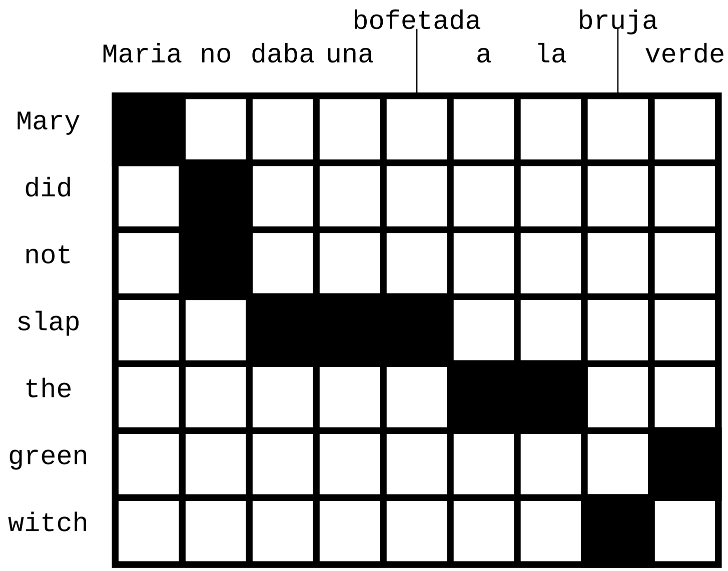
```
Translation ||| Reordering LM TM WordPenalty ||| Score
this is a small house ||| 0 -27.0908 -1.83258 -5 ||| -28.9234
this is a little house ||| 0 -28.1791 -1.83258 -5 ||| -30.0117
it is a small house ||| 0 -27.108 -3.21888 -5 ||| -30.3268
it is a little house ||| 0 -28.1963 -3.21888 -5 ||| -31.4152
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
```

Phrase-based models

Word alignment

- Notion of **word alignment** valuable

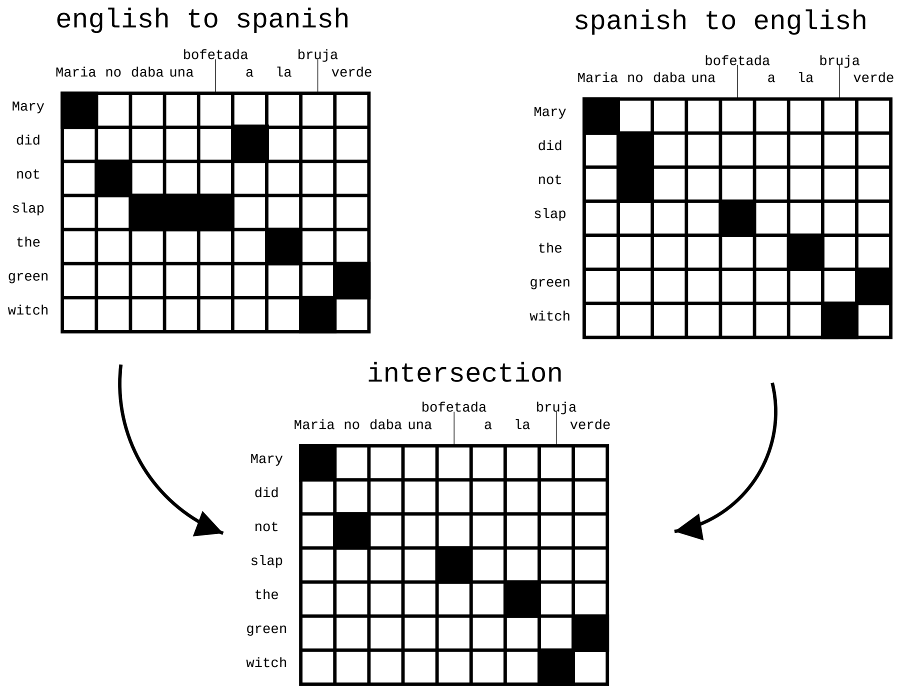
- Shared task at NAACL 2003 and ACL 2005 workshops



Word alignment with IBM models

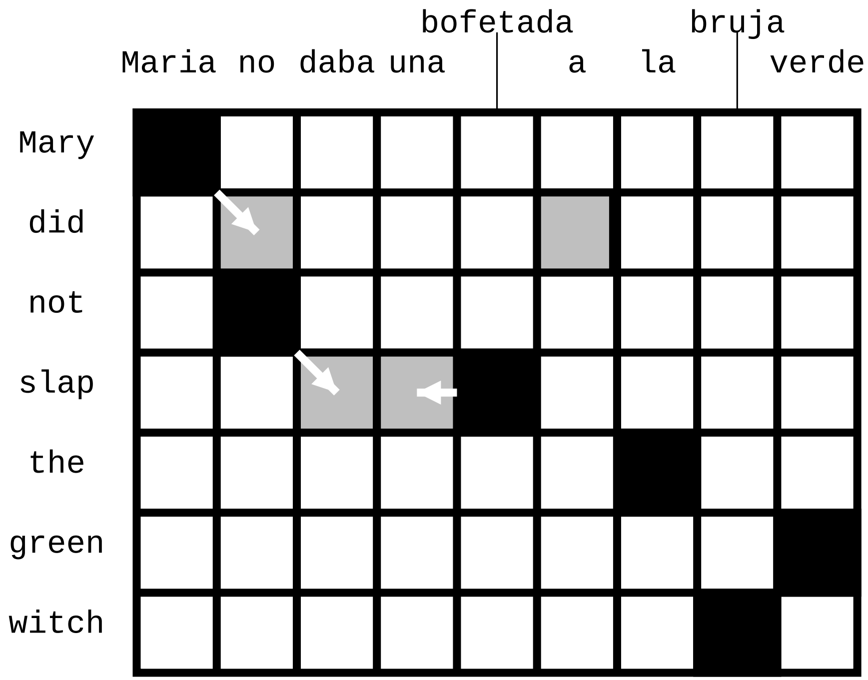
- IBM Models create a *many-to-one* mapping
  - words are aligned using an **alignment function**
  - a function may return the same value for different input (one-to-many mapping)
  - a function can not return multiple values for one input (*no many-to-one* mapping)
- But we need *many-to-many* mappings

Symmetrizing word alignments



- *Intersection* of GIZA++ bidirectional alignments

Symmetrizing word alignments



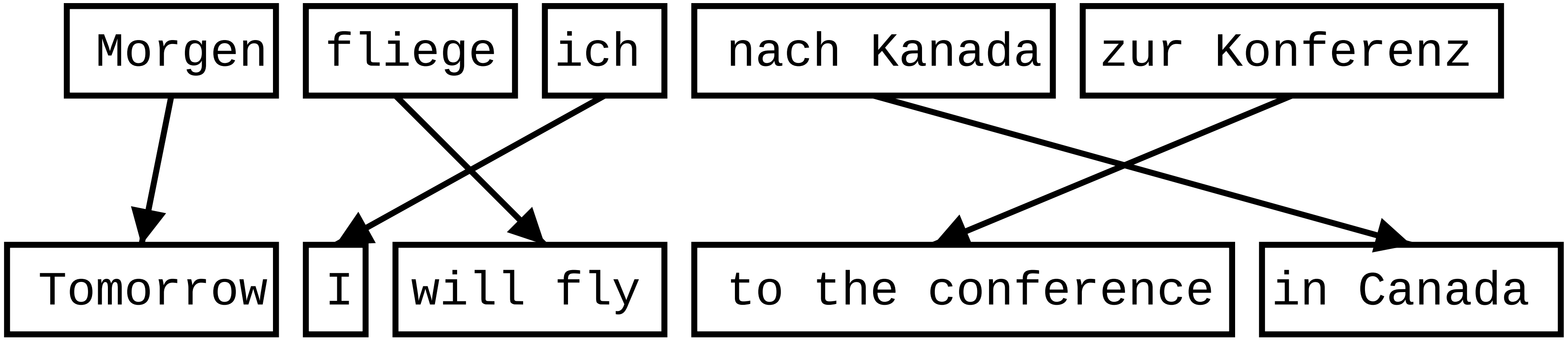
- *Grow* additional alignment points [Och and Ney, CompLing2003]

Growing heuristic

```
GROW-DIAG-FINAL(e2f,f2e):
  neighboring = ((-1,0),(0,-1),(1,0),(0,1),(-1,-1),(-1,1),(1,-1),(1,1))
  alignment = intersect(e2f,f2e);
  GROW-DIAG(); FINAL(e2f); FINAL(f2e);

GROW-DIAG():
  iterate until no new points added
    for english word e = 0 ... en
      for foreign word f = 0 ... fn
        if ( e aligned with f )
          for each neighboring point ( e-new, f-new ):
            if ( ( e-new not aligned and f-new not aligned ) and
                 ( e-new, f-new ) in union( e2f, f2e ) )
              add alignment point ( e-new, f-new )
FINAL(a):
  for english word e-new = 0 ... en
    for foreign word f-new = 0 ... fn
      if ( ( e-new not aligned or f-new not aligned ) and
           ( e-new, f-new ) in alignment a )
        add alignment point ( e-new, f-new )
```

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

Phrase-based translation model

- Major components of phrase-based model
  - **phrase translation model** $\phi(\mathbf{f}|\mathbf{e})$
  - **reordering model** $\omega^{\text{length}(\mathbf{e})}$
  - **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 $\phi(\mathbf{f}|\mathbf{e})$

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

Advantages of phrase-based translation

- *Many-to-many* translation can handle non-compositional phrases
- Use of *local context* in translation
- The more data, the *longer phrases* can be learned

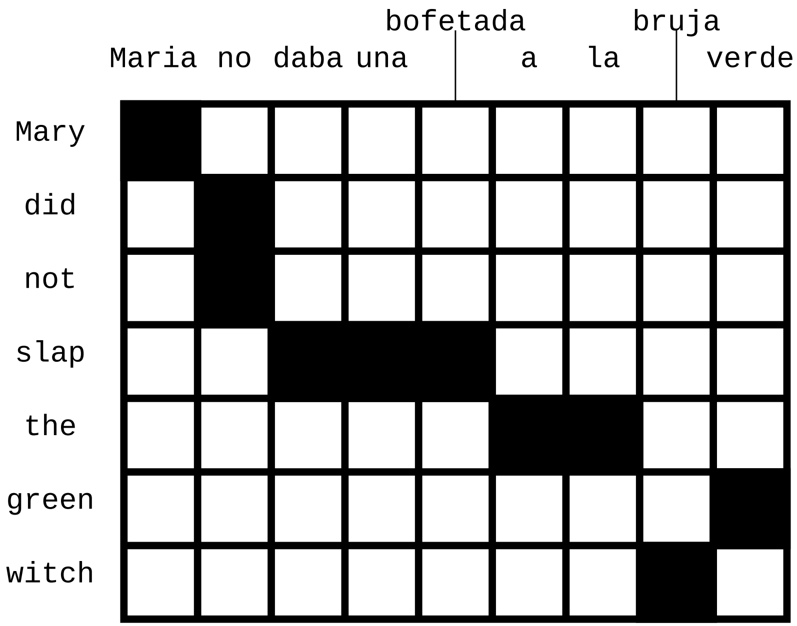
Phrase translation table

- Phrase translations for *den Vorschlag*

English	$\phi(\mathbf{e}|\mathbf{f})$	**English**	$\phi(\mathbf{e}|\mathbf{f})$
the proposal	0.6227	the suggestions	0.0114
's proposal	0.1068	the proposed	0.0114
a proposal	0.0341	the motion	0.0091
the idea	0.0250	the idea of	0.0091
this proposal	0.0227	the proposal ,	0.0068
proposal	0.0205	its proposal	0.0068
of the proposal	0.0159	it	0.0068
the proposals	0.0159	...	...

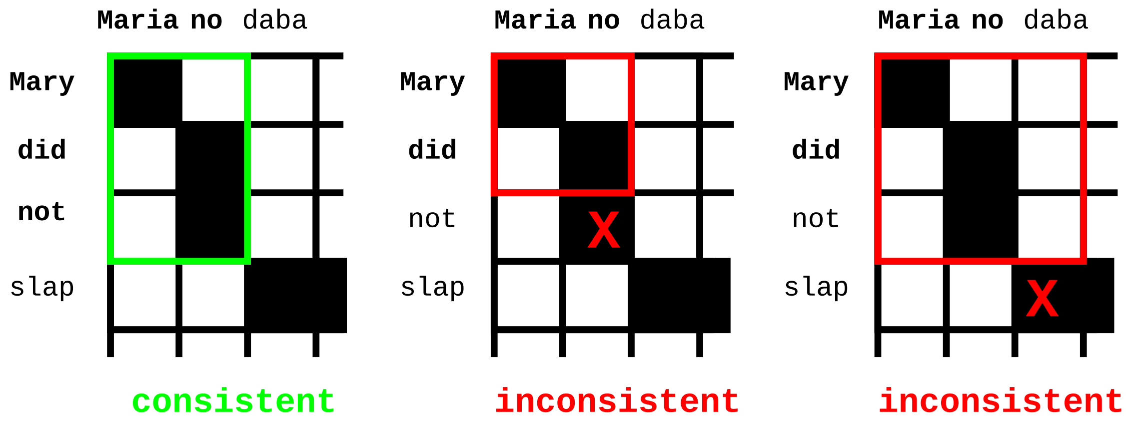
How to learn the phrase translation table?

- Start with the *word alignment*:



- Collect all phrase pairs that are **consistent** with the word alignment

Consistent with word alignment



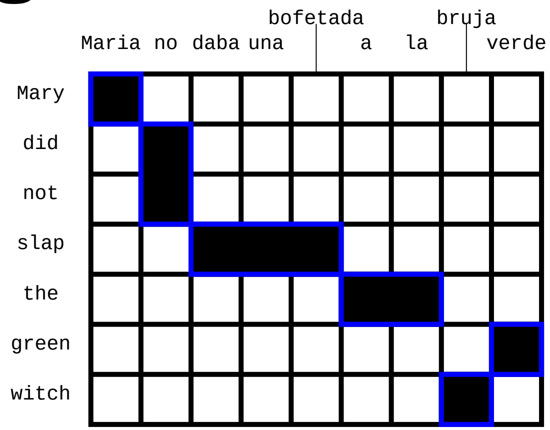
- **Consistent with the word alignment** :=

phrase alignment has to *contain all alignment points* for all covered words

$$(\bar{e}, \bar{f}) \in BP \Leftrightarrow \quad \forall e_i \in \bar{e} : (e_i, f_j) \in A \rightarrow f_j \in \bar{f}$$

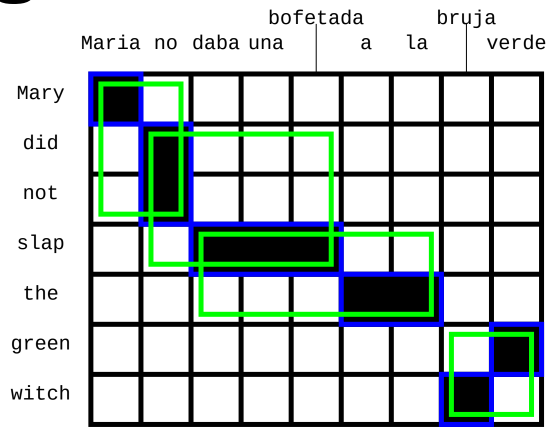
$$\text{AND} \quad \forall f_j \in \bar{f} : (e_i, f_j) \in A \rightarrow e_i \in \bar{e}$$

Word alignment induced phrases



(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green)

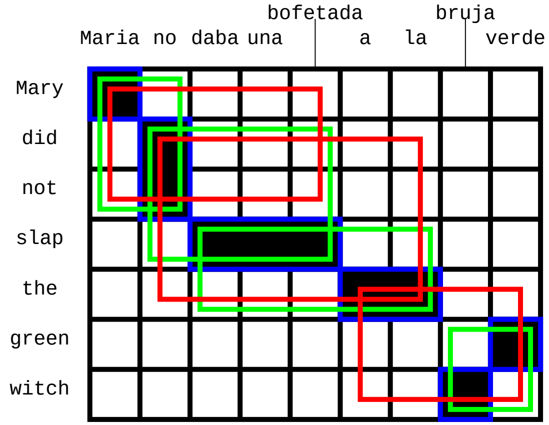
Word alignment induced phrases



(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green),

(Maria no, Mary did not), (no daba una bofetada, did not slap), (daba una bofetada a la, slap the),
(bruja verde, green witch)

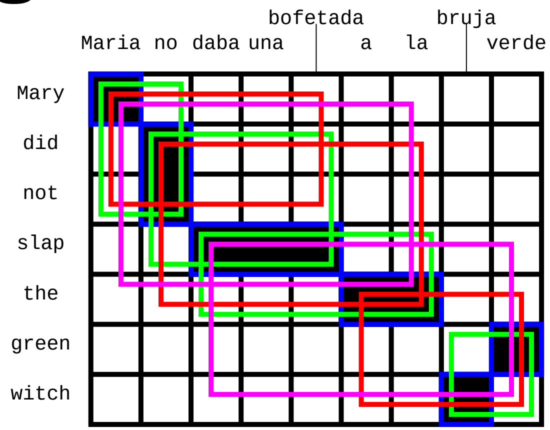
Word alignment induced phrases



(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green),

(Maria no, Mary did not), (no daba una bofetada, did not slap), (daba una bofetada a la, slap the),
(bruja verde, green witch), (Maria no daba una bofetada, Mary did not slap),
(no daba una bofetada a la, did not slap the), (a la bruja verde, the green witch)

Word alignment induced phrases



(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green),

(Maria no, Mary did not), (no daba una bofetada, did not slap), (daba una bofetada a la, slap the),

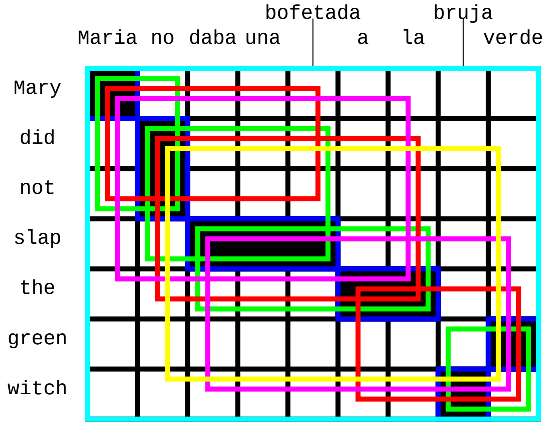
(bruja verde, green witch), (Maria no daba una bofetada, Mary did not slap),

(no daba una bofetada a la, did not slap the), (a la bruja verde, the green witch),

(Maria no daba una bofetada a la, Mary did not slap the),

(daba una bofetada a la bruja verde, slap the green witch)

Word alignment induced phrases (5)



(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green),

(Maria no, Mary did not), (no daba una bofetada, did not slap), (daba una bofetada a la, slap the), (bruja verde, green witch), (Maria no daba una bofetada, Mary did not slap), (no daba una bofetada a la, did not slap the), (a la bruja verde, the green witch), (Maria no daba una bofetada a la, Mary did not slap the), (daba una bofetada a la bruja verde, slap the green witch), (no daba una bofetada a la bruja verde, did not slap the green witch), (Maria no daba una bofetada a la bruja verde, Mary did not slap the green witch)

Probability distribution of phrase pairs

- We need a **probability distribution** $\phi(\overline{f}|\overline{e})$ over the collected phrase pairs

$\Rightarrow$ Possible *choices*

- *relative frequency* of collected phrases: $\phi(\overline{f}|\overline{e}) = \frac{\text{count}(\overline{f},\overline{e})}{\sum_{\overline{f}} \text{count}(\overline{f},\overline{e})}$
- or, conversely $\phi(\overline{e}|\overline{f})$
- use *lexical translation probabilities*

Reordering

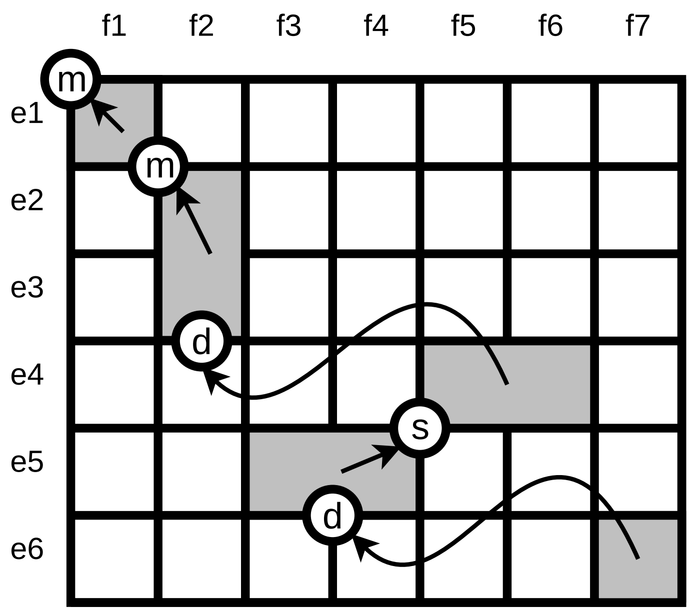
- *Monotone* translation
  - do not allow any reordering
  - $\rightarrow$ worse translations

- *Limiting* reordering (to movement over max. number of words) helps

- *Distance-based* reordering cost
  - moving a foreign phrase over n words: cost ω^n

- *Lexicalized* reordering model

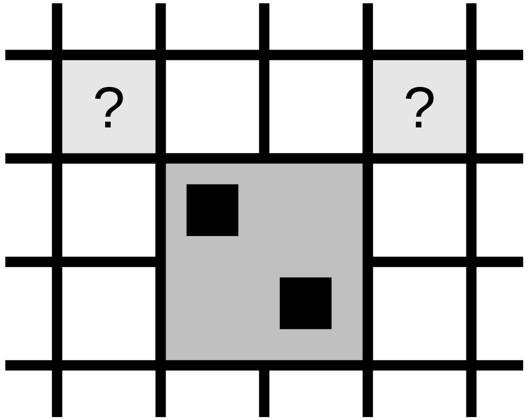
Lexicalized reordering models



[from Koehn et al., 2005, IWSLT]

- Three **orientation** types: **monotone**, **swap**, **discontinuous**
- Probability $p(swap|e,f)$ depends on foreign (and English) *phrase* involved

Learning lexicalized reordering models

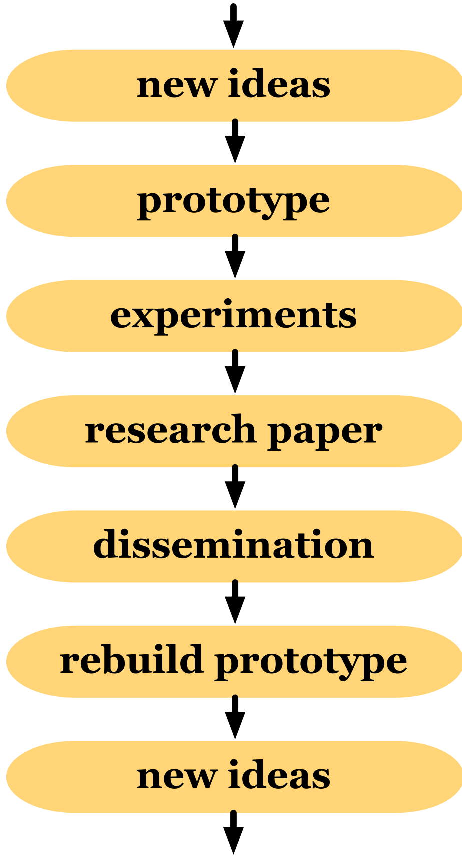


[from Koehn et al., 2005, IWSLT]

- Orientation type is *learned during phrase extractions*
- *Alignment point* to the *top left* (monotone) or *top right* (swap)?
- For more, see [Tillmann, 2003] or [Koehn et al., 2005]

Open Source Machine Translation

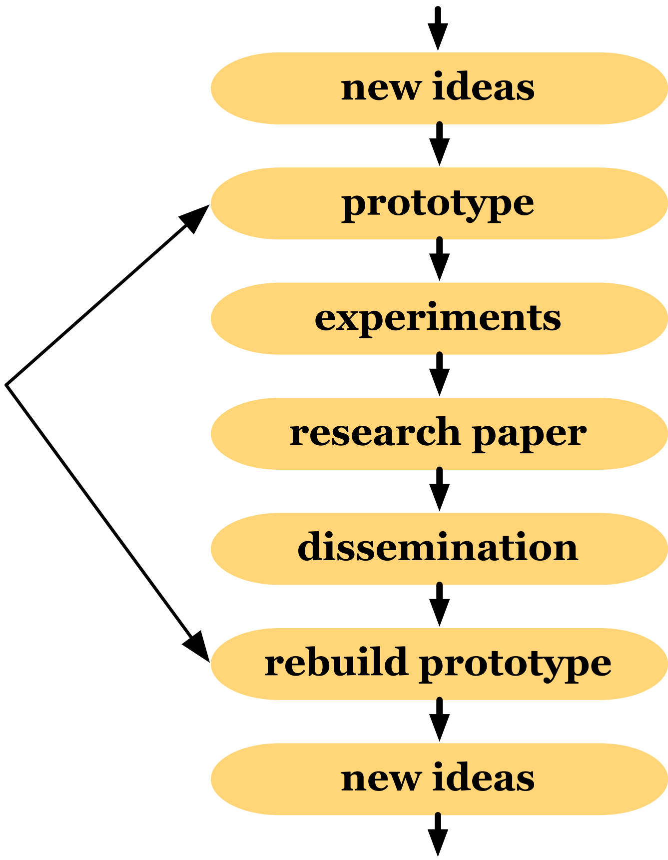
Research Process



Research Process

SMT is increasingly
a big systems field

building prototypes
requires huge efforts



Requirements for Building MT Systems

- **Data resources**
  - *parallel* corpora (translated texts)
  - *monolingual* corpora, especially for output language
- **Support tools**
  - basic *corpus preparation*: tokenization, sentence alignment
  - *linguistic* tools: tagger, parsers, morphology, semantic processing
- **MT tools**
  - word alignment, *training*
  - *decoding* (translation engine)
  - tuning (optimization)
  - re-ranking, incl. posterior methods

Who will do MT Research?

- If MT research requires the development of *many resources*
  - who will be able to do relevant research?
  - who will be able to deploy the technology?

- A *few* big labs?

- ... or a *broad network* of academic and commercial institutions?

MT is diverse

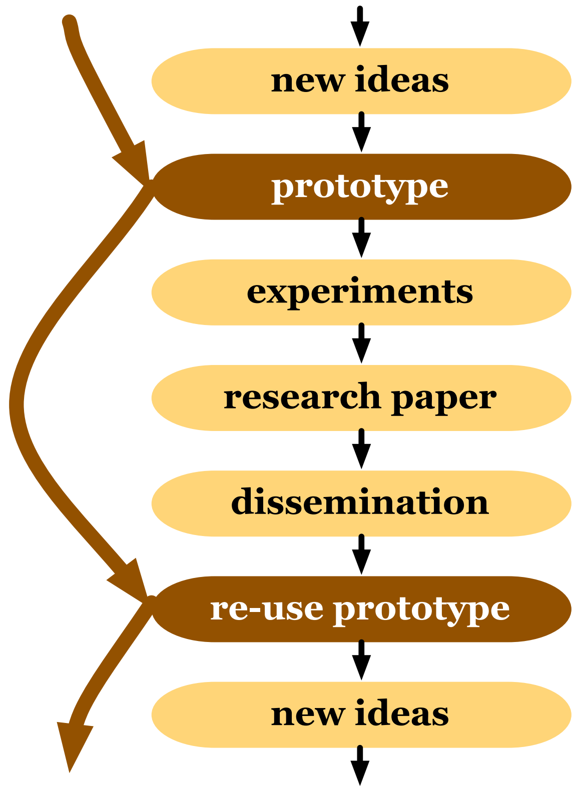
- Many different **stakeholders**
  - academic researchers
  - commercial developers
  - multi-lingual or trans-lingual content providers
  - end users of online translation services
  - human translation service providers
- Many different **language pairs**
  - few languages with rich resources: *English, Spanish, German, Chinese, ...*
  - many second tier languages: *Czech, Danish, Greek, ...*
  - many under-resourced languages: *Gaelic, Basque, ...*

Open Research

SMT is increasingly
a big systems field

building prototypes
requires huge efforts

sharing of resources reduces duplication of efforts



Making Open Research Work

- Non-restrictive **licensing**

- Active **development**
  - working high-quality prototype
  - ongoing development
  - open to contributions

- **Support** and dissemination
  - support by email, web sites, documentation
  - offering tutorials and courses

Moses: Open Source Toolkit

- **Open source** statistical machine translation system (developed from scratch 2006)
  - state-of-the-art *phrase-based* approach
  - novel methods: *factored translation models*, *confusion network decoding*
  - support for *very large models* through *memory-efficient* data structures

- Documentation, source code, binaries **available** at `http://www.statmt.org/moses/`

- Development also **supported by**
  - EC-funded *TC-STAR* project
  - *US* funding agencies DARPA, NSF
  - universities (Edinburgh, Maryland, MIT, ITC-irst, RWTH Aachen, ...)

Call for Participation: 3rd MT Marathon

- Prague, Czech Republic, January 26-30
- Events
  - winter school (5-day course on MT)
  - research showcase
  - open source showcase: call for papers, due December 2nd
  - open source hands-on projects
- Sponsored by EuroMatrix project — free of charge

Syntax-based models

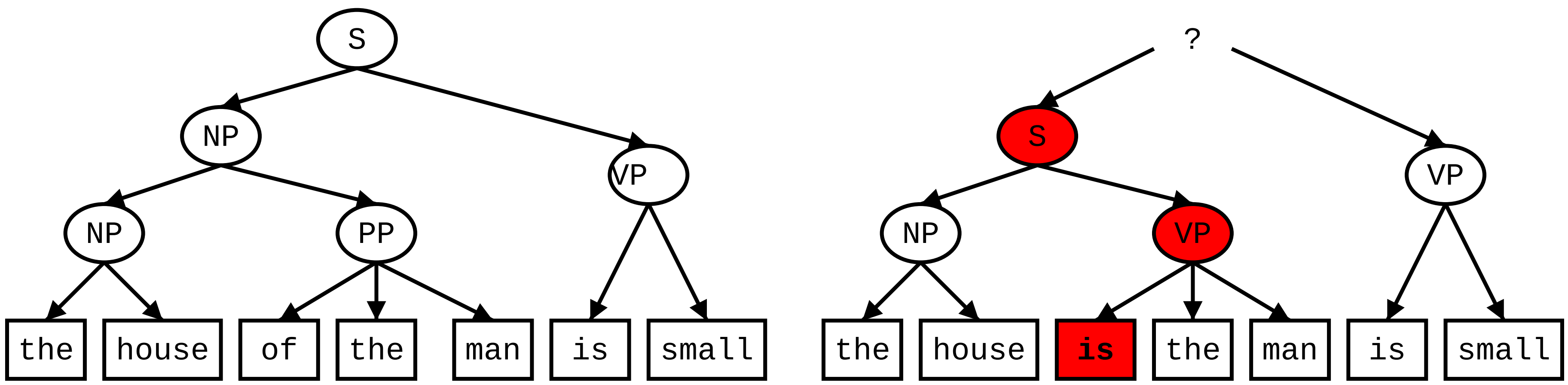
Advantages of Syntax-Based Translation

- *Reordering* for syntactic reasons
  - e.g., move German object to end of sentence
- Better explanation for *function words*
  - e.g., prepositions, determiners
- Conditioning to *syntactically related words*
  - translation of verb may depend on subject or object
- Use of *syntactic language models*
  - ensuring grammatical output

Syntactic Language Model

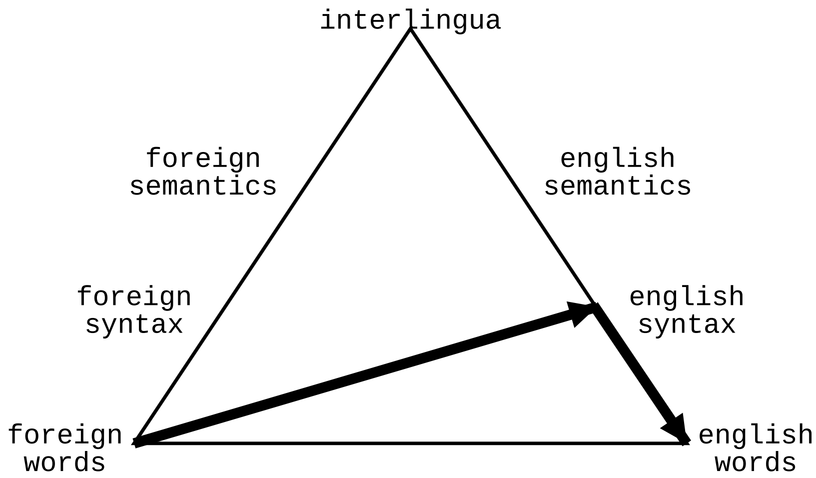
- *Good syntax tree* → good English

- Allows for *long distance constraints*



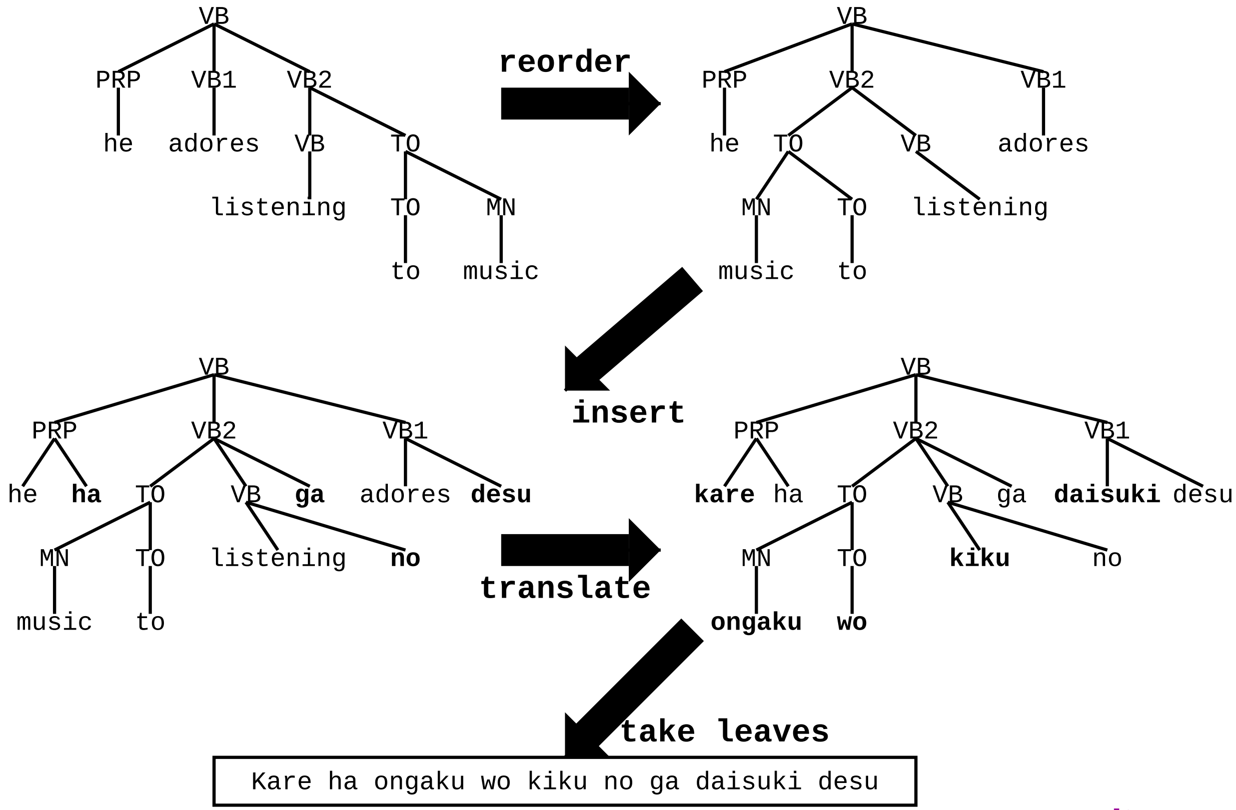
- Left translation preferred by syntactic LM

String to Tree Translation



- Use of English *syntax trees* [Yamada and Knight, 2001]
  - exploit *rich resources* on the English side
  - obtained with statistical parser [Collins, 1997]
  - *flattened tree* to allow more reorderings
  - works well with syntactic language model

Yamada and Knight [2001]



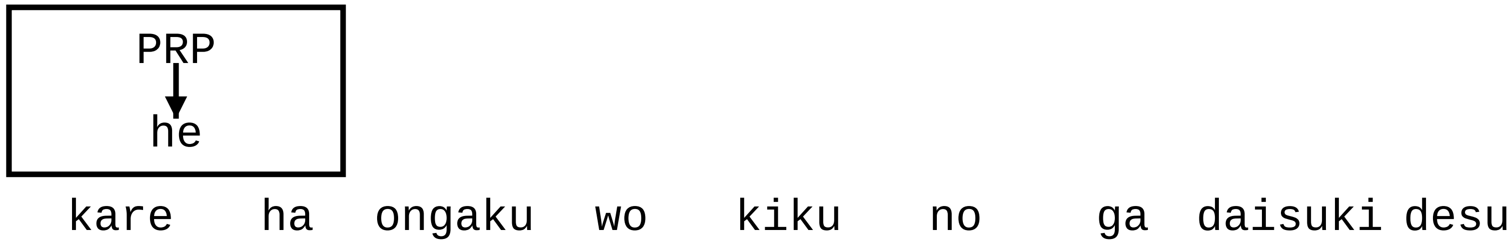
[from Yamada and Knight, 2001]

Reordering Table

Original Order	Reordering	p(reorder|original)
PRP VB1 VB2	PRP VB1 VB2	0.074
PRP VB1 VB2	**PRP VB2 VB1**	**0.723**
PRP VB1 VB2	VB1 PRP VB2	0.061
PRP VB1 VB2	VB1 VB2 PRP	0.037
PRP VB1 VB2	VB2 PRP VB1	0.083
PRP VB1 VB2	VB2 VB1 PRP	0.021
VB TO	VB TO	0.107
VB TO	**TO VB**	**0.893**
TO NN	TO NN	0.251
TO NN	**NN TO**	**0.749**

Decoding as Parsing

- **Chart Parsing**

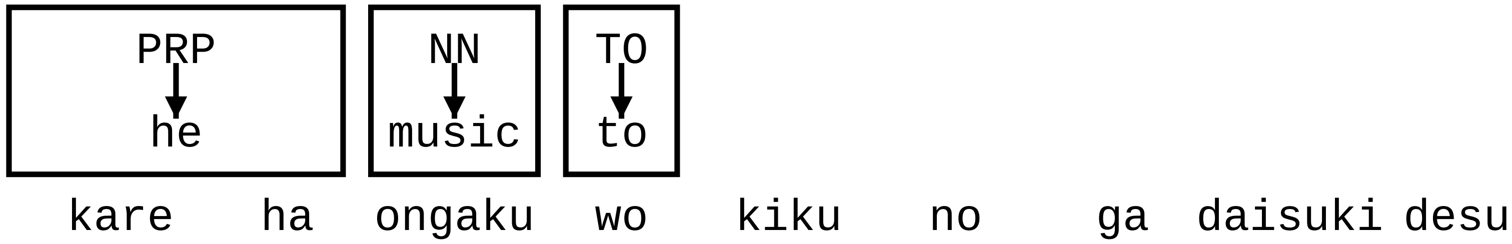


- Pick Japanese *words*

- Translate into *tree stumps*

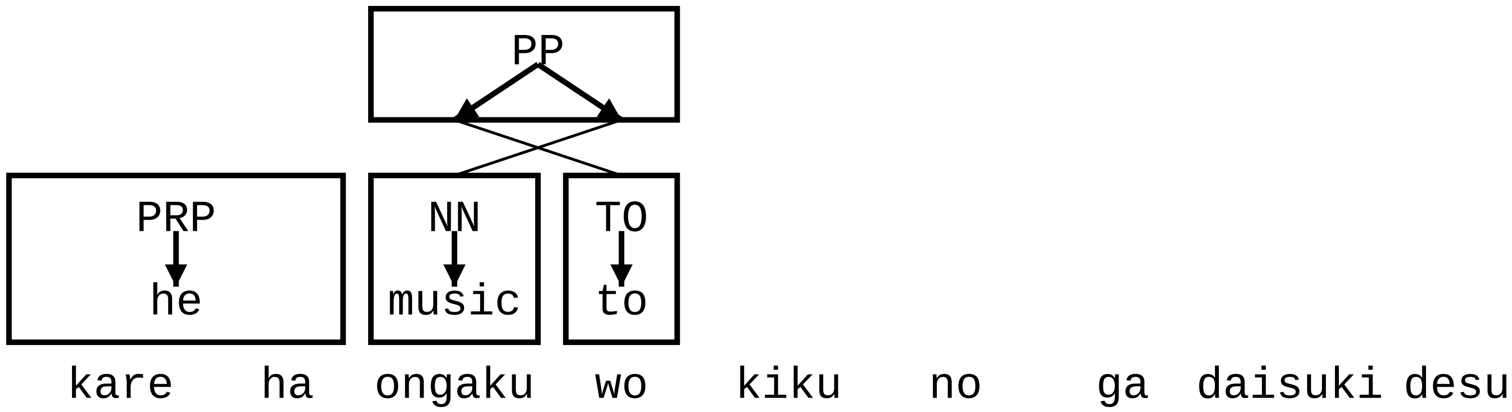
Decoding as Parsing

- Chart Parsing



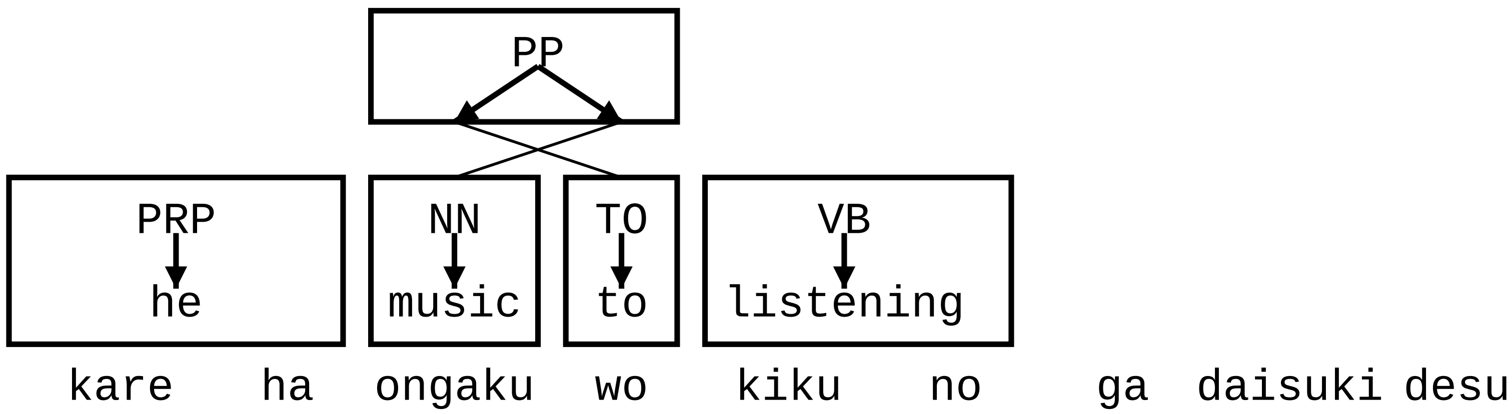
- Pick Japanese words
- Translate into tree stumps

Decoding as Parsing



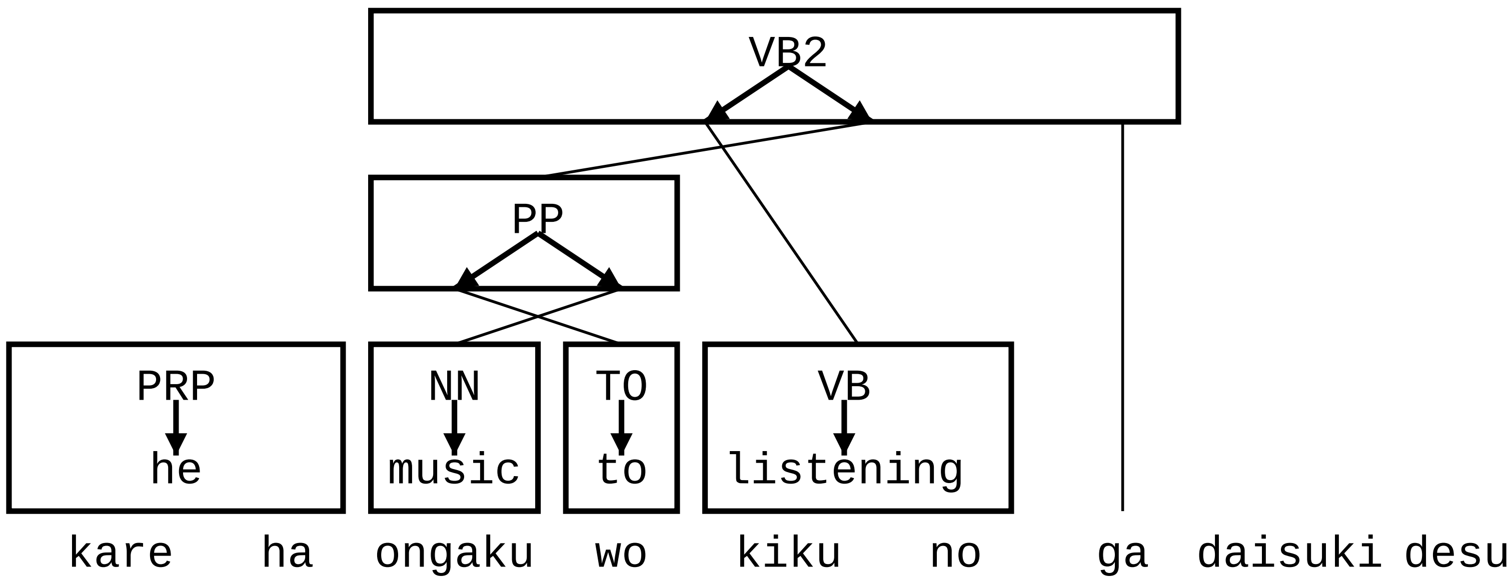
- Adding some *more entries*...

Decoding as Parsing

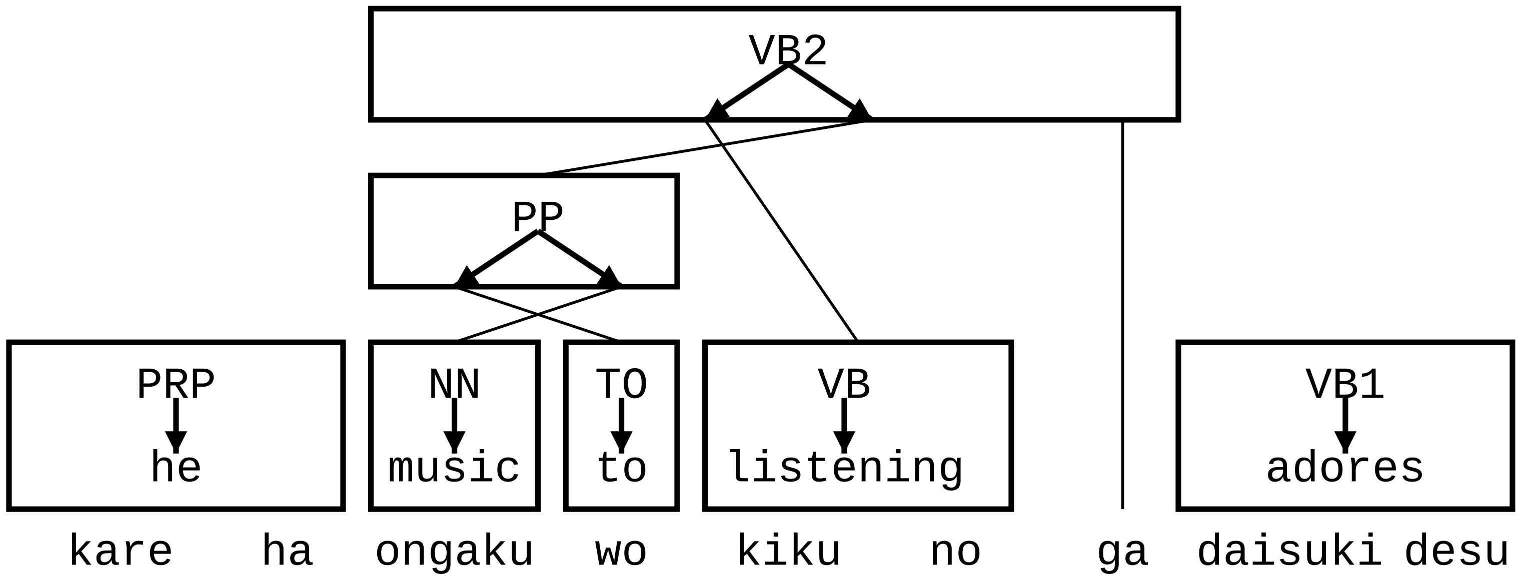


- *Combine entries*

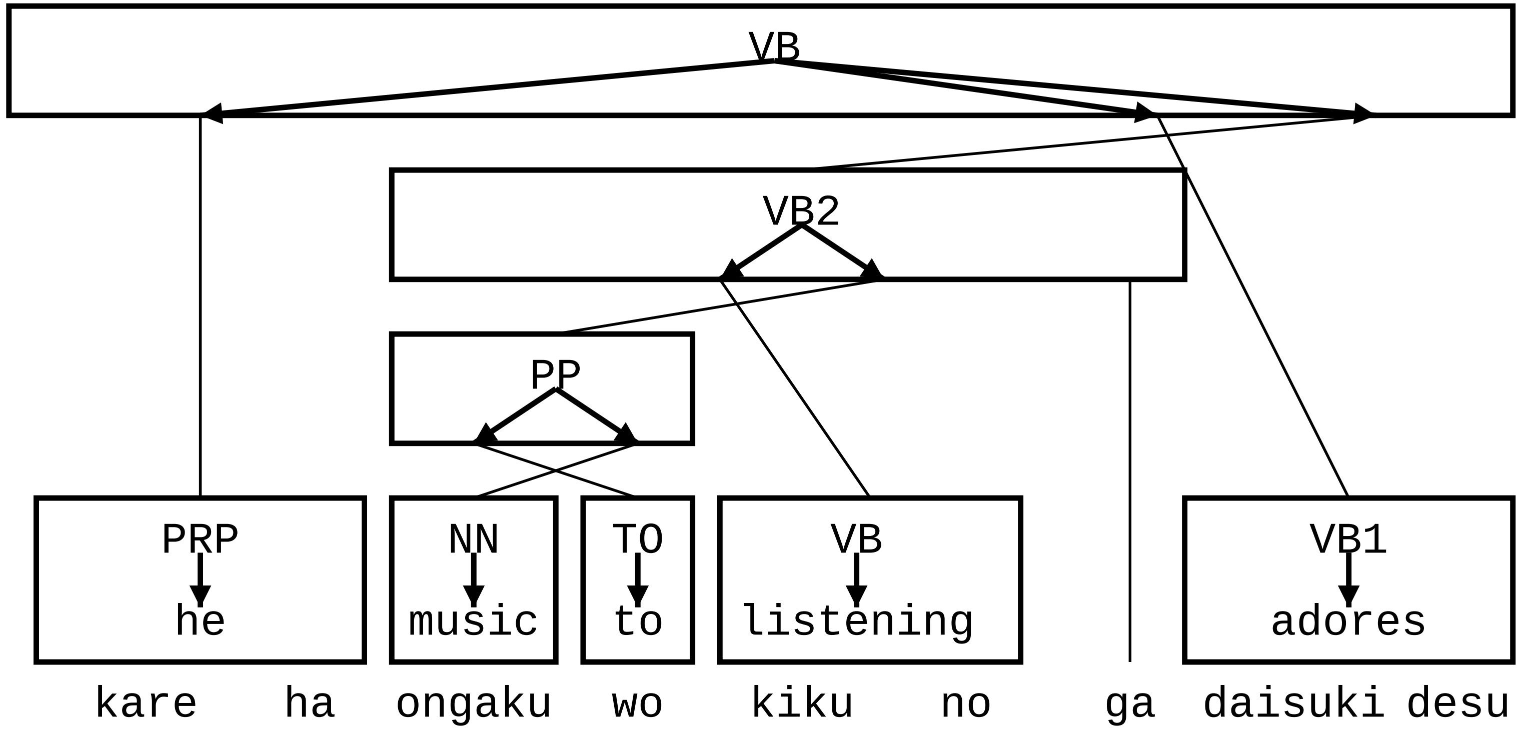
Decoding as Parsing



Decoding as Parsing



Decoding as Parsing



- *Finished* when all foreign words covered

Yamada and Knight: Training

- *Parsing* of the English side
  - using Collins statistical parser

- *EM training*
  - translation model is used to map training sentence pairs
  - EM training finds low-perplexity model

→ *unity of training and decoding* as in IBM models

Is the Model Realistic?

- Do English trees *match* foreign strings?
- Crossings between French-English [Fox, 2002]
  - 0.29-6.27 per sentence, depending on how it is measured
- Can be reduced by
  - *flattening tree*, as done by [Yamada and Knight, 2001]
  - detecting *phrasal* translation
  - *special treatment* for small number of constructions
- Most coherence between **dependency structures**

Chiang: Hierarchical Phrase Model

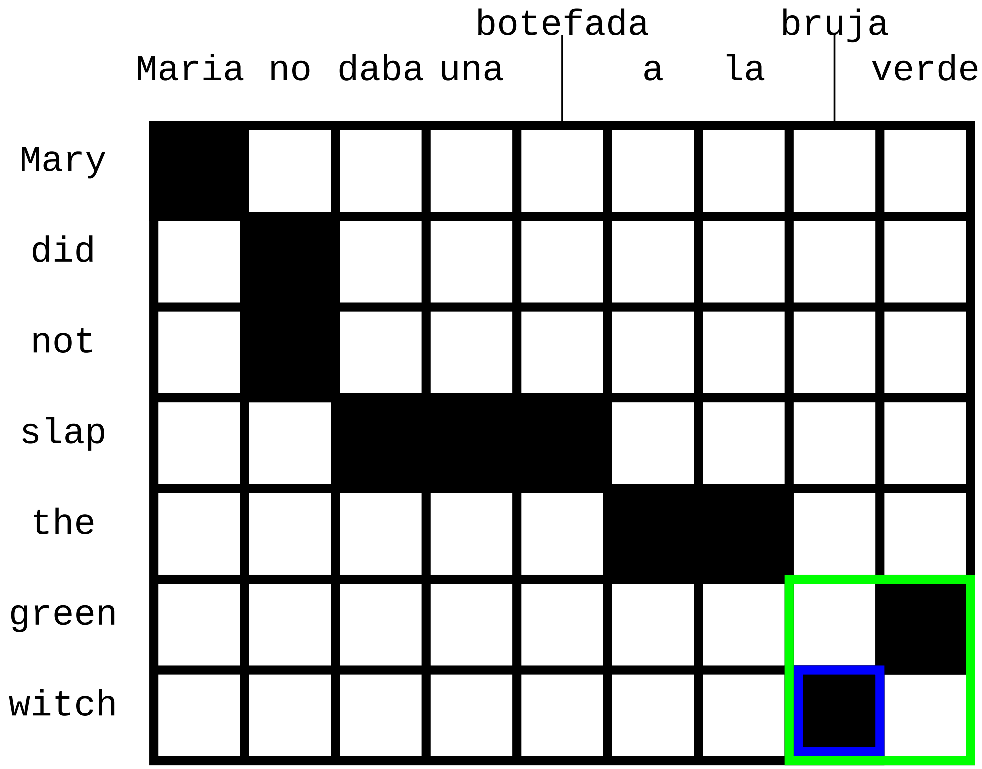
- **Chiang** [ACL, 2005] (best paper award!)
  - context free bi-grammar
  - *one non-terminal* symbol
  - right hand side of rule may include non-terminals and terminals

- *Competitive* with phrase-based models in 2005 DARPA/NIST evaluation

Types of Rules

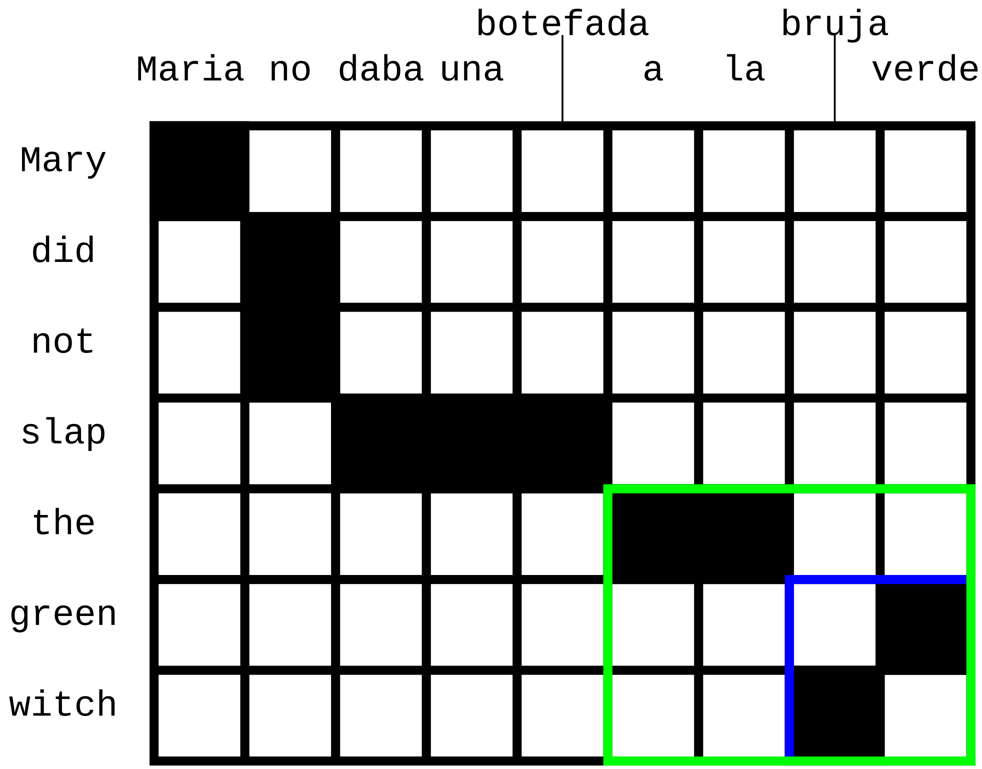
- *Word* translation
  - $X \rightarrow$ *maison* $\|$ *house*
- *Phrasal* translation
  - $X \rightarrow$ *daba una bofetada* $|$ *slap*
- *Mixed* non-terminal / terminal
  - $X \rightarrow X$ *bleue* $\|$ *blue* X
  - $X \rightarrow$ *ne* X *pas* $\|$ *not* X
  - $X \rightarrow X1\ X2 \,\|\, X2$ *of* $X1$
- Technical rules
  - $S \rightarrow S\ X \,\|\, S\ X$
  - $S \rightarrow X \,\|\, X$

Learning Hierarchical Rules



$X \rightarrow X$ verde $\|$ green X

Learning Hierarchical Rules



$X \rightarrow$ a la $X \parallel$ the X

Details of Chiang's Model

- Too many rules

$\rightarrow$ *filtering* of rules necessary

- *Efficient* parse decoding possible
  - hypothesis stack for each span of foreign words
  - only *one non-terminal* $\rightarrow$ hypotheses comparable
  - *length limit* for spans that do not start at beginning

Clause Level Restructuring [Collins et al.]

- Why **clause structure**?
  - languages *differ vastly* in their clause structure (English: SVO, Arabic: VSO, German: fairly *free order*; a lot details differ: position of adverbs, sub clauses, etc.)
  - large-scale restructuring is a *problem* for phrase models

- **Restructuring**
  - *reordering* of constituents (main focus)
  - add/drop/change of *function words*

- Details see [Collins, Kucerova and Koehn, ACL 2005]

Clause Structure

```
S   PPER-SB   Ich    I
    VAFIN-HD  werde   will
    VP-OC     PPER-DA Ihnen   you
              NP-OA   ART-OA  die   the
                      ADJ-NK  entsprechenden   corresponding
                      NN-NK   Anmerkungen   comments
              VVFIN   aushaendigen     pass on
              $,      ,    ,
              S-MO    KOUS-CP damit   so that
                      PPER-SB Sie    you
                      VP-OC   PDS-OA  das   that
                              ADJD-MO eventuell    perhaps
                              PP-MO   APRD-MO  bei   in
                                      ART-DA   der   the
                                      NN-NK    Abstimmung  vote
                              VVINF   uebernehmen    include
                      VMFIN   koennen   can
$. .             .
```

MAIN CLAUSE

SUB-ORDINATE CLAUSE

- *Syntax tree* from German parser
  - statistical parser by Amit Dubay, trained on TIGER treebank

Reordering When Translating

```
S     PPER-SB   Ich                                  I
      VAFIN-HD  werde                                will
      PPER-DA   Ihnen                                you
      NP-OA     ART-OA  die                          the
                ADJ-NK  entsprechenden                corresponding
                NN-NK   Anmerkungen                   comments
      VVFIN     aushaendigen                         pass on
$,    ,                                              ,
S-MO  KOUS-CP   damit                                so that
      PPER-SB   Sie                                  you
      PDS-OA    das                                  that
      ADJD-MO   eventuell                            perhaps
      PP-MO     APRD-MO  bei                         in
                ART-DA   der                          the
                NN-NK    Abstimmung                   vote
      VVINF     uebernehmen                          include
      VMFIN     koennen                              can
$.  .                                                .
```

- *Reordering* when translating into English
  - tree is *flattened*
  - clause level constituents line up

Clause Level Reordering

S	PPER-SB	Ich		1	I
	VAFIN-HD	werde		2	will
	PPER-DA	Ihnen		4	you
	NP-OA	ART-OA	die	5	the
		ADJ-NK	entsprechenden		corresponding
		NN-NK	Anmerkungen		comments
	VVFIN	aushaendigen		3	pass on
$,	,				,
S-MO	KOUS-CP	damit		1	so that
	PPER-SB	Sie		2	you
	PDS-OA	das		6	that
	ADJD-MO	eventuell		4	perhaps
	PP-MO	APRD-MO	bei	7	in
		ART-DA	der		the
		NN-NK	Abstimmung		vote
	VVINF	uebernehmen		5	include
	VMFIN	koennen		3	can
$.	.				.

- Clause level reordering is a *well defined task*
  - label German constituents with their *English order*
  - done this for 300 sentences, two annotators, high agreement

Systematic Reordering German → English

- Many types of reorderings are **systematic**
  - *move verb group together*
  - *subject - verb - object*
  - *move negation in front of verb*

⇒ *Write rules by hand*
  - apply rules to test and training data
  - train standard *phrase-based* SMT system

System	BLEU
baseline system	25.2%
with manual rules	26.8%

Other Syntax-Based Approaches

- ISI: extending work of Yamada/Knight
  - more *complex rules*
  - performance approaching phrase-based
- Prague: Translation via *dependency structures*
  - parallel Czech–English dependency treebank
  - tecto-grammatical translation model [EACL 2003]
- U.Alberta/Microsoft: *treelet translation*
  - translating from English into foreign languages
  - using dependency parser in English
  - project *dependency tree* into foreign language for training
  - map parts of the dependency tree ("treelets") into foreign languages

Other Syntax-Based Approaches

- Context feature model for rule selection and reordering
  - SVM for rule selection in hierarchical model [Chan et al., 2007]
  - maximum entropy model for reordering [Xiong et al., 2008; He et al., 2008]

- *Reranking* phrase-based SMT output with syntactic features
  - create n-best list with phrase-based system
  - POS tag and parse candidate translations
  - rerank with syntactic features
  - see [Koehn, 2003] and JHU Workshop [Och et al., 2003]

- JHU Summer workshop 2005
  - **Genpar**: tool for syntax-based SMT

Syntax: Does it help?

- *Getting there*
  - for some languages competitive with best phrase-based systems
- *Some evidence*
  - work on reordering German
  - ISI: better for Chinese–English
  - automatically trained tree transfer systems promising
- Challenges
  - if real syntax, we need *good parsers* — are they good enough?
  - syntactic annotations add a level of *complexity*

  $\rightarrow$ difficult to handle, slow to train and decode
  - few researchers good at statistical modeling and syntactic theories

Factored Translation Models

Factored Translation Models

- **Motivation**
- Example
- Model and Training
- Decoding
- Experiments

Statistical machine translation today

- Best performing methods based on **phrases**
  - short sequences of words
  - no use of explicit syntactic information
  - no use of morphological information
  - currently best performing method

- Progress in **syntax-based** translation
  - tree transfer models using syntactic annotation
  - still shallow representation of words and non-terminals
  - active research, improving performance

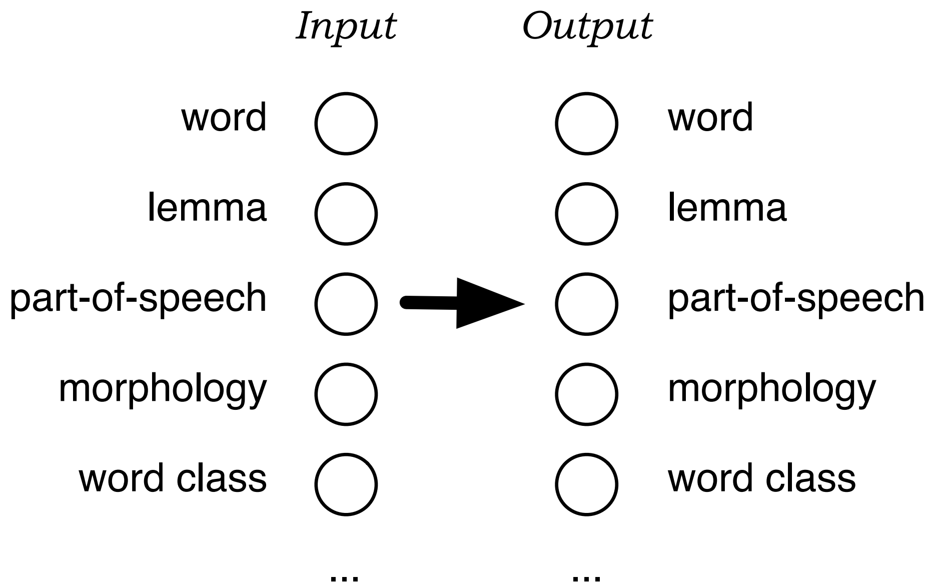
One motivation: morphology

- Models treat *car* and *cars* as completely different words
  - training occurrences of *car* have no effect on learning translation of *cars*
  - if we only see *car*, we do not know how to translate *cars*
  - rich morphology (German, Arabic, Finnish, Czech, ...) $\rightarrow$ many word forms

- Better approach
  - analyze surface word forms into **lemma** and **morphology**, e.g.: *car +plural*
  - translate lemma and morphology separately
  - generate target surface form

Factored translation models

- **Factored represention** of words



- Goals
  - **Generalization**, e.g. by translating lemmas, not surface forms
  - **Richer model**, e.g. using syntax for reordering, language modeling)

Related work

- **Back off** to representations with richer statistics (lemma, etc.)
  [Nießen and Ney, 2001, Yang and Kirchhoff 2006, Talbot and Osborne 2006]
- Use of additional annotation in **pre-processing** (POS, syntax trees, etc.)
  [Collins et al., 2005, Crego et al, 2006]
- Use of additional annotation in **re-ranking** (morphological features, POS, syntax trees, etc.)
  [Och et al. 2004, Koehn and Knight, 2005]

→ we pursue an *integrated approach*

- Use of syntactic **tree structure**
  [Wu 1997, Alshawi et al. 1998, Yamada and Knight 2001, Melamed 2004, Menezes and Quirk 2005, Chiang 2005, Galley et al. 2006]

→ may be *combined* with our approach

Factored Translation Models

- Motivation
- **Example**
- Model and Training
- Decoding
- Experiments

Decomposing translation: example

- **Translate** lemma and syntactic information **separately**

$$\begin{bmatrix} \text{lemma} \end{bmatrix} \Rightarrow \begin{bmatrix} \text{lemma} \end{bmatrix}$$

$$\begin{bmatrix} \text{part-of-speech} \\ \text{morphology} \end{bmatrix} \Rightarrow \begin{bmatrix} \text{part-of-speech} \\ \text{morphology} \end{bmatrix}$$

Decomposing translation: example

- **Generate surface** form on target side

$$\begin{array}{c} \begin{bmatrix} \text{surface} \end{bmatrix} \\ \Uparrow \\ \begin{bmatrix} \text{lemma} \\ \text{part-of-speech} \\ \text{morphology} \end{bmatrix} \end{array}$$

Translation process: example

Input: *(Autos, Auto, NNS)*

1. Translation step: lemma ⇒ lemma
   (?, car, ?), (?, auto, ?)

2. Generation step: lemma ⇒ part-of-speech
   (?, car, NN), (?, car, NNS), (?, auto, NN), (?, auto, NNS)

3. Translation step: part-of-speech ⇒ part-of-speech
   (?, car, NN), (?, car, NNS), (?, auto, NNP), (?, auto, NNS)

4. Generation step: lemma,part-of-speech ⇒ surface
   (car, car, NN), (cars, car, NNS), (auto, auto, NN), (autos, auto, NNS)

Factored Translation Models

- Motivation
- Example
- **Model and Training**
- Decoding
- Experiments

Model

- Extension of *phrase model*

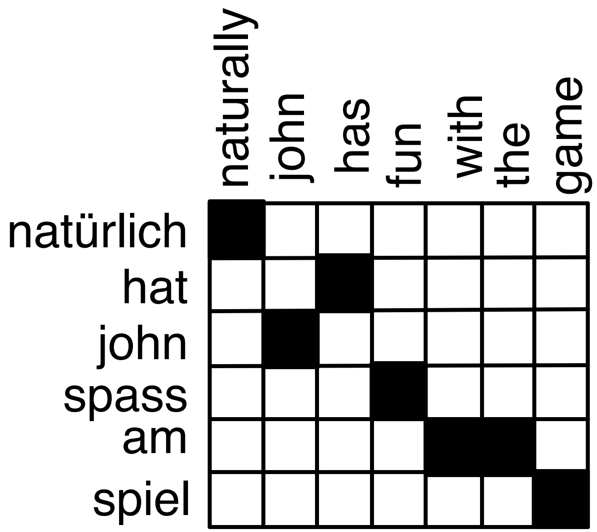
- Mapping of foreign words into English words broken up into steps
  - **translation step**: maps foreign factors into English factors (on the phrasal level)
  - **generation step**: maps English factors into English factors (for each word)

- Each step is modeled by one or more *feature functions*
  - fits nicely into log-linear model
  - weight set by discriminative training method

- Order of mapping steps is chosen to optimize search

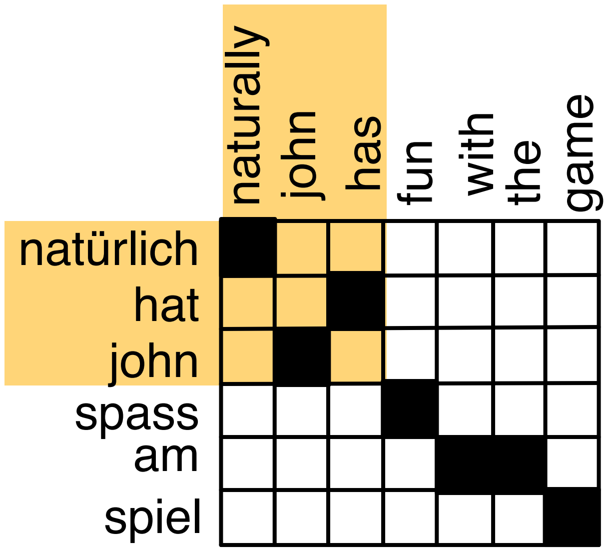
Phrase-based training

- Establish word alignment (GIZA++ and symmetrization)



Phrase-based training

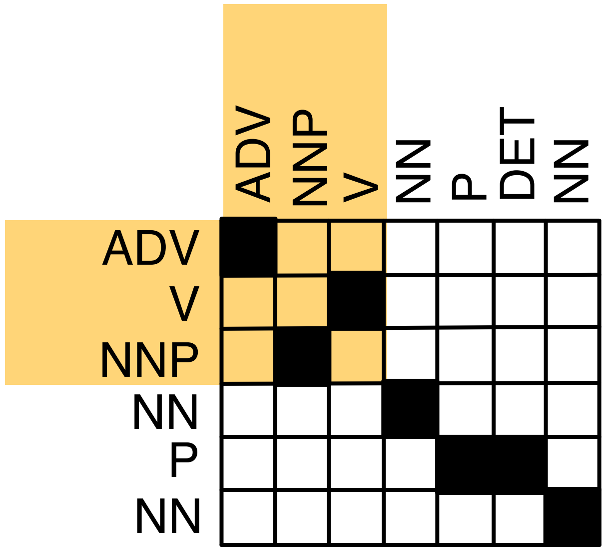
- Extract phrase



⇒ *natürlich hat john* — *naturally john has*

Factored training

- Annotate training with factors, extract phrase



⇒ *ADV V NNP* — *ADV NNP V*

Training of generation steps

- Generation steps map target factors to target factors
  – typically trained on target side of parallel corpus
  – may be trained on additional monolingual data
- Example: *The*/DET *man*/NN *sleeps*/VBZ
  – count collection
    - count(*the*,DET)++
    - count(*man*,NN)++
    - count(*sleeps*,VBZ)++
  – evidence for probability distributions (max. likelihood estimation)
    - p(DET|*the*), p(*the*|DET)
    - p(NN|*man*), p(*man*|NN)
    - p(VBZ|*sleeps*), p(*sleeps*|VBZ)

Factored Translation Models

- Motivation
- Example
- Model and Training
- **Decoding**
- Experiments

Phrase-based translation

- Task: *translate this sentence* from German into English

er	**geht**	**ja**	**nicht**	**nach**	**hause**

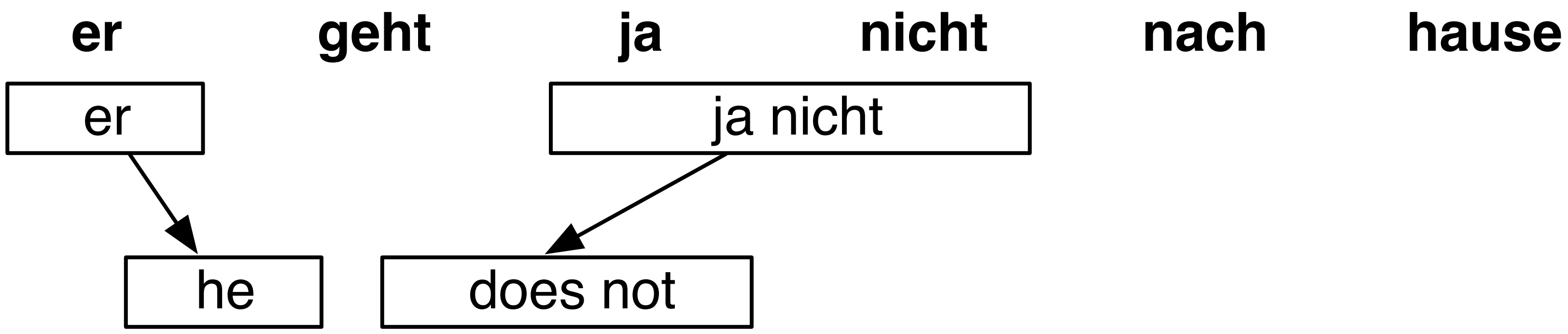
Translation step 1

- Task: translate this sentence from German into English

- *Pick* phrase in input, *translate*

Translation step 2

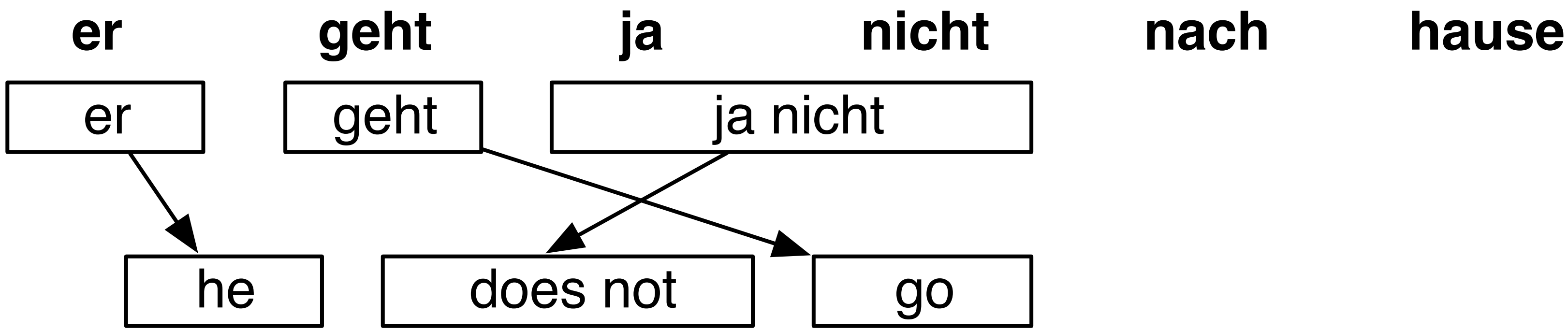
- Task: translate this sentence from German into English



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

Translation step 3

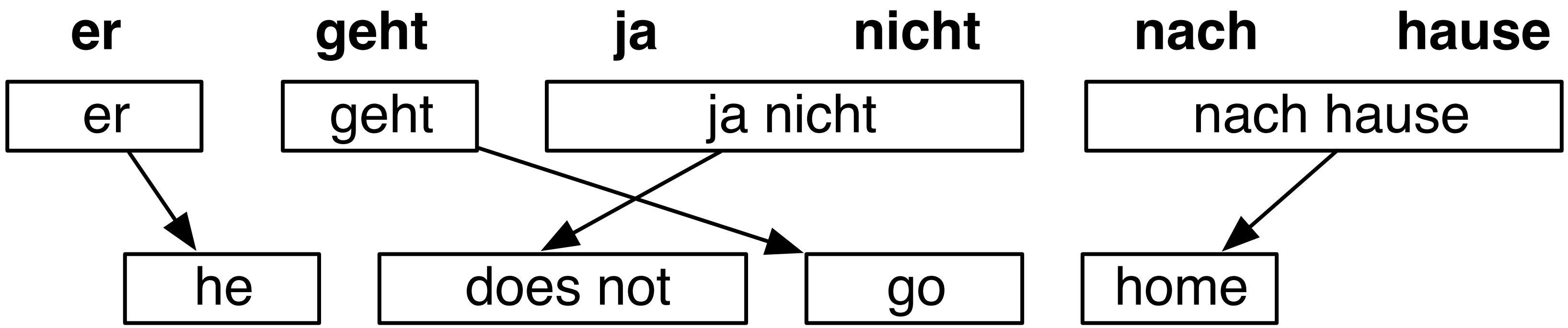
- Task: translate this sentence from German into English



- Pick phrase in input, translate

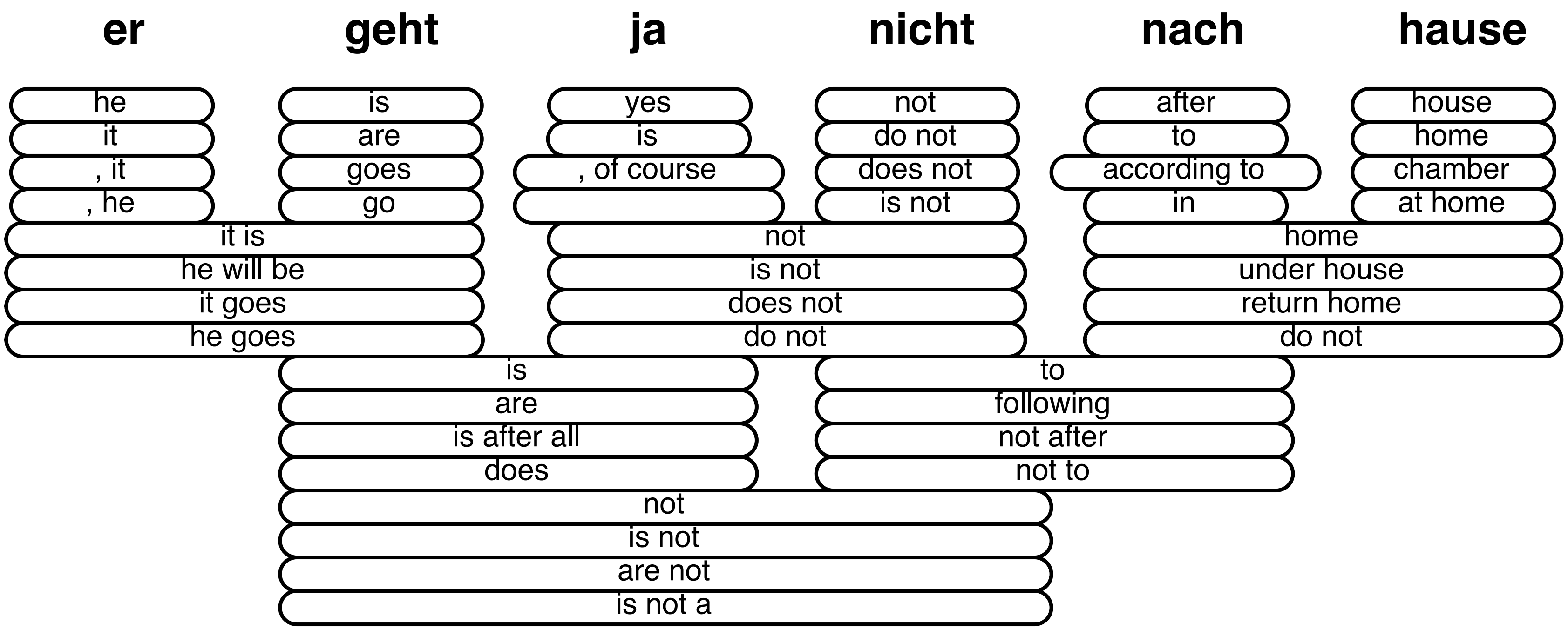
Translation step 4

- Task: translate this sentence from German into English



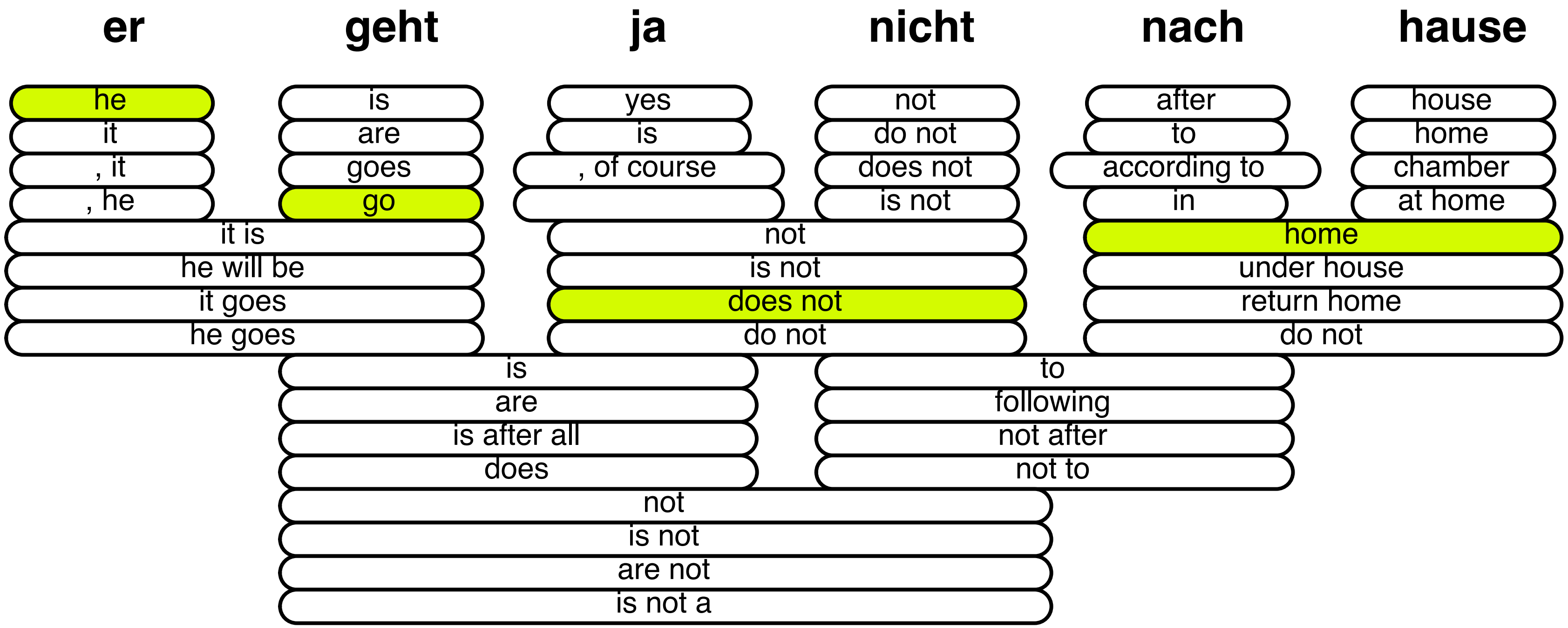
- Pick phrase in input, translate

Translation options



- *Many translation options* to choose from

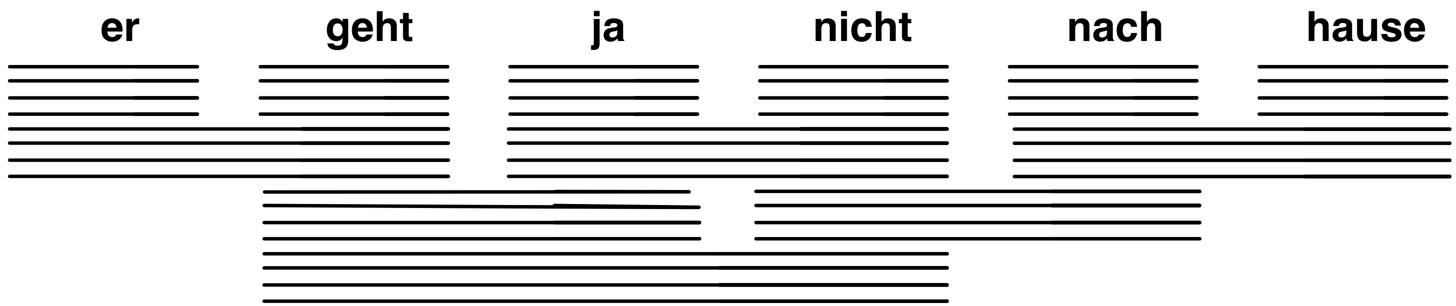
Translation options



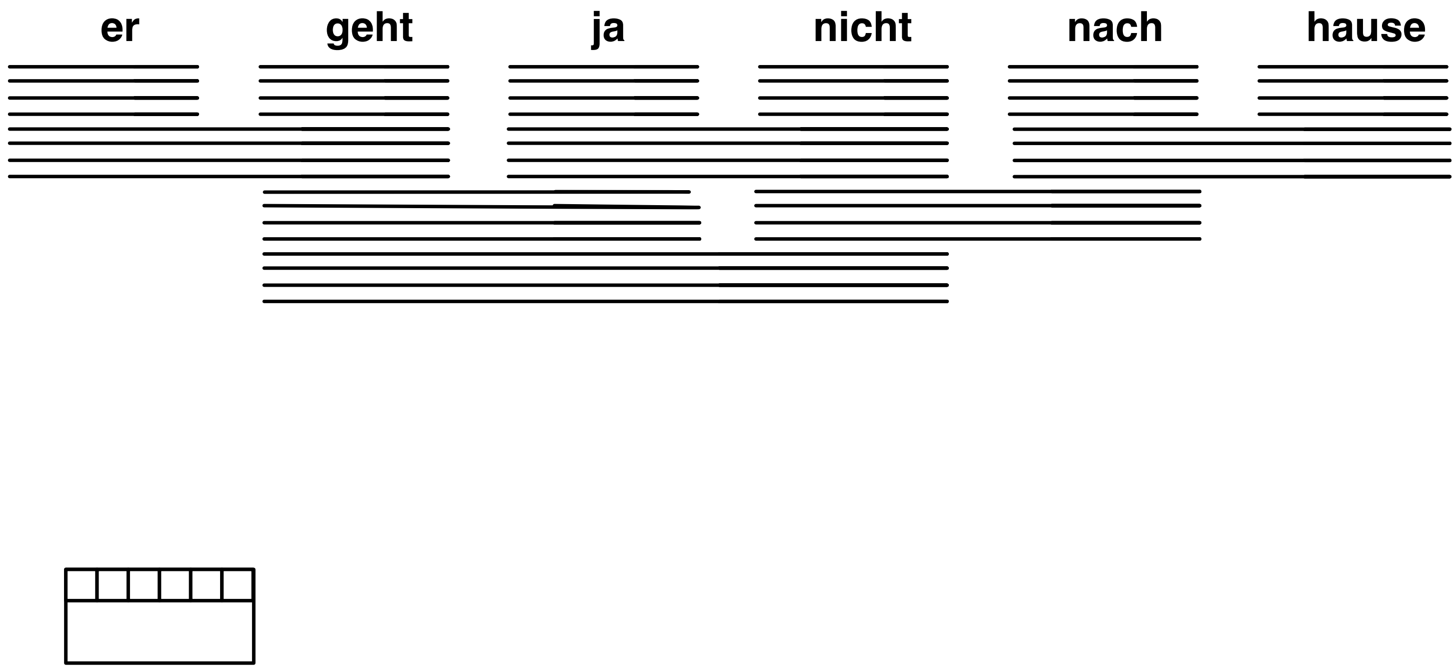
- The machine translation decoder does not know the right answer

→ *Search problem* solved by heuristic beam search

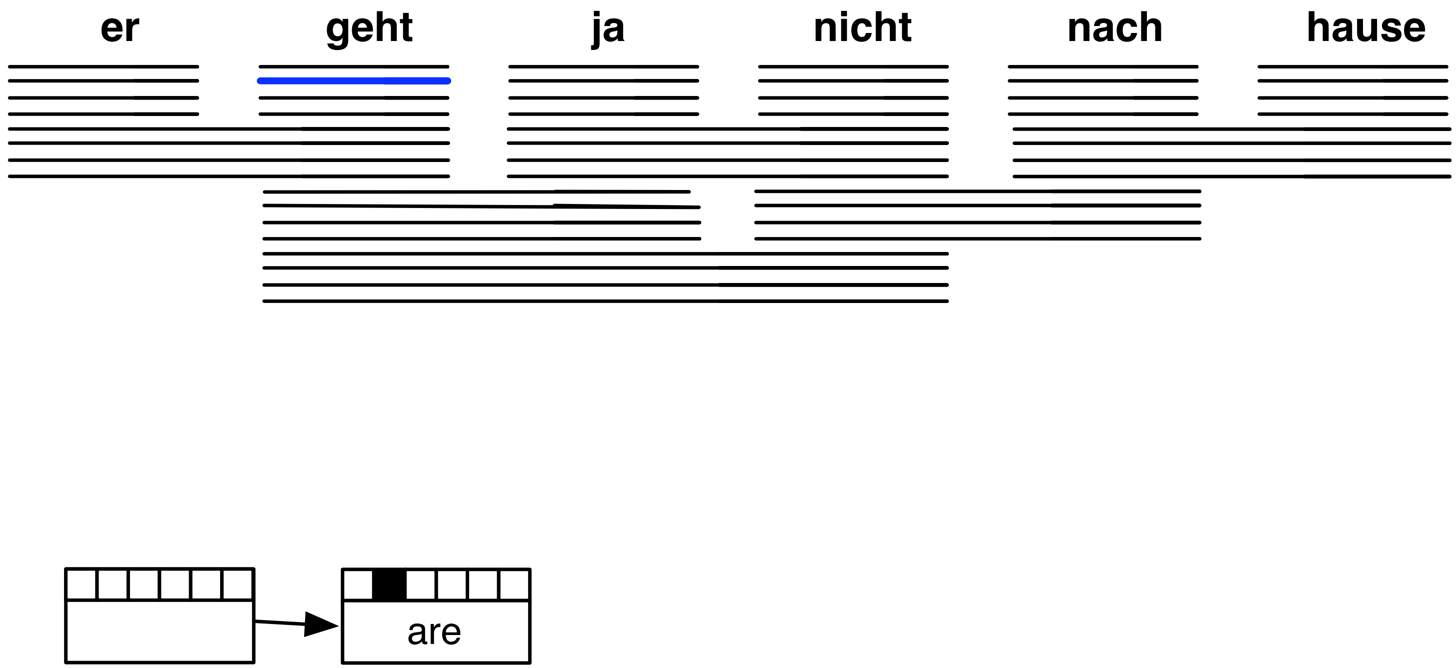
Decoding process: precompute translation options



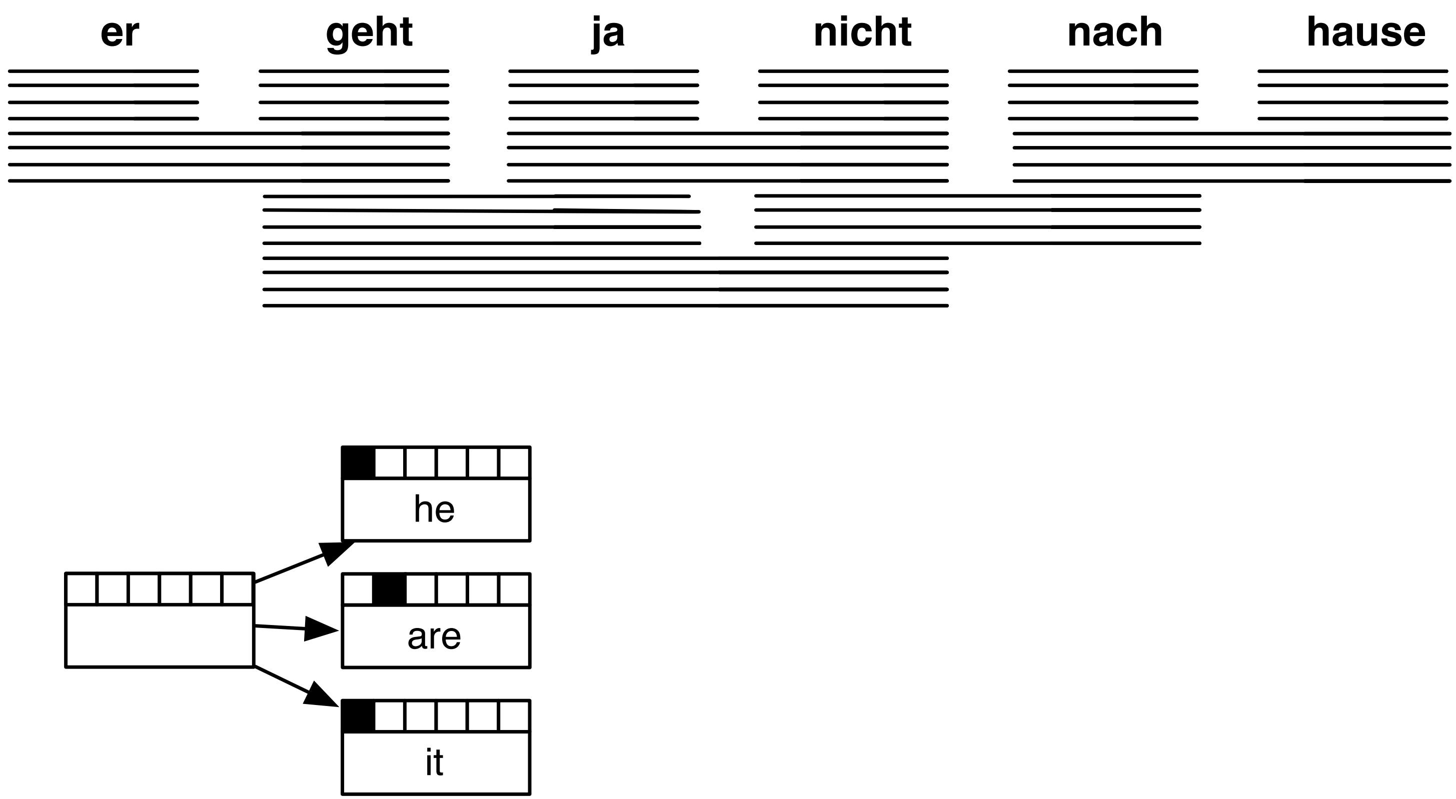
Decoding process: start with initial hypothesis



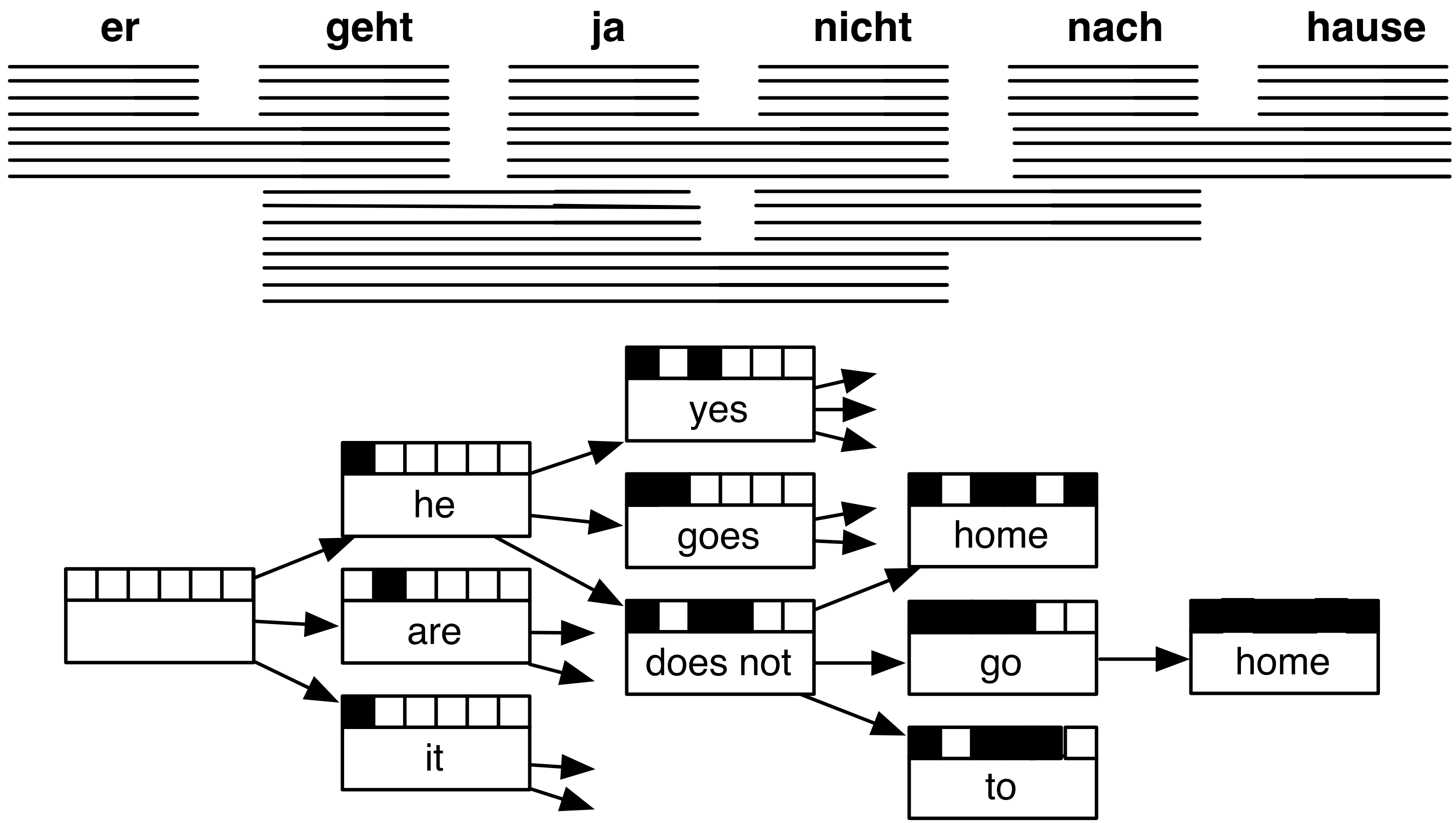
Decoding process: hypothesis expansion



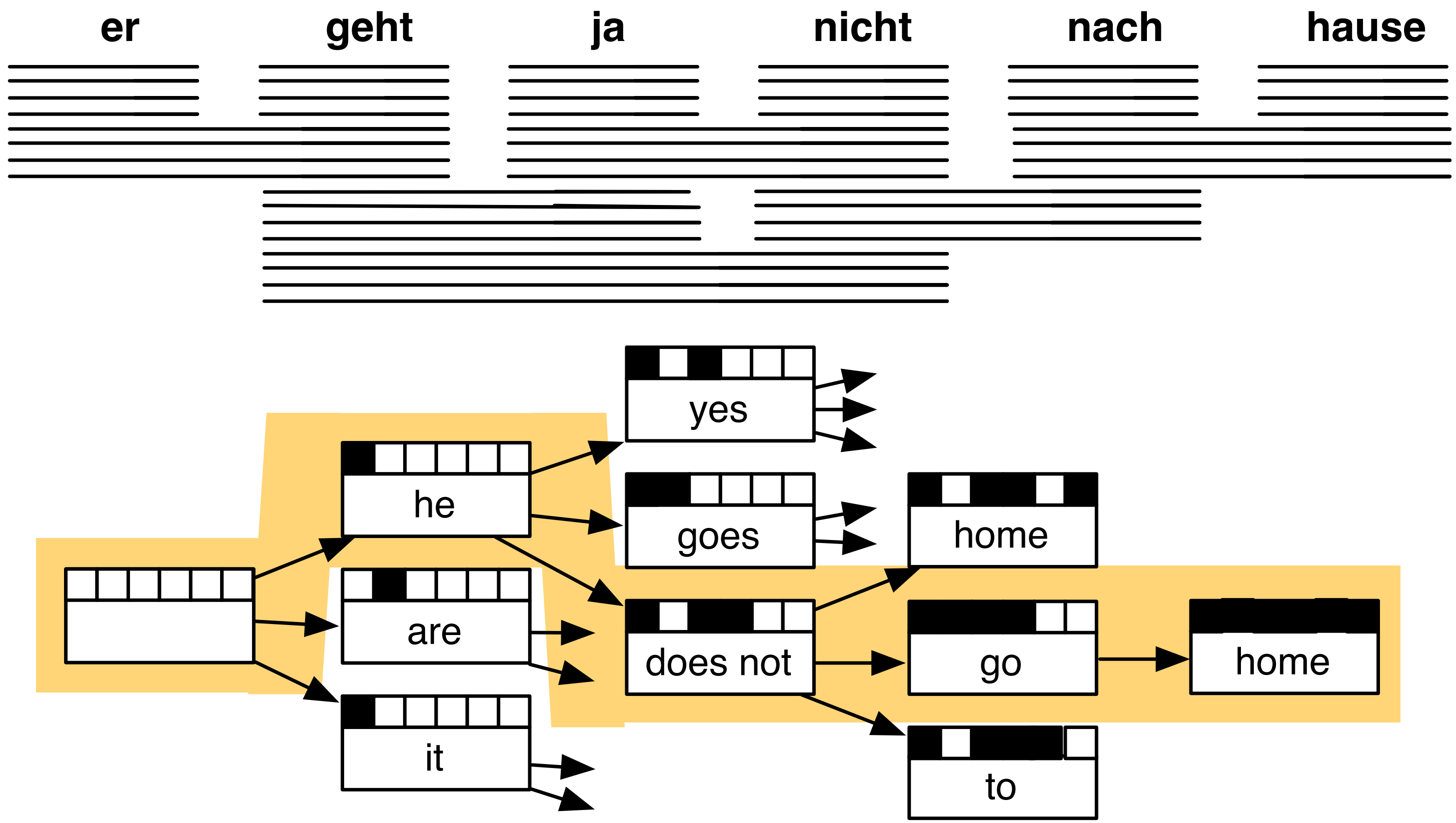
Decoding process: hypothesis expansion



Decoding process: hypothesis expansion



Decoding process: find best path

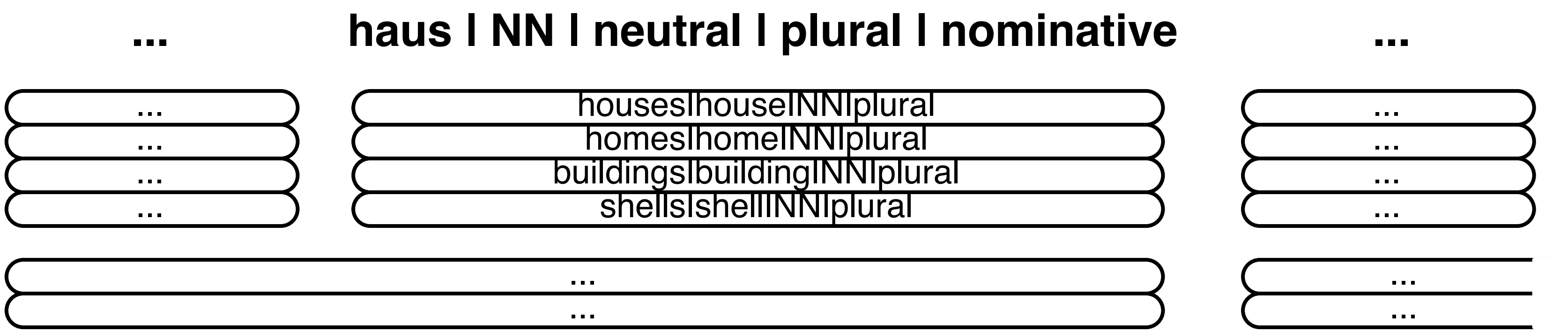


Factored model decoding

- Factored model decoding introduces *additional complexity*
- Hypothesis expansion not any more according to simple translation table, but by *executing a number of mapping steps*, e.g.:
  1. translating of *lemma* → *lemma*
  2. translating of *part-of-speech, morphology* → *part-of-speech, morphology*
  3. generation of *surface form*
- Example: *haus|NN|neutral|plural|nominative*
  → { *houses|house|NN|plural, homes|home|NN|plural, buildings|building|NN|plural, shells|shell|NN|plural* }
- Each time, a hypothesis is expanded, these mapping steps have to applied

Efficient factored model decoding

- Key insight: executing of mapping steps can be *pre-computed* and stored as translation options
  - apply mapping steps to all input phrases
  - store results as *translation options*
- → decoding algorithm *unchanged*



Efficient factored model decoding

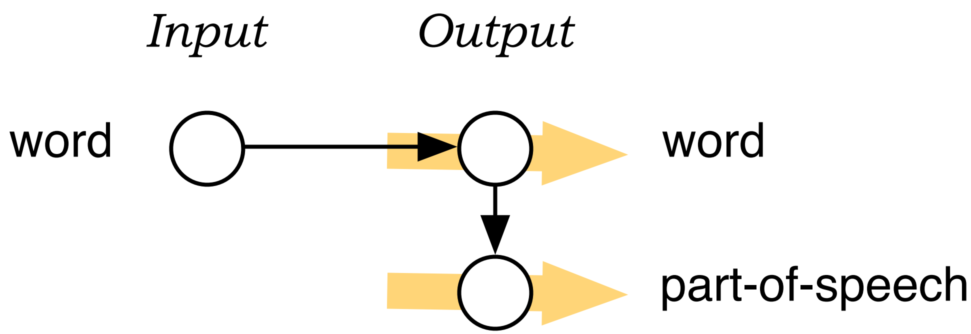
- Problem: *Explosion* of translation options
  - originally limited to 20 per input phrase
  - even with simple model, now 1000s of mapping expansions possible

- Solution: *Additional pruning* of translation options
  - *keep only the best* expanded translation options
  - current default 50 per input phrase
  - decoding only about 2-3 times slower than with surface model

Factored Translation Models

- Motivation
- Example
- Model and Training
- Decoding
- **Experiments**
- Outlook

Adding linguistic markup to output



- Generation of POS tags on the target side
- Use of high order language models over POS (7-gram, 9-gram)
- Motivation: syntactic tags should enforce syntactic sentence structure model not strong enough to support major restructuring

Some experiments

- English–German, Europarl, 30 million word, test2006

Model	BLEU
best published result	18.15
baseline (surface)	18.04
surface + POS	18.15

- German–English, News Commentary data (WMT 2007), 1 million word

Model	BLEU
Baseline	18.19
With POS LM	19.05

- Improvements under sparse data conditions
- Similar results with CCG supertags [Birch et al., 2007]

Sequence models over morphological tags

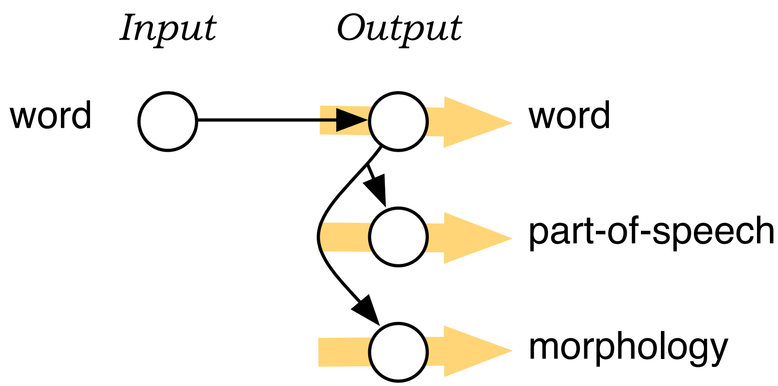
die	hellen	Sterne	erleuchten	das	schwarze	Himmel
(the)	*(bright)*	*(stars)*	*(illuminate)*	*(the)*	*(black)*	*(sky)*
fem	*fem*	*fem*	*-*	*neutral*	*neutral*	*male*
plural	*plural*	*plural*	*plural*	*sgl.*	*sgl.*	*sgl*
nom.	*nom.*	*nom.*	*-*	*acc.*	*acc.*	*acc.*

- Violation of noun phrase agreement in gender
  - *das schwarze* and *schwarze Himmel* are perfectly fine bigrams
  - but: *das schwarze Himmel* is not

- If relevant n-grams does not occur in the corpus, a lexical n-gram model would *fail to detect* this mistake

- Morphological sequence model: $p(\text{N-male}|\text{J-male}) > p(\text{N-male}|\text{J-neutral})$

Local agreement (esp. within noun phrases)



- High order language models over POS and morphology
- Motivation
  - *DET-sgl NOUN-sgl* good sequence
  - *DET-sgl NOUN-plural* bad sequence

Agreement within noun phrases

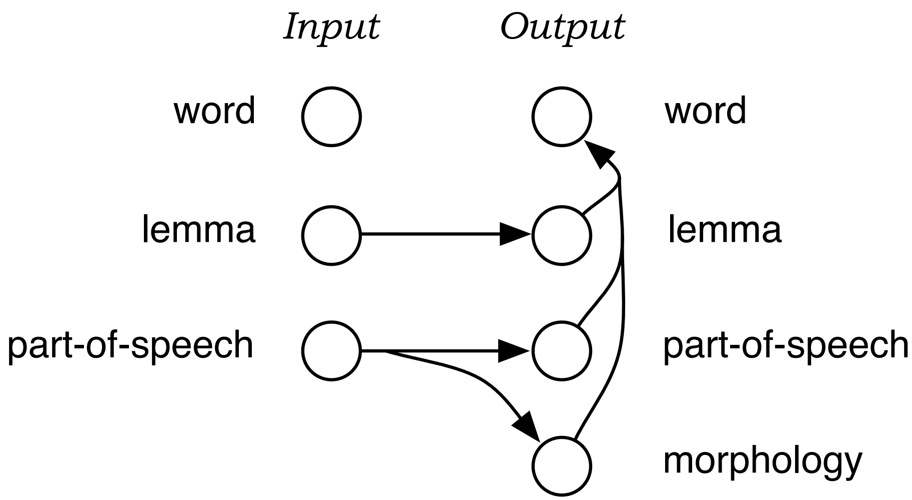
- Experiment: 7-gram POS, morph LM in addition to 3-gram word LM
- Results

Method	Agreement errors in NP	devtest	test
baseline	15% in NP $\geq$ 3 words	18.22 BLEU	18.04 BLEU
factored model	4% in NP $\geq$ 3 words	18.25 BLEU	18.22 BLEU

- Example
  - baseline: *... zur zwischenstaatlichen methoden ...*
  - factored model: *... zu zwischenstaatlichen methoden ...*

- Example
  - baseline: *... das zweite wichtige änderung ...*
  - factored model: *... die zweite wichtige änderung ...*

Morphological generation model



- Our motivating example
- Translating lemma and morphological information more robust

Initial results

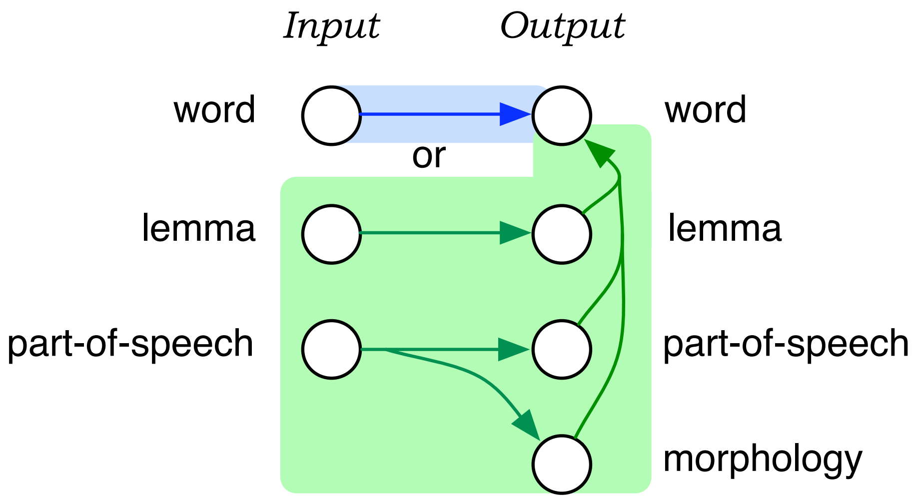
- Results on 1 million word News Commentary corpus (German–English)

System	In-doman	Out-of-domain
Baseline	18.19	15.01
With POS LM	19.05	15.03
Morphgen model	14.38	11.65

- What went wrong?
  – why back-off to lemma, when we know how to translate surface forms?

$\rightarrow$ loss of information

Solution: alternative decoding paths



- Allow both surface form translation and morphgen model
  - prefer surface model for known words
  - morphgen model acts as back-off

Results

- Model now beats the baseline:

System	In-doman	Out-of-domain
Baseline	**18.19**	**15.01**
With POS LM	19.05	15.03
Morphgen model	14.38	11.65
Both model paths	**19.47**	**15.23**

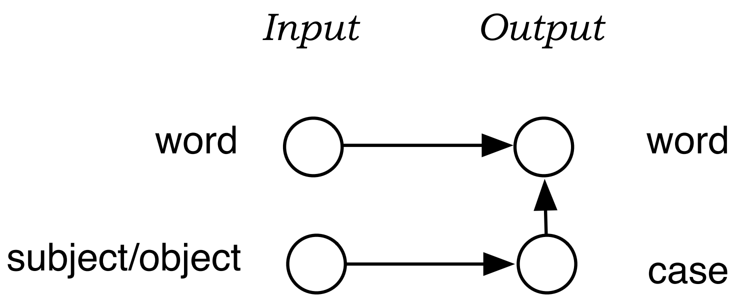
Adding annotation to the source

- Source words may **lack sufficient information** to map phrases
  - English-German: what case for noun phrases?
  - Chinese-English: plural or singular
  - pronoun translation: what do they refer to?

- Idea: **add additional information** to the source that makes the required information available locally (where it is needed)

- see [Avramidis and Koehn, ACL 2008] for details

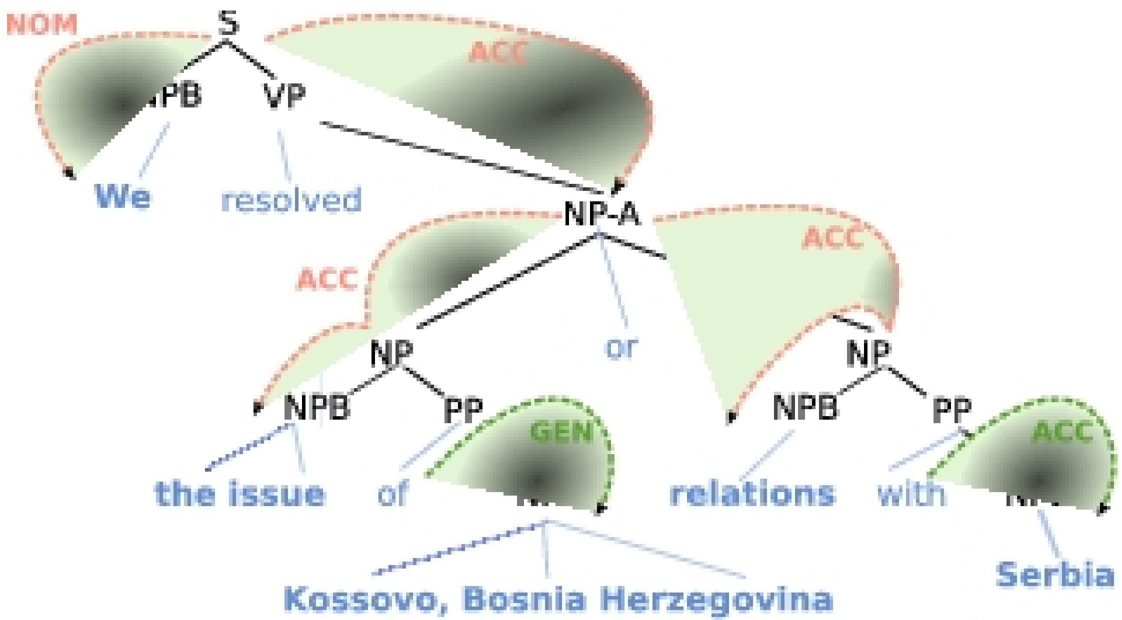
Case Information for English–Greek



- Detect in English, if noun phrase is subject/object (using parse tree)
- Map information into case morphology of Greek
- Use case morphology to generate correct word form

Obtaining Case Information

- Use syntactic parse of English input
  (method similar to semantic role labeling)



Results English-Greek

- Automatic BLEU scores

System	devtest	test07
baseline	18.13	18.05
enriched	18.21	18.20

- Improvement in verb inflection

System	Verb count	Errors	Missing
baseline	311	19.0%	7.4%
enriched	294	5.4%	2.7%

- Improvement in noun phrase inflection

System	NPs	Errors	Missing
baseline	247	8.1%	3.2%
enriched	239	5.0%	5.0%

- Also successfully applied to English-Czech

Discriminative Training

Overview

- Evolution from generative to discriminative models
  - IBM Models: purely generative
  - MERT: discriminative training of generative components
  - More features $\rightarrow$ better discriminative training needed
- Perceptron algorithm
- Problem: overfitting
- Problem: matching reference translation

The birth of SMT: generative models

- The definition of translation probability follows a **mathematical derivation**

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

- Occasionally, some **independence assumptions** are thrown in
  for instance IBM Model 1: word translations are independent of each other

$$p(\mathbf{e}|\mathbf{f}, a) = \frac{1}{Z} \prod_i p(e_i | f_{a(i)})$$

- Generative story leads to **straight-forward estimation**
  - maximum likelihood estimation of component probability distribution
  - **EM algorithm** for discovering hidden variables (alignment)

Log-linear models

- IBM Models provided mathematical justification for factoring **components** together

$$p_{LM} \times p_{TM} \times p_D$$

- These may be **weighted**

$$p_{LM}^{\lambda_{LM}} \times p_{TM}^{\lambda_{TM}} \times p_D^{\lambda_D}$$

- **Many components** p_i with weights λ_i

$$\prod_i p_i^{\lambda_i} = exp(\sum_i \lambda_i log(p_i))$$

$$log \prod_i p_i^{\lambda_i} = \sum_i \lambda_i log(p_i)$$

Knowledge sources

- Many different **knowledge sources** useful
  - language model
  - reordering (distortion) model
  - phrase translation model
  - word translation model
  - word count
  - phrase count
  - drop word feature
  - phrase pair frequency
  - additional language models
  - additional features

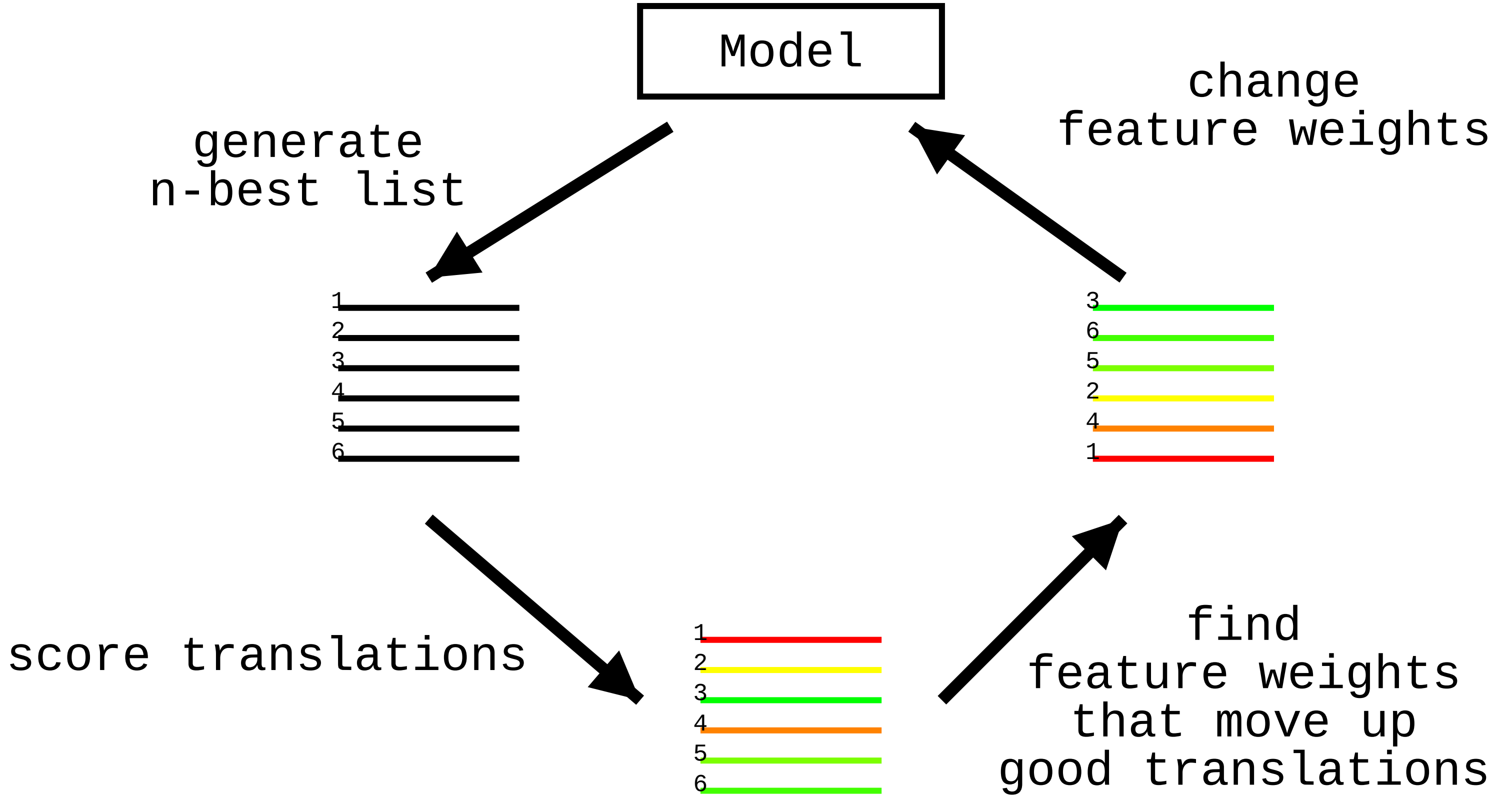
Set feature weights

- Contribution of components p_i determined by weight λ_i

- Methods
  - *manual setting* of weights: try a few, take best
  - *automate* this process

- Learn weights
  - set aside a **development corpus**
  - set the weights, so that **optimal translation performance** on this development corpus is achieved
  - requires *automatic scoring* method (e.g., BLEU)

Discriminative training



Discriminative vs. generative models

- Generative models
  - translation process is broken down to *steps*
  - each step is modeled by a *probability distribution*
  - each probability distribution is estimated from the data by *maximum likelihood*

- Discriminative models
  - model consist of a number of *features* (e.g. the language model score)
  - each feature has a *weight*, measuring its value for judging a translation as correct
  - feature weights are *optimized on development data*, so that the system output matches correct translations as close as possible

Discriminative training

- Training set (*development set*)
  - different from original training set
  - small (maybe 1000 sentences)
  - must be different from test set
- Current model *translates* this development set
  - *n-best list* of translations (n=100, 10000)
  - translations in n-best list can be *scored*
- Feature weights are *adjusted*
- N-Best list generation and feature weight adjustment repeated for a number of iterations

Learning task

- Task: *find weights*, so that feature vector of the correct translations *ranked first*

rank	translation	LM	TM	WP	SER
1	Mary not give slap witch green .	-17.2	-5.2	-7	1
2	Mary not slap the witch green .	-16.3	-5.7	-7	1
3	Mary not give slap of the green witch .	-18.1	-4.9	-9	1
4	Mary not give of green witch .	-16.5	-5.1	-8	1
5	Mary did not slap the witch green .	-20.1	-4.7	-8	1
6	Mary did not slap green witch .	-15.5	-3.2	-7	1
7	Mary not slap of the witch green .	-19.2	-5.3	-8	1
8	Mary did not give slap of witch green .	-23.2	-5.0	-9	1
9	Mary did not give slap of the green witch .	-21.8	-4.4	-10	1
10	Mary did slap the witch green .	-15.5	-6.9	-7	1
11	**Mary did not slap the green witch .**	**-17.4**	**-5.3**	**-8**	**0**
12	Mary did slap witch green .	-16.9	-6.9	-6	1
13	Mary did slap the green witch .	-14.3	-7.1	-7	1
14	Mary did not slap the of green witch .	-24.2	-5.3	-9	1
15	Mary did not give slap the witch green .	-25.2	-5.5	-9	1

(LM, TM, WP: feature vector)

Och's minimum error rate training (MERT)

- **Line search** for best feature weights

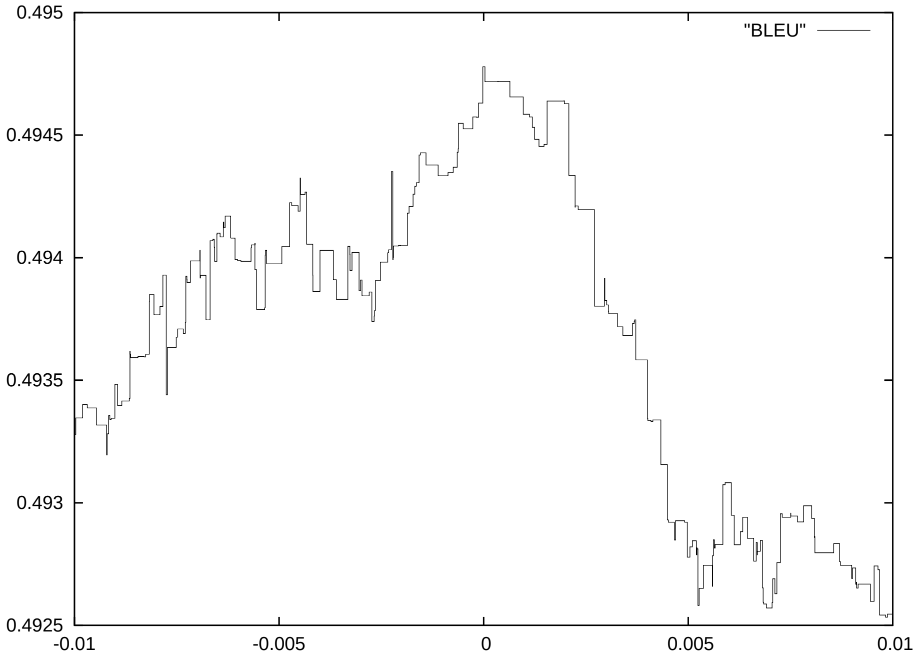
```
given:  sentences with n-best list of
translations
iterate n times
    randomize starting feature weights
        iterate until convergences
            for each feature
                find best feature weight
                update if different from current
return best feature weights found in any
iteration
```

Methods to adjust feature weights

- **Maximum entropy** [Och and Ney, ACL2002]
  - match *expectation* of feature values of model and data
- **Minimum error rate** training [Och, ACL2003]
  - try to *rank best translations first* in n-best list
  - can be adapted for various error metrics, even BLEU
- **Ordinal regression** [Shen et al., NAACL2004]
  - *separate* k worst from the k best translations

BLEU error surface

- Varying one parameter: a rugged line with many local optima



Unstable outcomes: weights vary

component	run 1	run 2	run 3	run 4	run 5	run 6
distance	0.059531	0.071025	0.069061	0.120828	0.120828	0.072891
lexdist 1	0.093565	0.044724	0.097312	0.108922	0.108922	0.062848
lexdist 2	0.021165	0.008882	0.008607	0.013950	0.013950	0.030890
lexdist 3	0.083298	0.049741	0.024822	-0.000598	-0.000598	0.023018
lexdist 4	0.051842	0.108107	0.090298	0.111243	0.111243	0.047508
lexdist 5	0.043290	0.047801	0.020211	0.028672	0.028672	0.050748
lexdist 6	0.083848	0.056161	0.103767	0.032869	0.032869	0.050240
lm 1	0.042750	0.056124	0.052090	0.049561	0.049561	0.059518
lm 2	0.019881	0.012075	0.022896	0.035769	0.035769	0.026414
lm 3	0.059497	0.054580	0.044363	0.048321	0.048321	0.056282
ttable 1	0.052111	0.045096	0.046655	0.054519	0.054519	0.046538
ttable 1	0.052888	0.036831	0.040820	0.058003	0.058003	0.066308
ttable 1	0.042151	0.066256	0.043265	0.047271	0.047271	0.052853
ttable 1	0.034067	0.031048	0.050794	0.037589	0.037589	0.031939
phrase-pen.	0.059151	0.062019	-0.037950	0.023414	0.023414	-0.069425
word-pen	-0.200963	-0.249531	-0.247089	-0.228469	-0.228469	-0.252579

Unstable outcomes: scores vary

- Even different scores with different runs (varying 0.40 on dev, 0.89 on test)

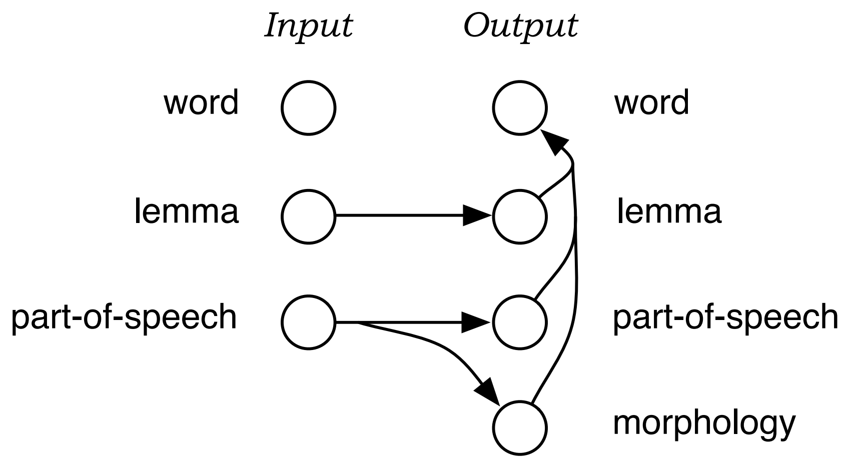
run	iterations	dev score	test score
1	8	50.16	51.99
2	9	50.26	51.78
3	8	50.13	51.59
4	12	50.10	51.20
5	10	50.16	51.43
6	11	50.02	51.66
7	10	50.25	51.10
8	11	50.21	51.32
9	10	50.42	51.79

More features: more components

- We would like to add **more components** to our model
  - multiple language models
  - domain adaptation features
  - various special handling features
  - using linguistic information

→ MERT becomes even **less reliable**
  - runs many more iterations
  - fails more frequently

More features: factored models



- Factored translation models break up phrase mapping into smaller steps
  - multiple translation tables
  - multiple generation tables
  - multiple language models and sequence models on factors

→ **Many more features**

Millions of features

- Why **mix** of discriminative training and generative models?
- Discriminative training of all components
  - phrase table [Liang et al., 2006]
  - language model [Roark et al, 2004]
  - additional features
- **Large-scale** discriminative training
  - millions of features
  - training of full training set, not just a small development corpus

Perceptron algorithm

- Translate each sentence
- If no match with reference translation: update features

```
set all lambda = 0
do until convergence
    for all foreign sentences f
        set e-best to best translation according to model
        set e-ref to reference translation
        if e-best != e-ref
            for all features feature-i
                lambda-i += feature-i(f,e-ref)
                          - feature-i(f,e-best)
```

Problem: overfitting

- Fundamental problem in machine learning
  - what works best for training data, may not work well in general
  - **rare, unrepresentative features** may get too much weight

- **Especially severe problem** in phrase-based models
  - **long phrase pairs** explain well *individual sentences*
  - ... but are less general, *suspect to noise*
  - EM training of phrase models [Marcu and Wong, 2002] has same problem

Solutions

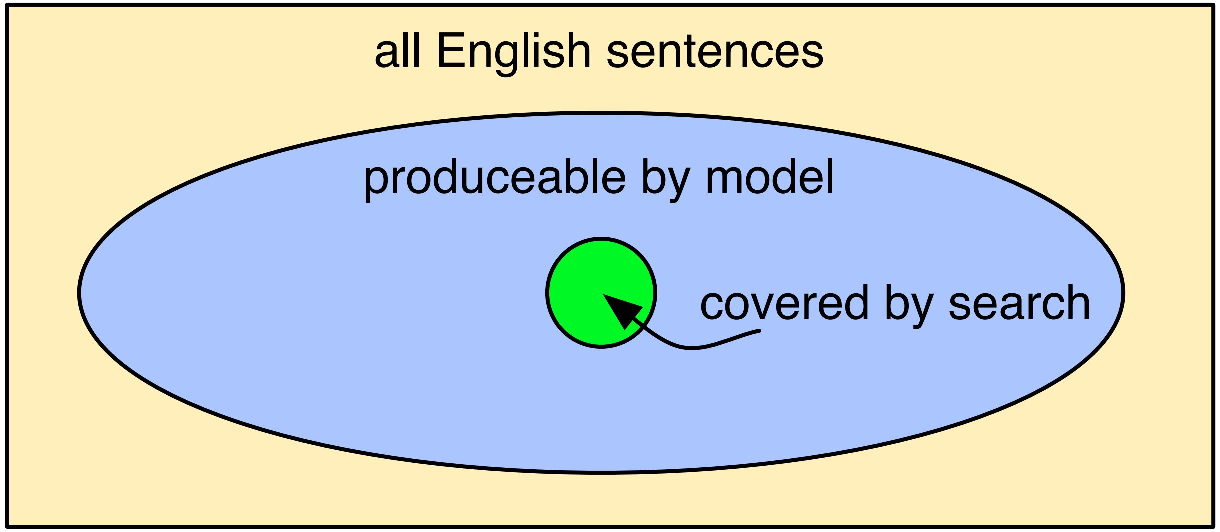
- **Restrict to short phrases**, e.g., maximum 3 words (current approach)
  - limits the power of phrase-based models
  - ... but not very much [Koehn et al, 2003]

- **Jackknife**
  - collect phrase pairs from one part of corpus
  - optimize their feature weights on another part

- IBM direct model: **only one-to-many** phrases [Ittycheriah and Salim Roukos, 2007]

Problem: reference translation

- Reference translation may be anywhere in this box



- If produceable by model $\longrightarrow$ we can compute feature scores
- If not $\longrightarrow$ we can not

Some solutions

- **Skip sentences**, for which reference can not be produced
  - invalidates large amounts of training data
  - biases model to shorter sentences
- Declare candidate translations closest to reference as **surrogate**
  - closeness measured for instance by smoothed BLEU score
  - may be not a very good translation: odd feature values, training is severely distorted

Better solution: early updating?

- At some point the reference translation **falls out** of the search space
  - for instance, due to *unknown words*:

Reference: `The group attended the meeting in Najaf ...`

System: `The group meeting was attended in UNKNOWN ...`

only update features involved in this part

- Early updating [Collins et al., 2005]:
  - stop search, when reference translation is not covered by model
  - only update **features involved in partial** reference / system output

Conclusions

- Currently have proof-of-concept implementation
- Future work: Overcome various technical challenges
  - reference translation may not be produceable
  - overfitting
  - mix of binary and real-valued features
  - scaling up
- More and more features are unavoidable, let's deal with them