

---

# Statistical Machine Translation

## Part 1: Morning Session

Philipp Koehn

10 September 2007





## Before we begin...

- What is about to happen?
  - a journey through the methods of SMT systems
  - focus mostly on the (very) current
  - there will be some maths
- What is **not** about to happen?
  - a guide on how to use statistical machine translation
  - an introduction to tools used in statistical machine translation

---

# Topics

- Philipp Koehn (morning)
  - Introduction
  - Word-based models and the EM algorithm
  - Decoding
  - Phrase-based models
- Kevin Knight (afternoon)
  - Syntax-based statistical MT
  - Learning syntax models from data
  - Decoding for syntax models
  - Tree Automata
- This will take a while...

---

## Fair warning

- Quotes:

*It must be recognized that the notion "probability of a sentence" is an entirely useless one, under any known interpretation of this term.*

Noam Chomsky, 1969

*Whenever I fire a linguist our system performance improves.*

Frederick Jelinek, 1988

- Warning: We may agree more with Jelinek than Chomsky (well, at least we know people who do)

# Machine translation

- Task: make sense of foreign text like

## 毒品

本冊子為家長們提供實際和有用的關於毒品的信息，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。這本小冊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

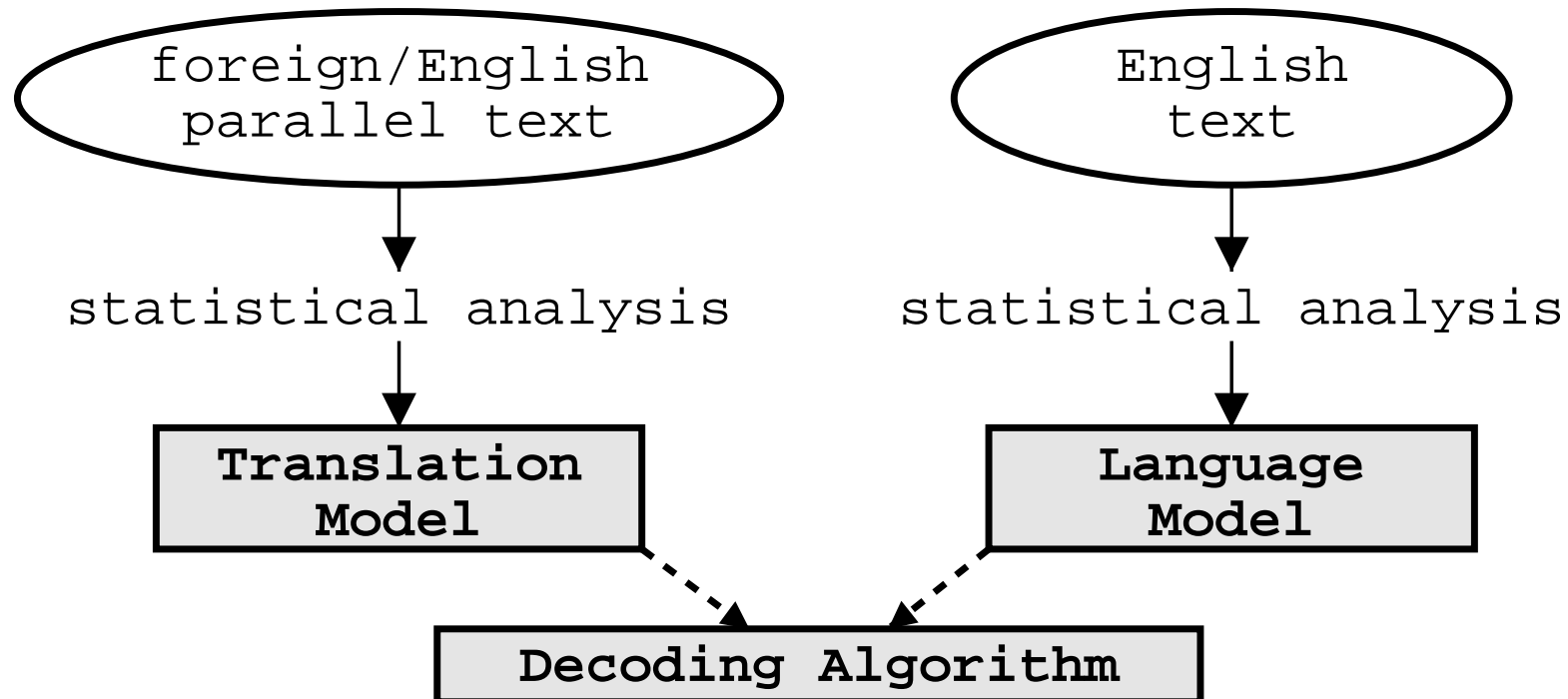
---

## 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
    - 3.5 million words a year
    - *300 million a lifetime*
    - soon computers will be able to see more translated text than humans read in a lifetime
- ⇒ Machine *can learn* how to translated foreign languages

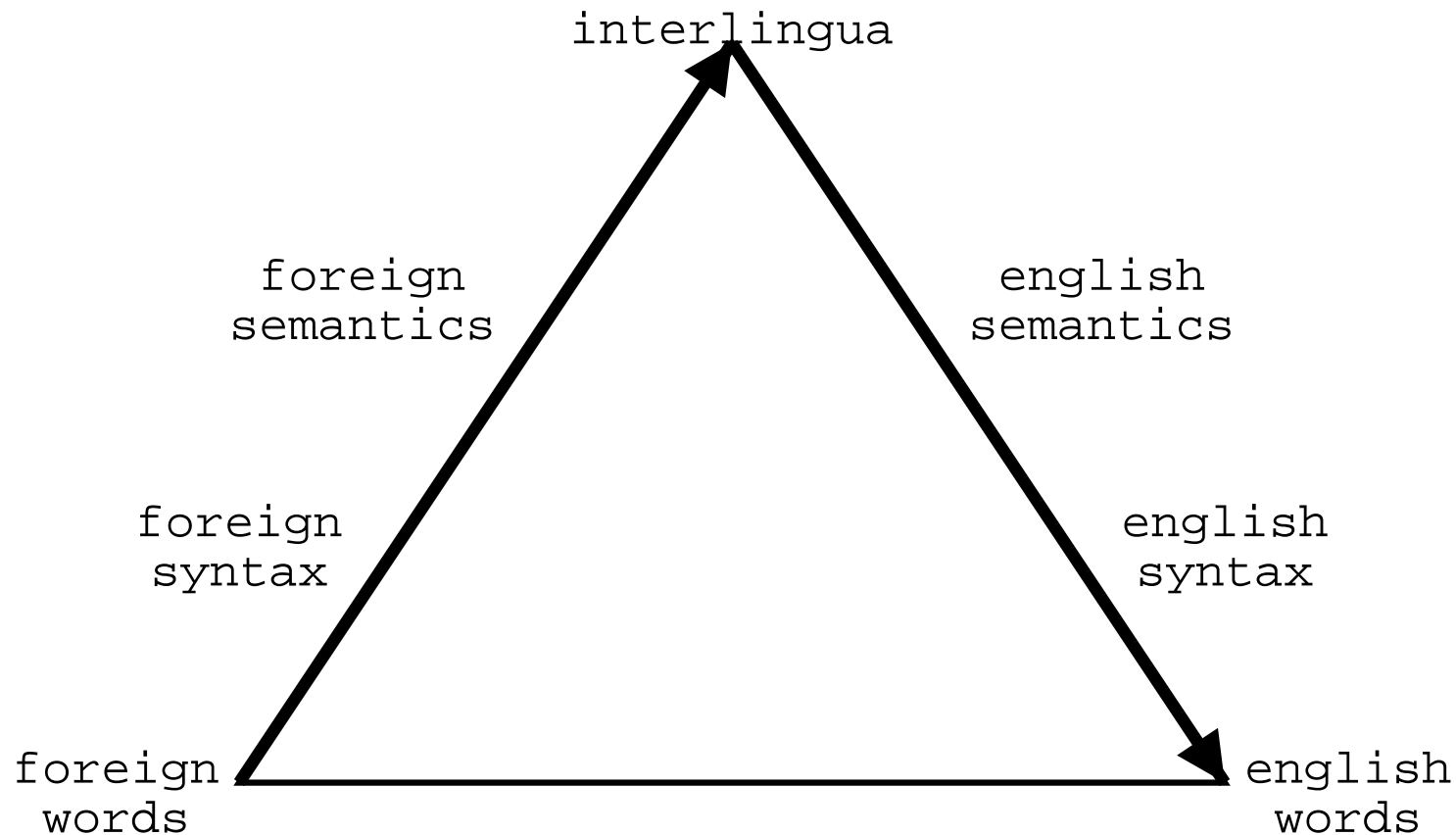
# Statistical machine translation

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

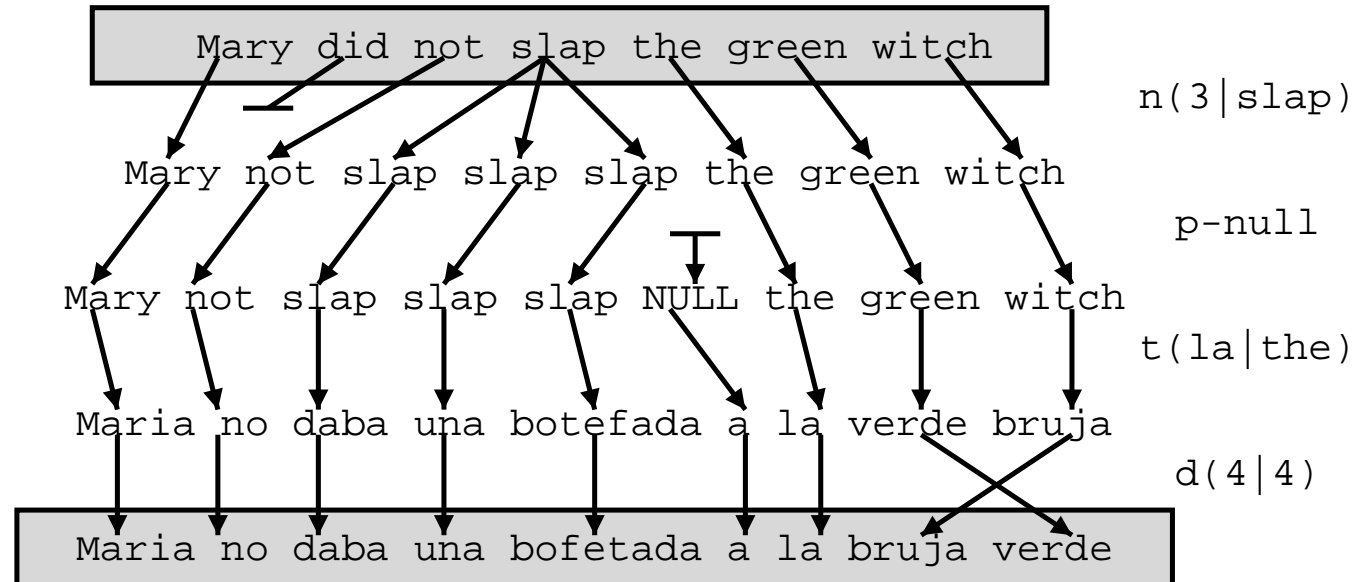




# The machine translation pyramid



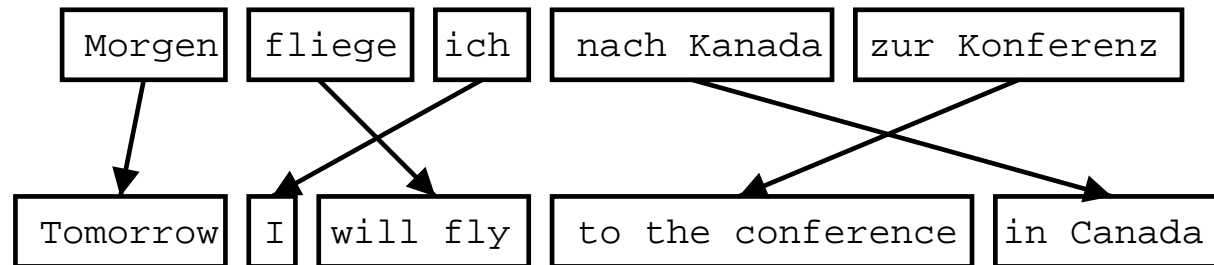
## Word-based models



[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]

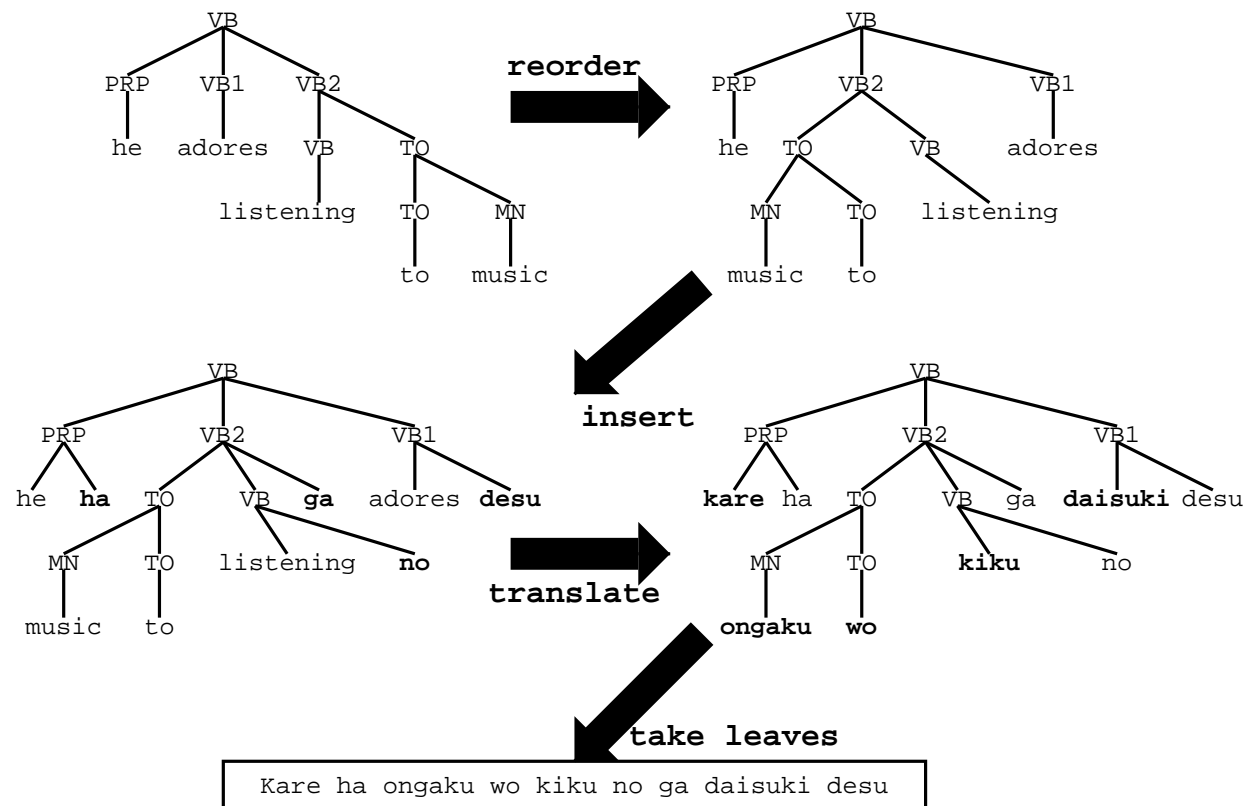
## Phrase-based models



[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

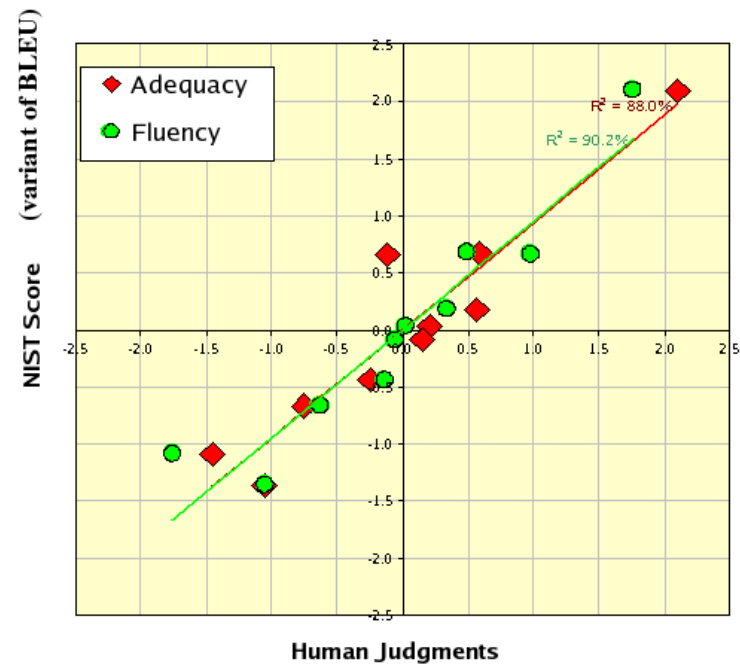
- Why **automatic evaluation** metrics?
  - Manual evaluation is *too slow*
  - Evaluation on large test sets *reveals minor improvements*
  - **Automatic tuning** to improve machine translation performance
- History
  - Word Error Rate
  - **BLEU** since 2002
- BLEU in short: *Overlap with reference* 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

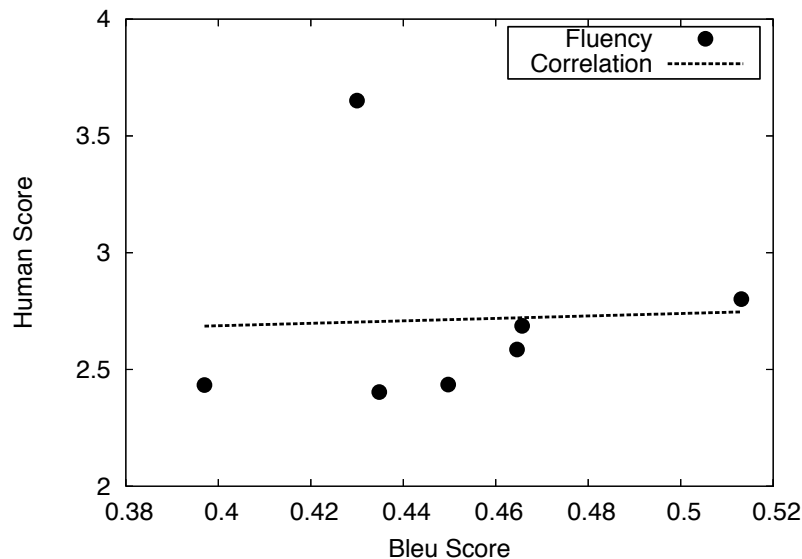
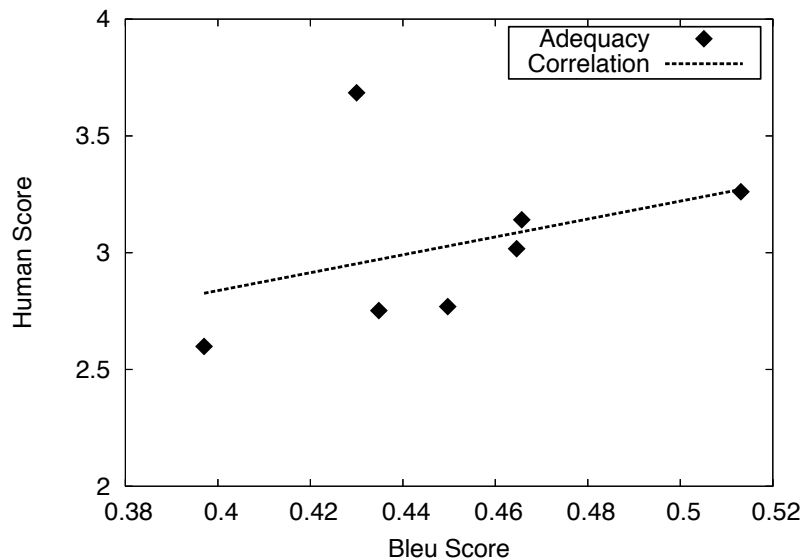


[from George Doddington, NIST]

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



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

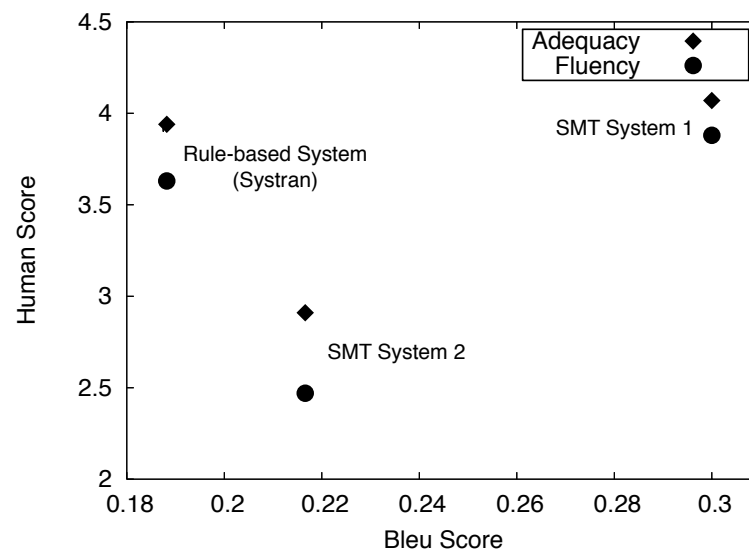


[from Callison-Burch et al., 2006, EACL]

- 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]



- Comparison of

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

- *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]





## Translate into vs. out of a language

- Some languages are *easier* to translate into than out of

Language	From	Into	Diff
da	23.4	23.3	0.0
<b>de</b>	<b>22.2</b>	<b>17.7</b>	<b>-4.5</b>
el	23.8	22.9	-0.9
<b>en</b>	<b>23.8</b>	<b>27.4</b>	<b>+3.6</b>
es	26.7	29.6	+2.9
fr	26.1	31.1	+5.1
fi	19.1	12.4	-6.7
it	24.3	25.4	+1.1
nl	19.7	20.7	+1.1
pt	26.1	27.0	+0.9
sv	24.8	22.1	-2.6

[from Koehn, 2005: Europarl]

- Morphologically rich languages* harder to generate (German, Finnish)

---

# Backtranslations

- Checking translation quality by **back-transliteration**
- *The spirit is willing, but the flesh is weak*
- English → Russian → English
- *The vodka is good but the meat is rotten*





## Backtranslations II

- *Does not correlate* with unidirectional performance

Language	From	Into	Back
da	28.5	25.2	56.6
de	25.3	17.6	48.8
el	27.2	23.2	<b>56.5</b>
es	30.5	30.1	52.6
fi	21.8	13.0	44.4
it	27.8	25.3	49.9
nl	23.0	21.0	46.0
pt	30.1	27.1	<b>53.6</b>
sv	30.2	24.8	54.4

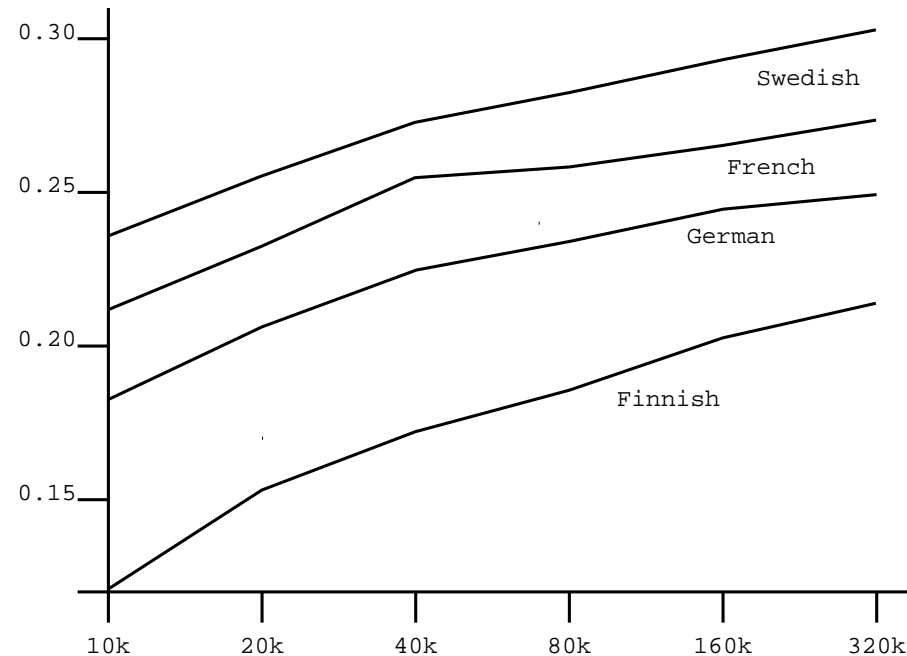
[from Koehn, 2005: Europarl]

---

## Available data

- Available *parallel text*
  - **Europarl**: *30 million words* in 11 languages <http://www.statmt.org/europarl/>
  - **Acquis Communautaire**: *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**
  - *100s of billions, trillions* on the web

## More data, better translations

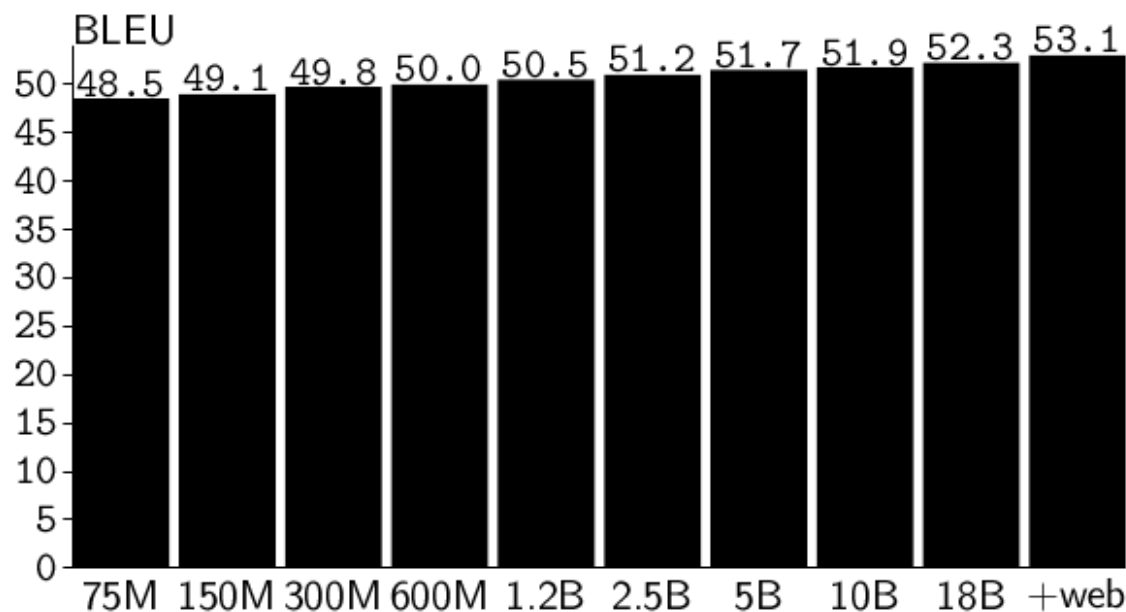


[from Koehn, 2003: Europarl]

- **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)



---

# Output of Chinese-English system

## **In the First Two Months Guangdong's Export of High-Tech Products 3.76 Billion US Dollars**

Xinhua News Agency, Guangzhou, March 16 (Reporter Chen Jizhong) - The latest statistics show that between January and February this year, Guangdong's export of high-tech products 3.76 billion US dollars, with a growth of 34.8% and accounted for the province's total export value of 25.5%. The export of high-tech products bright spots frequently now, the Guangdong provincial foreign trade and economic growth has made important contributions. Last year, Guangdong's export of high-tech products 22.294 billion US dollars, with a growth of 31 percent, an increase higher than the province's total export growth rate of 27.2 percent; exports of high-tech products net increase 5.270 billion us dollars, up for the traditional labor-intensive products as a result of prices to drop from the value of domestic exports decreased.

## **In the Suicide explosion in Jerusalem**

Xinhua News Agency, Jerusalem, March 17 (Reporter bell tsui flower nie Xiaoyang) - A man on the afternoon of 17 in Jerusalem in the northern part of the residents of rammed a bus near ignition of carry bomb, the wrongdoers in red-handed was killed and another nine people were slightly injured and sent to hospital for medical treatment.

---

## Partially excellent translations

### **In the First Two Months Guangdong's Export of High-Tech Products 3.76 Billion US Dollars**

Xinhua News Agency, Guangzhou, March 16 (Reporter Chen Jizhong) - The latest statistics show that between January and February this year, Guangdong's export of high-tech products 3.76 billion US dollars, with a growth of 34.8% and accounted for the province's total export value of 25.5%. The export of high-tech products bright spots frequently now, the Guangdong provincial foreign trade and economic growth has made important contributions. Last year, Guangdong's export of high-tech products 22.294 billion US dollars, with a growth of 31 percent, an increase higher than the province's total export growth rate of 27.2 percent; exports of high-tech products net increase 5.270 billion US dollars, up for the traditional labor-intensive products as a result of prices to drop from the value of domestic exports decreased.

### **In the Suicide explosion in Jerusalem**

Xinhua News Agency, Jerusalem, March 17 (Reporter bell tsui flower nie Xiaoyang) - A man on the afternoon of 17 in Jerusalem in the northern part of the residents of rammed a bus near ignition of carry bomb, the wrongdoers in red-handed was killed and another nine people were slightly injured and sent to hospital for medical treatment.

## Mangled grammar

### **In the First Two Months Guangdong's Export of High-Tech Products 3.76 Billion US Dollars**

Xinhua News Agency, Guangzhou, March 16 (Reporter Chen Jizhong) - The latest statistics show that between January and February this year, Guangdong's **export of high-tech products 3.76 billion US dollars**, with a growth of 34.8% and accounted for the province's total export value of 25.5%. **The export of high-tech products bright spots frequently now**, the Guangdong provincial foreign trade and economic growth has made important contributions. Last year, Guangdong's **export of high-tech products 22.294 billion US dollars**, with a growth of 31 percent, an increase higher than the province's total export growth rate of 27.2 percent; **exports of high-tech products net increase 5.270 billion us dollars**, up for the traditional labor-intensive products **as a result of prices to drop from the value of domestic exports decreased**.

### **In the Suicide explosion in Jerusalem**

Xinhua News Agency, Jerusalem, March 17 (Reporter bell tsui flower nie Xiaoyang) - A man on the afternoon of 17 in Jerusalem in the **northern part of the residents of rammed a bus near ignition of carry bomb**, the **wrongdoers in red-handed was** killed and another nine people were slightly injured and sent to hospital for medical treatment.



# 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 <i>Haus</i>	Count
<i>house</i>	8,000
<i>building</i>	1,600
<i>home</i>	200
<i>household</i>	150
<i>shell</i>	50

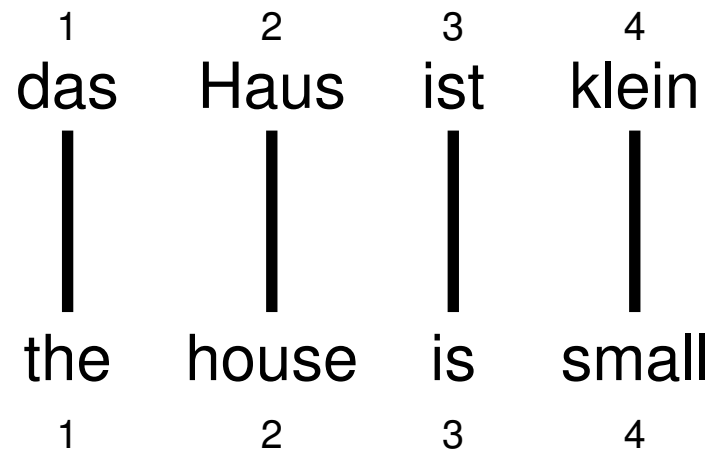
## Estimate translation probabilities

- *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

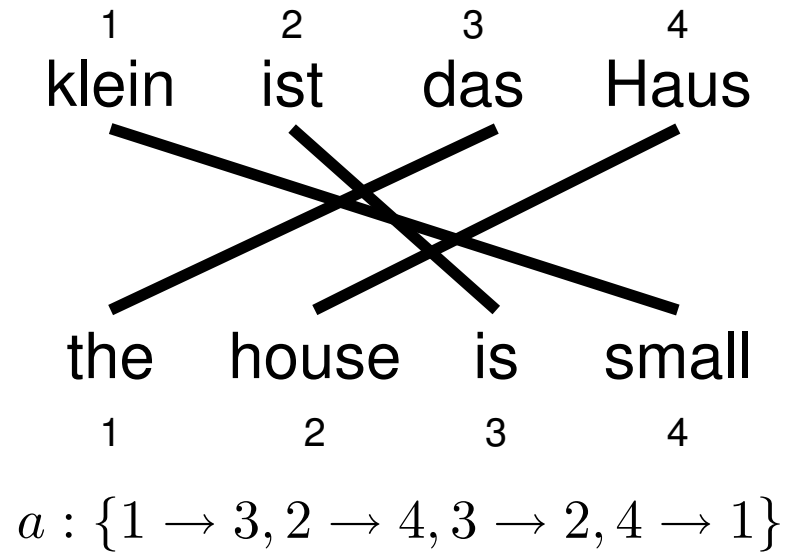
# Alignment function

- 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

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

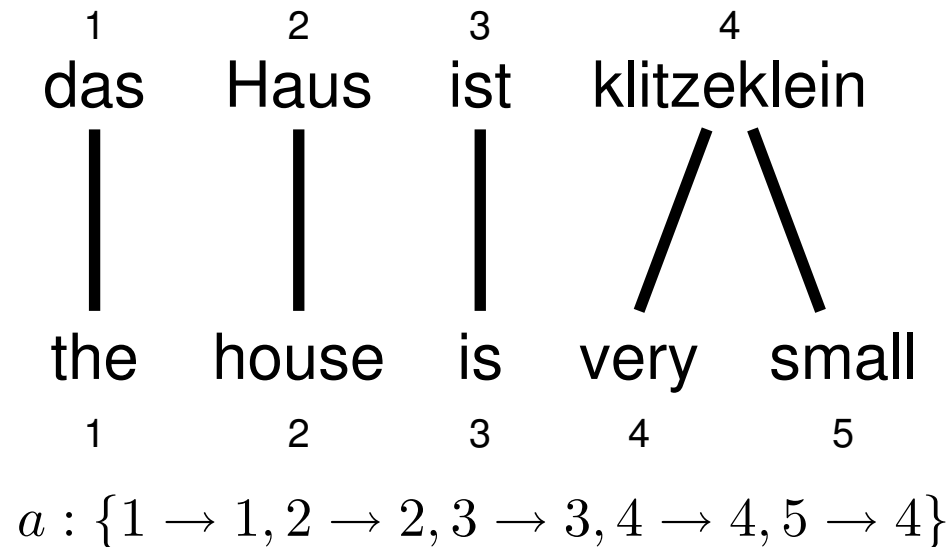
# Reordering

- Words may be **reordered** during translation



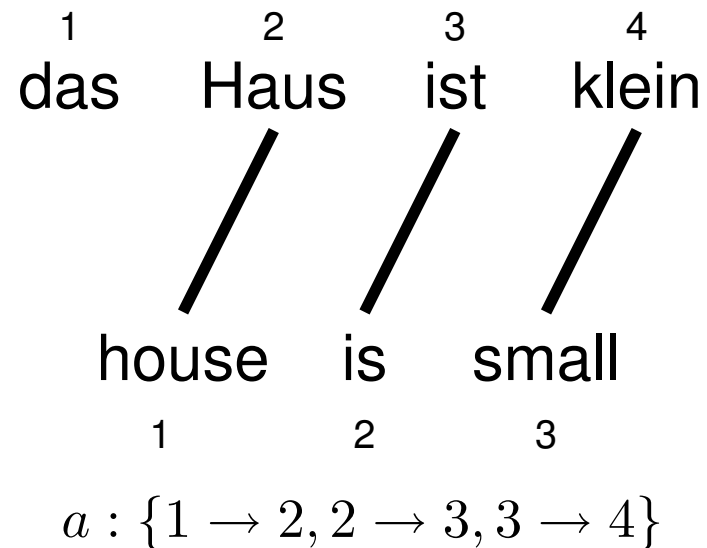
## One-to-many translation

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



## Dropping words

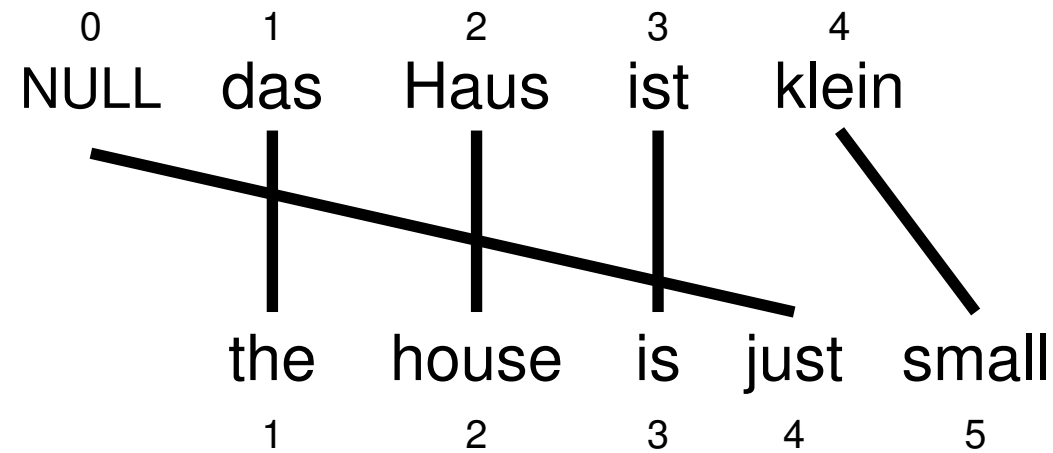
- Words may be **dropped** when translated
  - The German article *das* is dropped





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

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

- parameter  $\epsilon$  is a *normalization constant*



## Example

*das*

$e$	$t(e f)$
<i>the</i>	0.7
<i>that</i>	0.15
<i>which</i>	0.075
<i>who</i>	0.05
<i>this</i>	0.025

*Haus*

$e$	$t(e f)$
<i>house</i>	0.8
<i>building</i>	0.16
<i>home</i>	0.02
<i>household</i>	0.015
<i>shell</i>	0.005

*ist*

$e$	$t(e f)$
<i>is</i>	0.8
<i>'s</i>	0.16
<i>exists</i>	0.02
<i>has</i>	0.015
<i>are</i>	0.005

*klein*

$e$	$t(e f)$
<i>small</i>	0.4
<i>little</i>	0.4
<i>short</i>	0.1
<i>minor</i>	0.06
<i>petty</i>	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

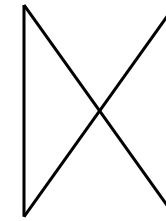
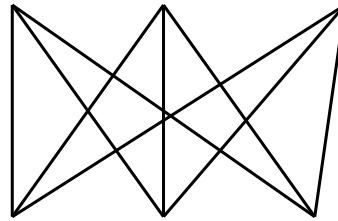
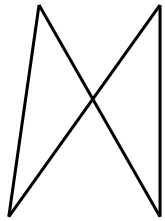
- We would like to *estimate* the lexical translation probabilities  $t(e|f)$  from a parallel corpus
- ... but we do not have the alignments
- **Chicken and egg problem**
  - if we had the *alignments*,
    - we could estimate the *parameters* of our generative model
  - if we had the *parameters*,
    - 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 ...

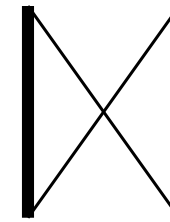
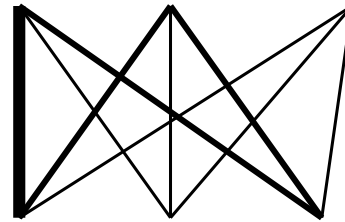
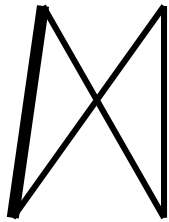


... the house ... the blue house ... the flower ...

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

## EM algorithm

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

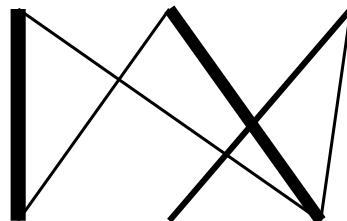
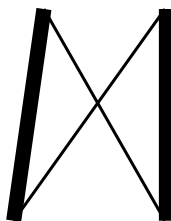


... the house ... the blue house ... the flower ...

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

## EM algorithm

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



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

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



# EM algorithm

... la maison ... la maison bleu ... la fleur ...  
/ | | X | |  
... the house ... the blue house ... the flower ...

- Convergence
- Inherent hidden structure revealed by EM

## EM algorithm

... la maison ... la maison bleu ... la fleur ...  
/ | | | X | |  
... the house ... the blue house ... the flower ...



$p(\text{la}|\text{the}) = 0.453$   
 $p(\text{le}|\text{the}) = 0.334$   
 $p(\text{maison}|\text{house}) = 0.876$   
 $p(\text{bleu}|\text{blue}) = 0.563$   
...

- 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

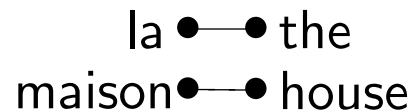
- We need to be able to compute:
  - Expectation-Step: probability of alignments
  - Maximization-Step: count collection

# IBM Model 1 and EM

- Probabilities**

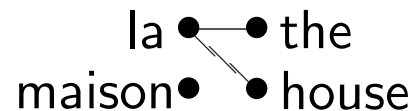
$$\begin{aligned}
 p(\text{the}|\text{la}) &= 0.7 & p(\text{house}|\text{la}) &= 0.05 \\
 p(\text{the}|\text{maison}) &= 0.1 & p(\text{house}|\text{maison}) &= 0.8
 \end{aligned}$$

- Alignments**



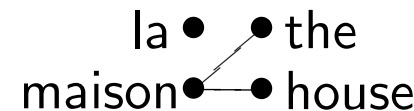
$$p(\mathbf{e}, a|\mathbf{f}) = 0.56$$

$$p(a|\mathbf{e}, \mathbf{f}) = 0.824$$



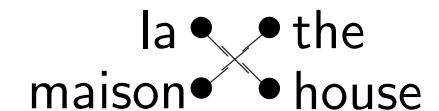
$$p(\mathbf{e}, a|\mathbf{f}) = 0.035$$

$$p(a|\mathbf{e}, \mathbf{f}) = 0.052$$



$$p(\mathbf{e}, a|\mathbf{f}) = 0.08$$

$$p(a|\mathbf{e}, \mathbf{f}) = 0.118$$



$$p(\mathbf{e}, a|\mathbf{f}) = 0.005$$

$$p(a|\mathbf{e}, \mathbf{f}) = 0.007$$

- Counts**

$$\begin{aligned}
 c(\text{the}|\text{la}) &= 0.824 + 0.052 & c(\text{house}|\text{la}) &= 0.052 + 0.007 \\
 c(\text{the}|\text{maison}) &= 0.118 + 0.007 & c(\text{house}|\text{maison}) &= 0.824 + 0.118
 \end{aligned}$$

# IBM Model 1 and EM: Expectation Step

- We need to compute  $p(a|\mathbf{e}, \mathbf{f})$
- Applying the *chain rule*:

$$p(a|\mathbf{e}, \mathbf{f}) = \frac{p(\mathbf{e}, a|\mathbf{f})}{p(\mathbf{e}|\mathbf{f})}$$

- We already have the formula for  $p(\mathbf{e}, \mathbf{a}|\mathbf{f})$  (definition of Model 1)



# IBM Model 1 and EM: Expectation Step

- We need to compute  $p(\mathbf{e}|\mathbf{f})$

$$\begin{aligned} p(\mathbf{e}|\mathbf{f}) &= \sum_a p(\mathbf{e}, a|\mathbf{f}) \\ &= \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)=0}^{l_f} p(\mathbf{e}, a|\mathbf{f}) \\ &= \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)=0}^{l_f} \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|f_{a(j)}) \end{aligned}$$



## IBM Model 1 and EM: Expectation Step

$$\begin{aligned}
 p(\mathbf{e}|\mathbf{f}) &= \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)=0}^{l_f} \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j | f_{a(j)}) \\
 &= \frac{\epsilon}{(l_f + 1)^{l_e}} \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)=0}^{l_f} \prod_{j=1}^{l_e} t(e_j | f_{a(j)}) \\
 &= \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j | f_i)
 \end{aligned}$$

- Note the trick in the last line
  - removes the need for an *exponential* number of products
  - this makes IBM Model 1 estimation **tractable**





# IBM Model 1 and EM: Expectation Step

- Combine what we have:

$$\begin{aligned} p(\mathbf{a}|\mathbf{e}, \mathbf{f}) &= p(\mathbf{e}, \mathbf{a}|\mathbf{f})/p(\mathbf{e}|\mathbf{f}) \\ &= \frac{\frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|f_{a(j)})}{\frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j|f_i)} \\ &= \prod_{j=1}^{l_e} \frac{t(e_j|f_{a(j)})}{\sum_{i=0}^{l_f} t(e_j|f_i)} \end{aligned}$$

# IBM Model 1 and EM: Maximization Step

- Now we have to *collect counts*
- Evidence from a sentence pair  $\mathbf{e}, \mathbf{f}$  that word  $e$  is a translation of word  $f$ :

$$c(e|f; \mathbf{e}, \mathbf{f}) = \sum_a p(a|\mathbf{e}, \mathbf{f}) \sum_{j=1}^{l_e} \delta(e, e_j) \delta(f, f_{a(j)})$$

- With the same simplification as before:

$$c(e|f; \mathbf{e}, \mathbf{f}) = \frac{t(e|f)}{\sum_{j=1}^{l_e} t(e|f_{a(j)})} \sum_{j=1}^{l_e} \delta(e, e_j) \sum_{i=0}^{l_f} \delta(f, f_i)$$

# IBM Model 1 and EM: Maximization Step

- After collecting these counts over a corpus, we can estimate the model:

$$t(e|f; \mathbf{e}, \mathbf{f}) = \frac{\sum_{(\mathbf{e}, \mathbf{f})} c(e|f; \mathbf{e}, \mathbf{f})}{\sum_f \sum_{(\mathbf{e}, \mathbf{f})} c(e|f; \mathbf{e}, \mathbf{f})}$$



# IBM Model 1 and EM: Pseudocode

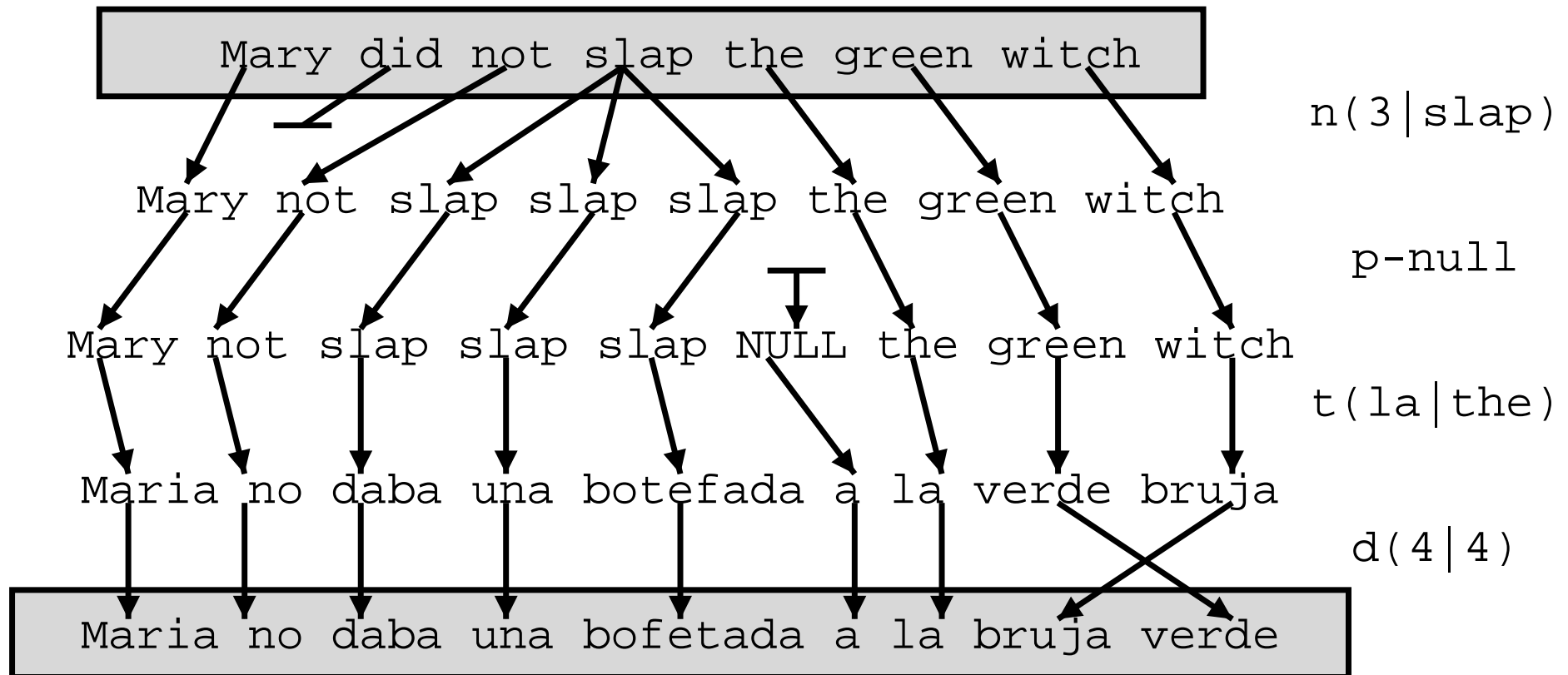
```
initialize  $t(e|f)$  uniformly
do
  set count( $e|f$ ) to 0 for all  $e, f$ 
  set total( $f$ ) to 0 for all  $f$ 
  for all sentence pairs ( $e_s, f_s$ )
    for all words  $e$  in  $e_s$ 
      total_s = 0
      for all words  $f$  in  $f_s$ 
        total_s +=  $t(e|f)$ 
      for all words  $e$  in  $e_s$ 
        for all words  $f$  in  $f_s$ 
          count( $e|f$ ) +=  $t(e|f) / \text{total}_s$ 
          total( $f$ ) +=  $t(e|f) / \text{total}_s$ 
    for all  $f$  in domain( total(.) )
      for all  $e$  in domain( count(.|f) )
         $t(e|f) = \text{count}(e|f) / \text{total}(f)$ 
until convergence
```

## Higher IBM Models

IBM Model 1	lexical translation
IBM Model 2	adds absolute <b>reordering model</b>
IBM Model 3	adds <b>fertility model</b>
IBM Model 4	relative reordering model
IBM Model 5	fixes <b>deficiency</b>

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

# IBM Model 4





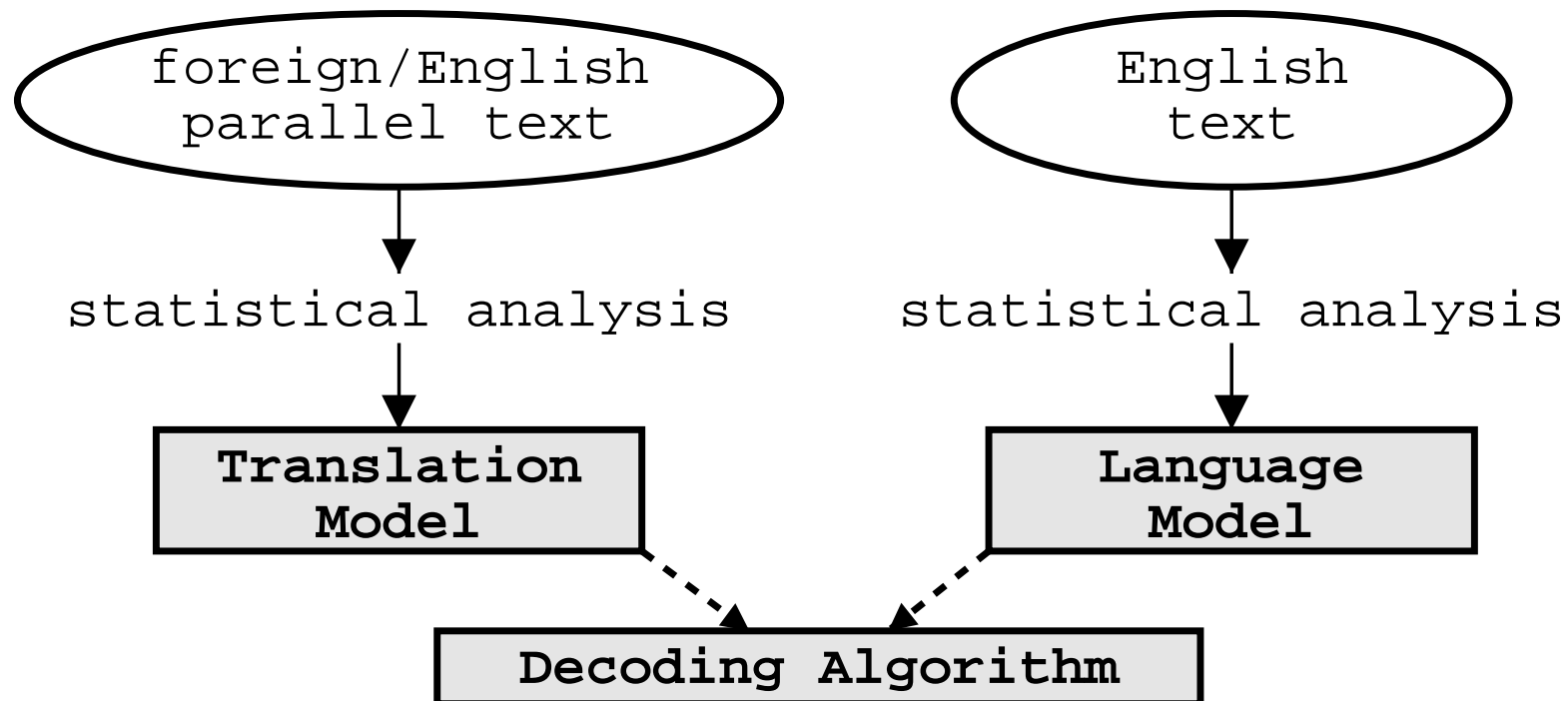
---

## Late morning session

- Decoding
- Phrase-based models

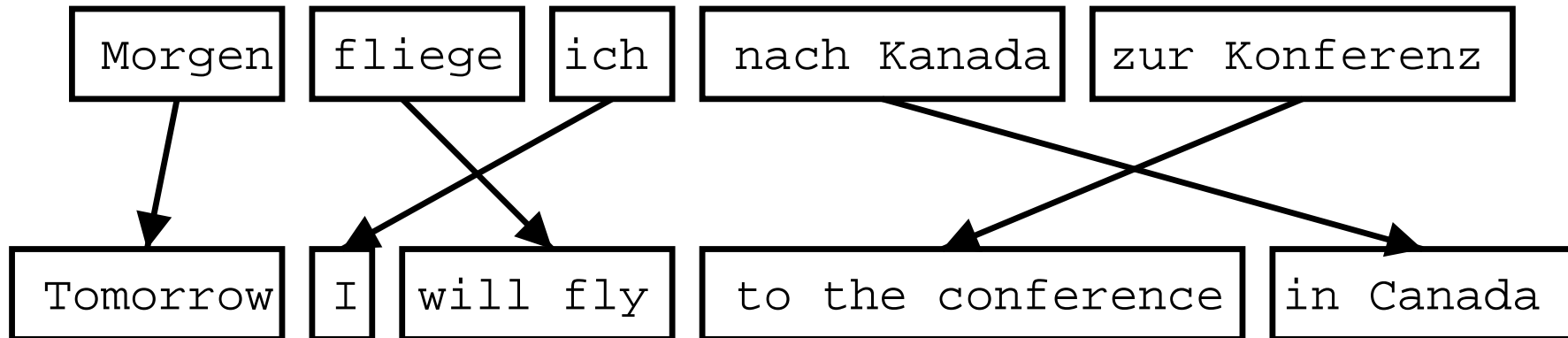
# Statistical Machine Translation

- 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” :

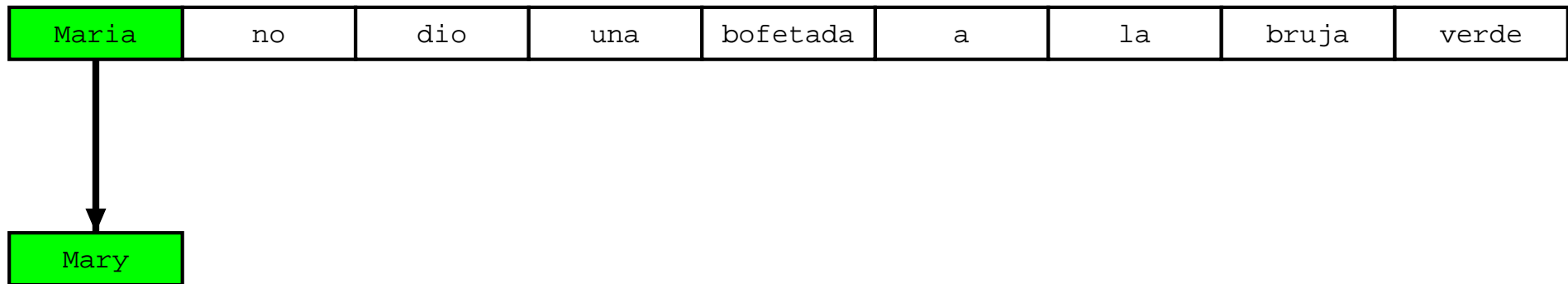
English	$\phi(e f)$	English	$\phi(e f)$
the proposal	0.6227	the suggestions	0.0114
's proposal	0.1068	the proposed	0.0114
a proposal	0.0341	the motion	0.0091
the idea	0.0250	the idea of	0.0091
this proposal	0.0227	the proposal ,	0.0068
proposal	0.0205	its proposal	0.0068
of the proposal	0.0159	it	0.0068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

Maria	no	dio	una	bofetada	a	la	bruja	verde
-------	----	-----	-----	----------	---	----	-------	-------

↓

Mary	did not
------	---------

- *One to many* translation

# Decoding Process



- Many to one translation

# Decoding Process



- *Many to one* translation



# Decoding Process

Maria	no	dio una bofetada	a la	bruja	verde
-------	----	------------------	------	-------	-------

Mary	did not	slap	the	green
------	---------	------	-----	-------



- *Reordering*

# Decoding Process



- Translation *finished*

## Translation Options

Maria	no	dio	una	bofetada	a	la	bruja	verde
<u>Mary</u>	<u>not</u>	<u>give</u>	<u>a</u>	<u>slap</u>	<u>to</u>	<u>the</u>	<u>witch</u>	<u>green</u>
	<u>did not</u>		<u>a slap</u>		<u>by</u>		<u>green witch</u>	
	<u>no</u>		<u>slap</u>		<u>to the</u>			
	<u>did not give</u>				<u>to</u>			
					<u>the</u>			
			<u>slap</u>			<u>the witch</u>		

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

# Hypothesis Expansion

Maria	no	dio	una	bofetada	a	la	bruja	verde
<u>Mary</u>	<u>not</u>	<u>give</u>	<u>a</u>	<u>slap</u>	<u>to</u>	<u>the</u>	<u>witch</u>	<u>green</u>
	<u>did not</u>		<u>a slap</u>		<u>by</u>		<u>green witch</u>	
	<u>no</u>		<u>slap</u>		<u>to the</u>			
	<u>did not give</u>				<u>to</u>			
					<u>the</u>			
			<u>slap</u>			<u>the witch</u>		

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

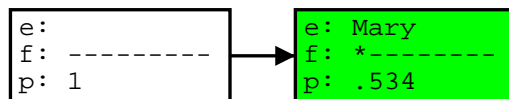
- Start with **empty hypothesis**
  - e: no English words
  - f: no foreign words covered
  - p: probability 1

# Hypothesis Expansion

Maria	no	dio	una	bofetada	a	la	bruja	verde
-------	----	-----	-----	----------	---	----	-------	-------

Mary	<u>not</u>	<u>give</u>	<u>a</u>	<u>slap</u>	<u>to</u>	<u>the</u>	<u>witch</u>	<u>green</u>
	<u>did not</u>		<u>a slap</u>		<u>by</u>		<u>green witch</u>	
	<u>no</u>		<u>slap</u>		<u>to the</u>			
	<u>did not give</u>				<u>to</u>			
					<u>the</u>			
			<u>slap</u>				<u>the witch</u>	



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

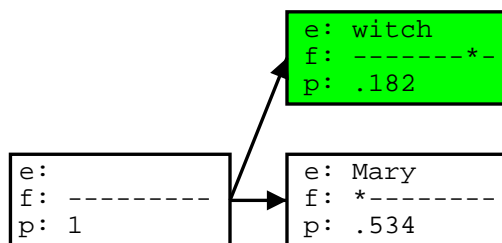
---

# A Quick Word on Probabilities

- Not going into detail here, but...
- *Translation Model*
  - phrase translation probability  $p(\text{Mary}|\text{Maria})$
  - reordering costs
  - phrase/word count costs
  - ...
- *Language Model*
  - uses trigrams:
  - $p(\text{Mary did not}) =$   
 $p(\text{Mary}|\text{START}) \times p(\text{did}|\text{Mary,START}) \times p(\text{not}|\text{Mary did})$

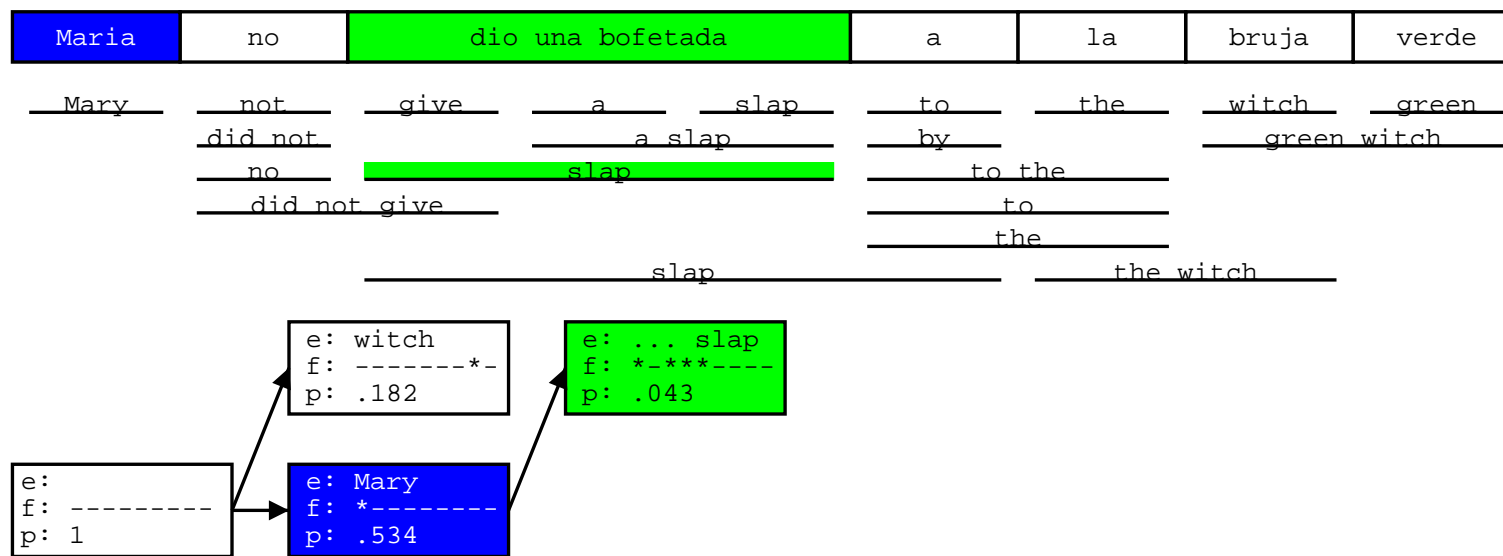
# Hypothesis Expansion

Maria	no	dio	una	bofetada	a	la	bruja	verde
<u>Mary</u>	<u>not</u>	<u>give</u>	<u>a</u>	<u>slap</u>	<u>to</u>	<u>the</u>	<u>witch</u>	<u>green</u>
	<u>did not</u>		<u>a slap</u>		<u>by</u>		<u>green witch</u>	
	<u>no</u>		<u>slap</u>		<u>to the</u>			
	<u>did not give</u>				<u>to</u>			
					<u>the</u>			
			<u>slap</u>			<u>the witch</u>		



- Add another *hypothesis*

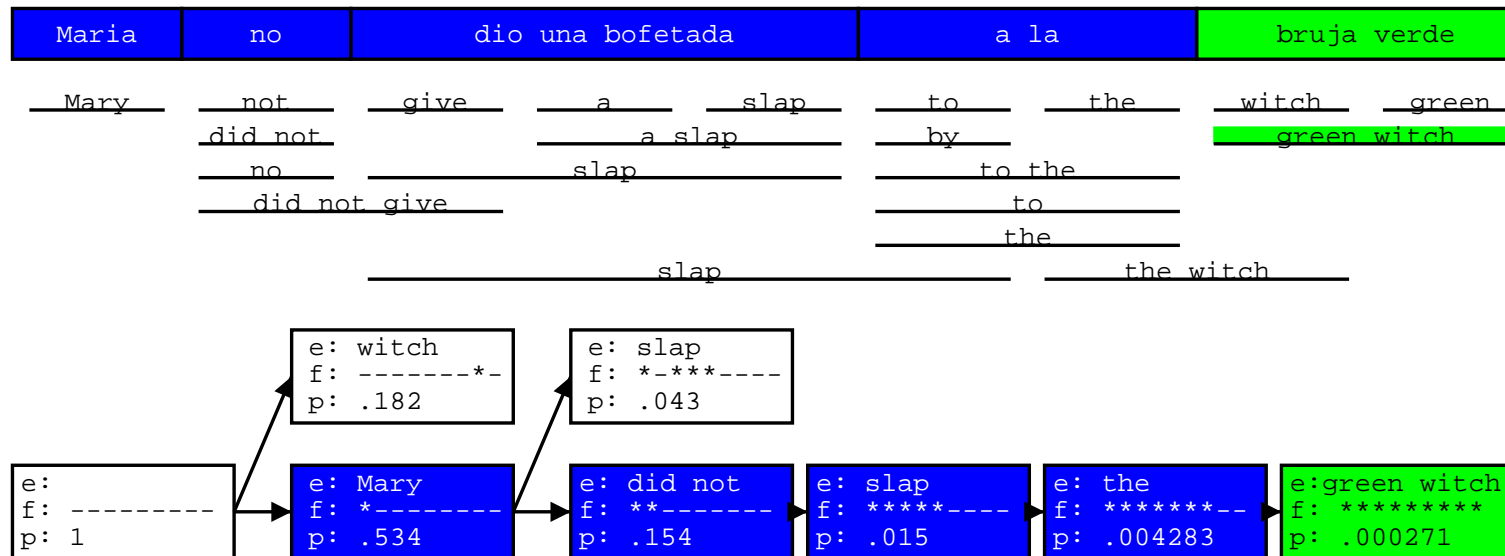
# Hypothesis Expansion



- Further *hypothesis expansion*

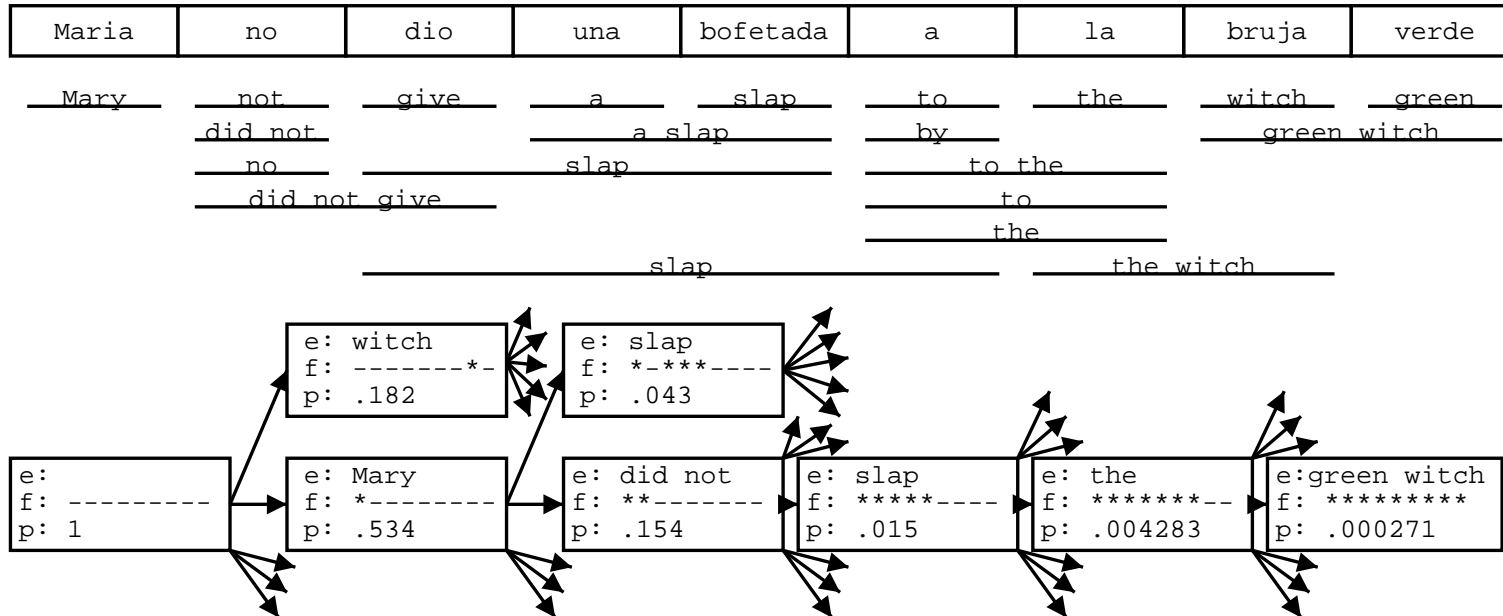


# Hypothesis Expansion



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

# Hypothesis Expansion



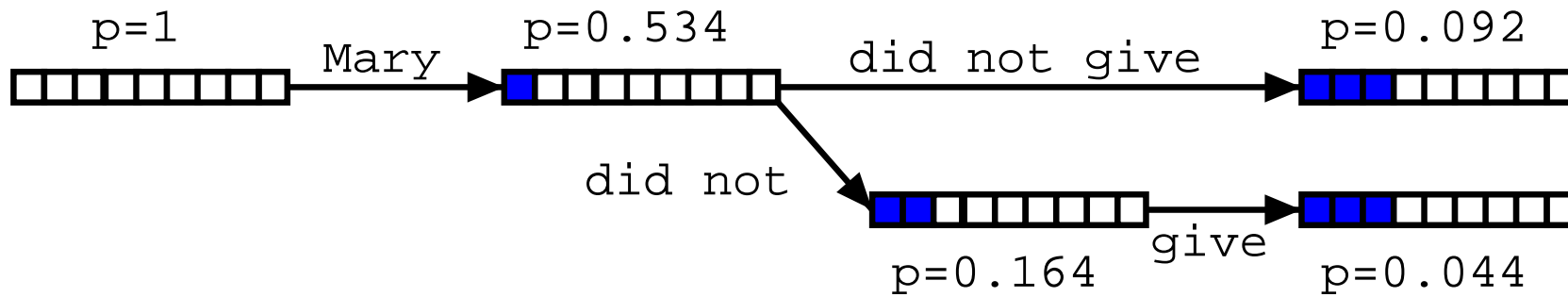
- Adding more hypothesis

⇒ *Explosion* of search space

# Explosion of Search Space

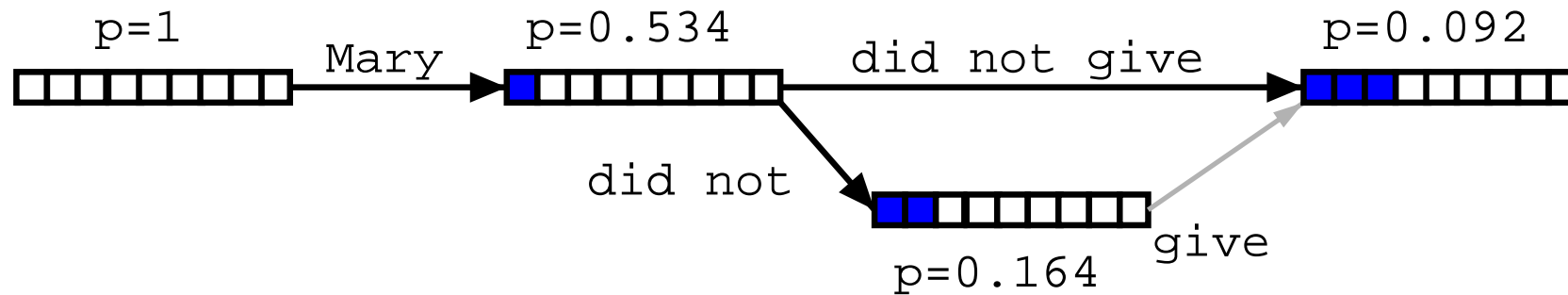
- Number of hypotheses is *exponential* with respect to sentence length
- ⇒ Decoding is NP-complete [Knight, 1999]
- ⇒ Need to *reduce search space*
- risk free: hypothesis **recombination**
  - risky: **histogram/threshold pruning**

# Hypothesis Recombination



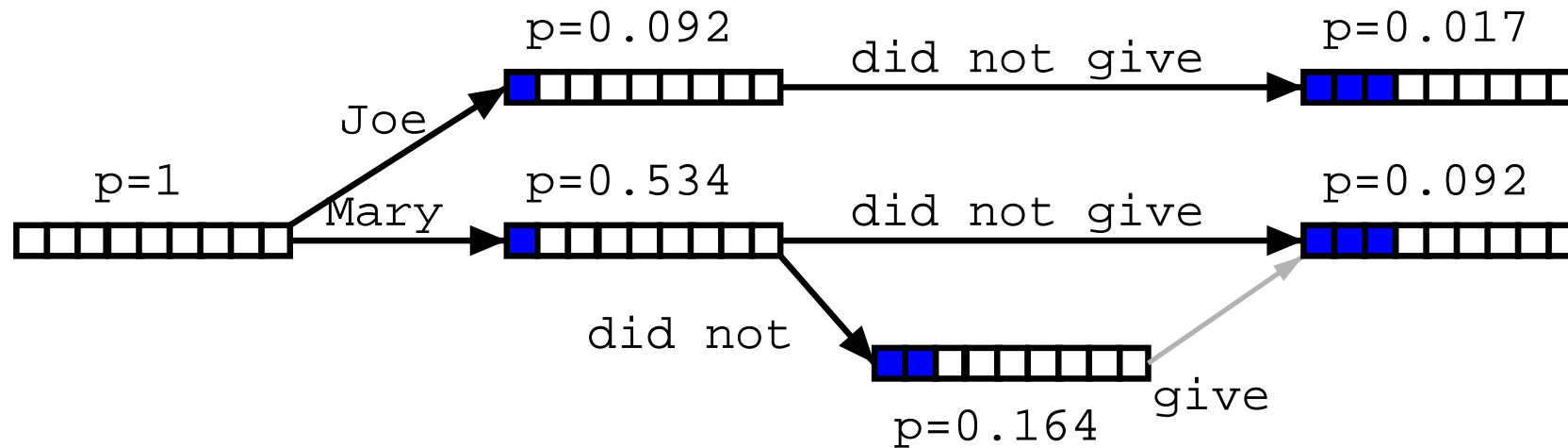
- Different paths to the *same* partial translation

# Hypothesis Recombination



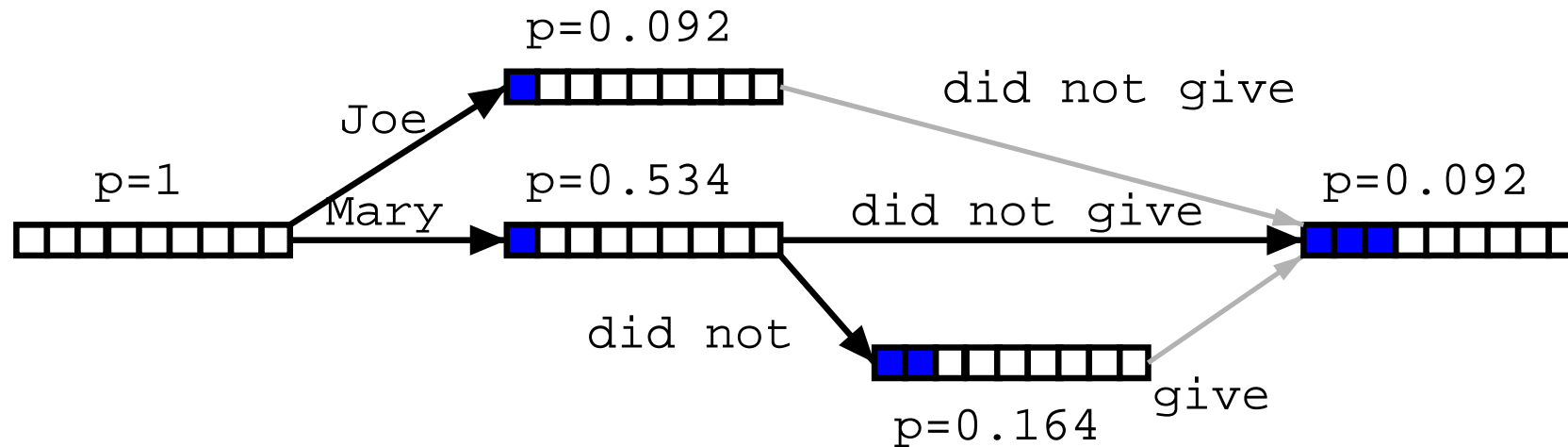
- Different paths to the same partial translation
- ⇒ *Combine paths*
- *drop weaker* path
  - keep pointer from weaker path (for lattice generation)

## 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)

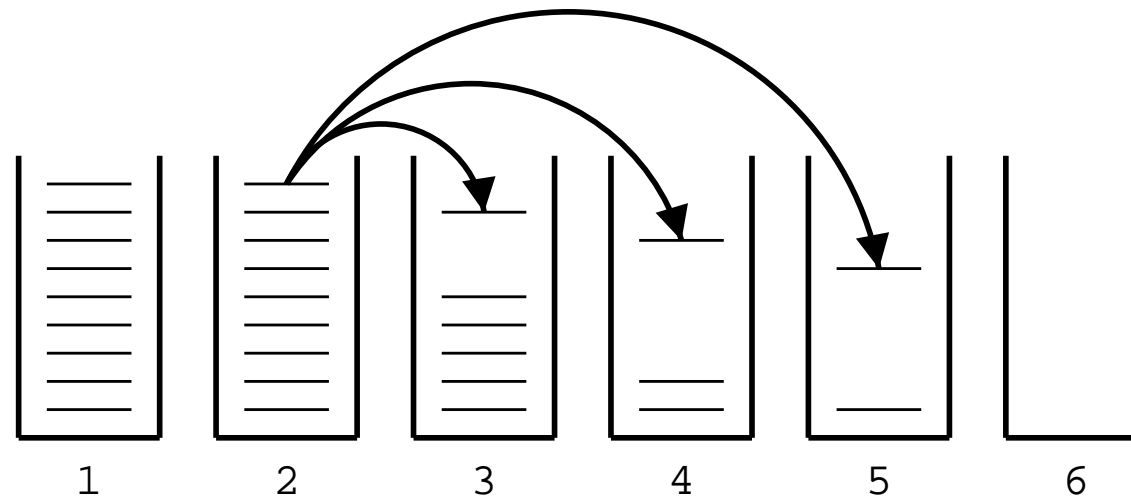
⇒ *Combine paths*

# 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  $\alpha$  times the cost of best hypothesis in stack (e.g.,  $\alpha = 0.001$ )



# 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

Maria no            dio una bofetada            a la            bruja verde

e: Mary did not  
 f: \*\*-----  
 p: 0.154

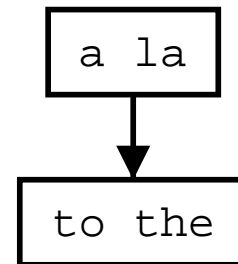
better  
 partial  
 translation

e: the  
 f: -----\*\*--  
 p: 0.354

covers  
 easier part  
 --> lower cost

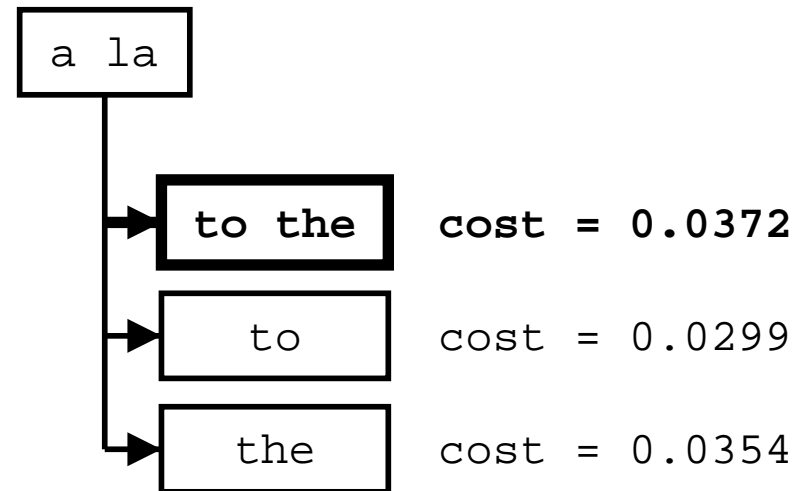
- Hypothesis that covers *easy part* of sentence is preferred
- ⇒ Need to consider **future cost** of uncovered parts

# Future Cost Estimation



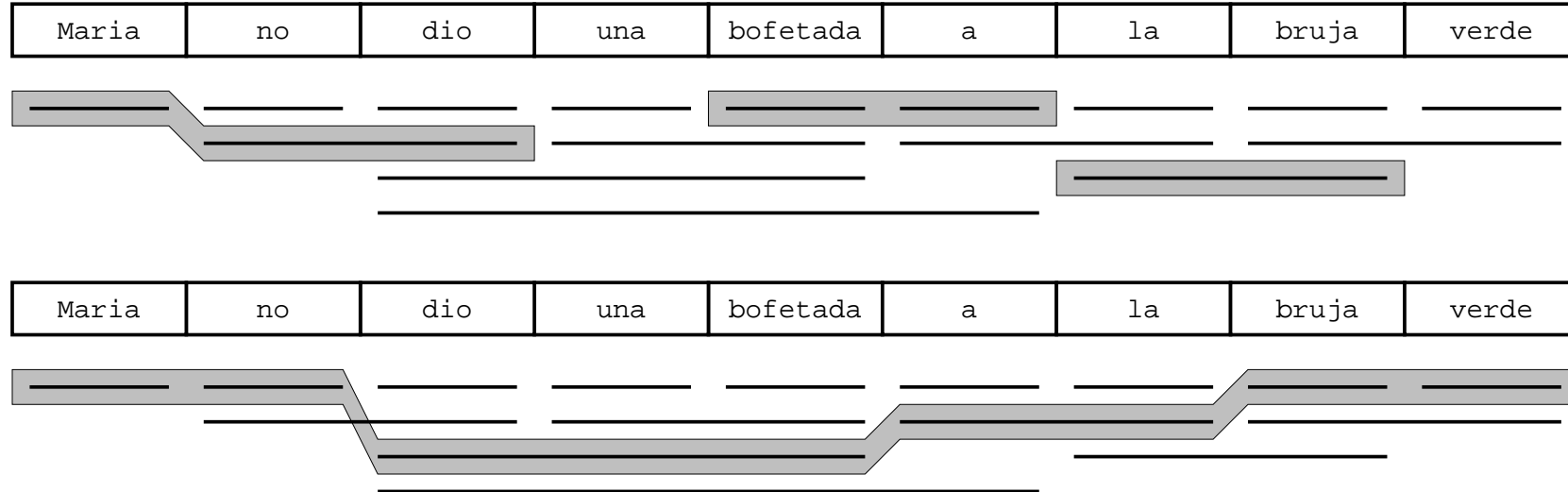
- *Estimate cost* to translate remaining part of input
  - Step 1: estimate future cost for each *translation option*
    - look up translation model cost
    - estimate language model cost (no prior context)
    - ignore reordering model cost
- $LM * TM = p(\text{to}) * p(\text{the}|\text{to}) * p(\text{to the}|\text{a la})$

## Future Cost Estimation: Step 2



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

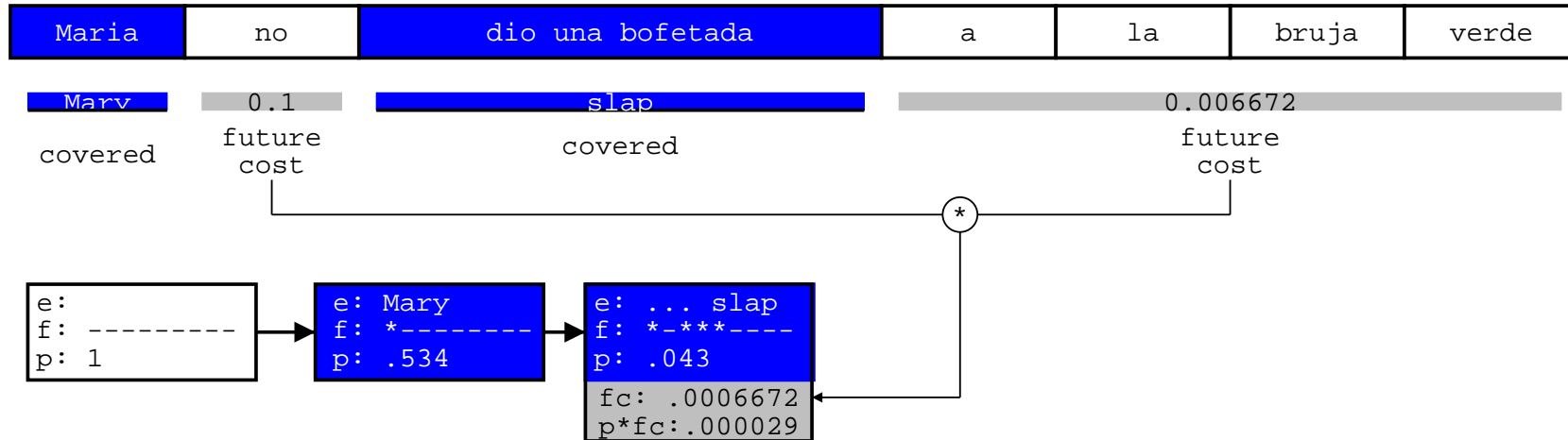
## Future Cost Estimation: Step 3



- 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

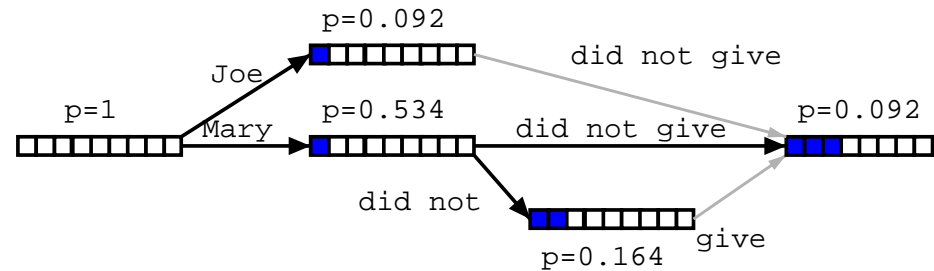
- Pruning might drop hypothesis that lead to the best path (**search error**)
- **A\* search**: safe pruning
  - future cost estimates have to be accurate or underestimates
  - **lower bound** for probability is established early by **depth first search**: compute cost for one complete translation
  - 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

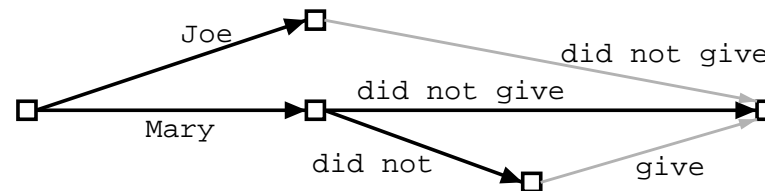
- Reordering may be **limited**
  - **Monotone** Translation: No reordering at all
  - Only phrase movements of at most  $n$  words
- Reordering limits *speed* up search (polynomial instead of exponential)
- Current reordering models are weak, so limits *improve* translation quality



# 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
```

# Moses: Open Source Toolkit



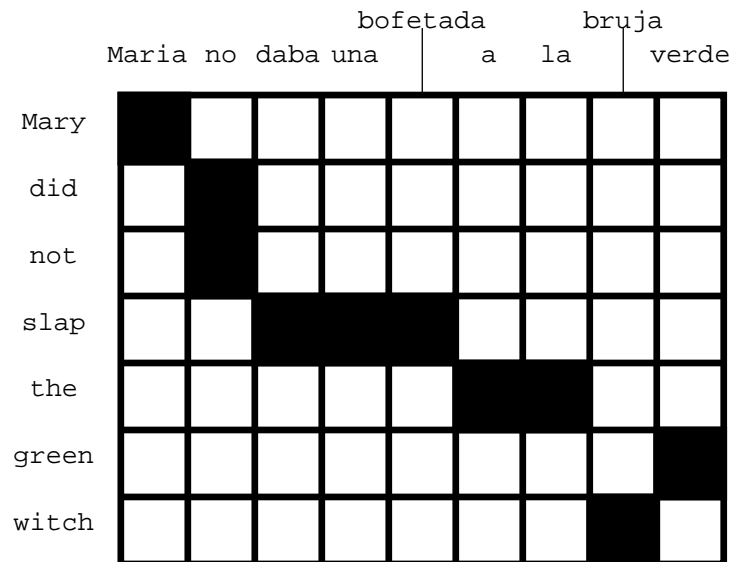
- **Open source** statistical machine translation system (developed from scratch 2006)
  - state-of-the-art *phrase-based* approach
  - novel methods: *factored translation models*, *confusion network decoding*
  - support for *very large models* through *memory-efficient* data structures
- Documentation,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



# Phrase-based models

# Word alignment

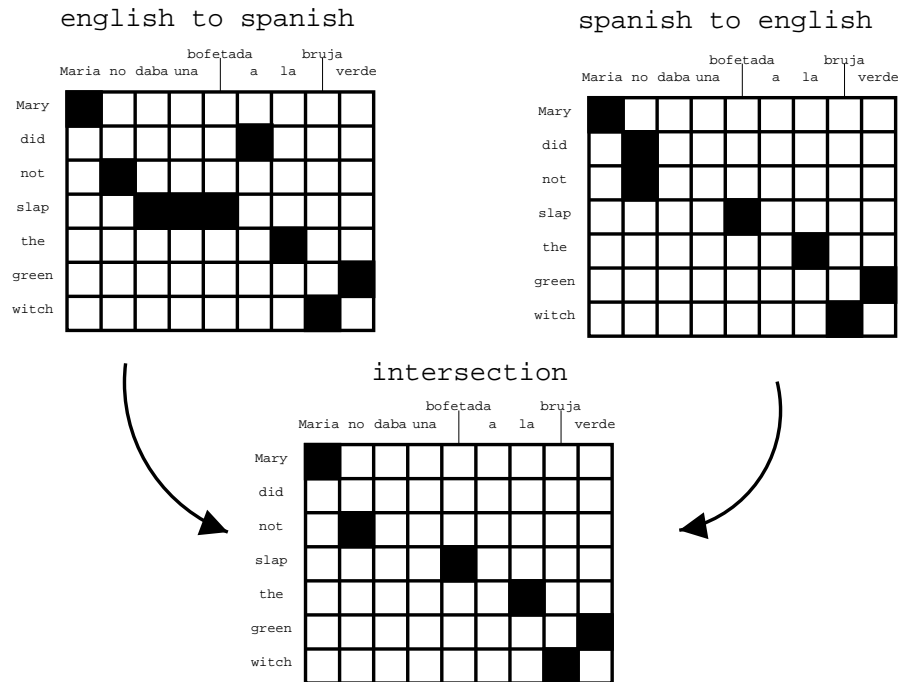
- Notion of **word alignment** valuable
- Shared task at NAACL 2003 and ACL 2005 workshops



## Word alignment with IBM models

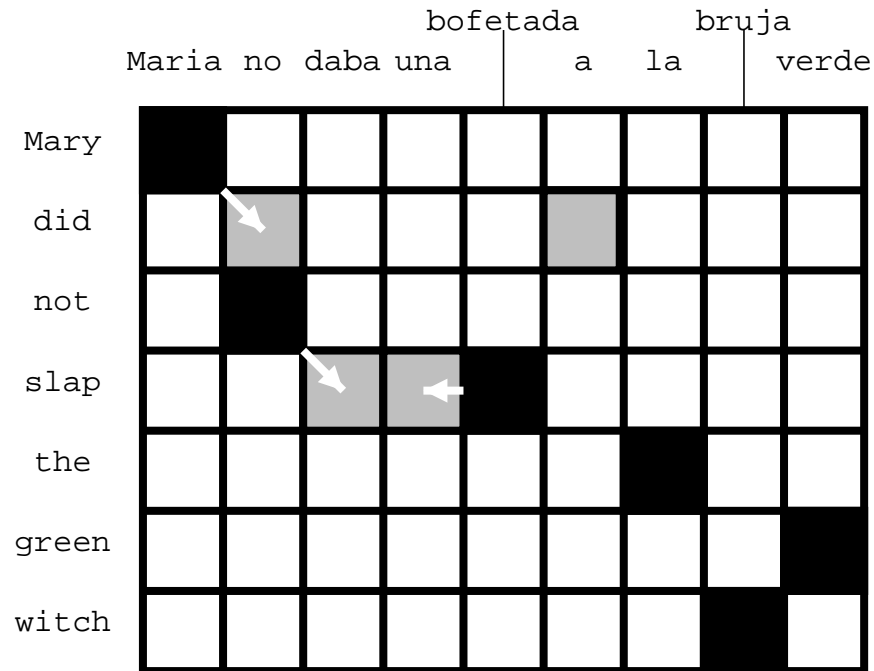
- IBM Models create a *many-to-one* mapping
  - words are aligned using an **alignment function**
  - a function may return the same value for different input (one-to-many mapping)
  - a function can not return multiple values for one input (*no many-to-one* mapping)
- But we need *many-to-many* mappings

# Symmetrizing word alignments



- *Intersection* of GIZA++ bidirectional alignments

# Symmetrizing word alignments



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



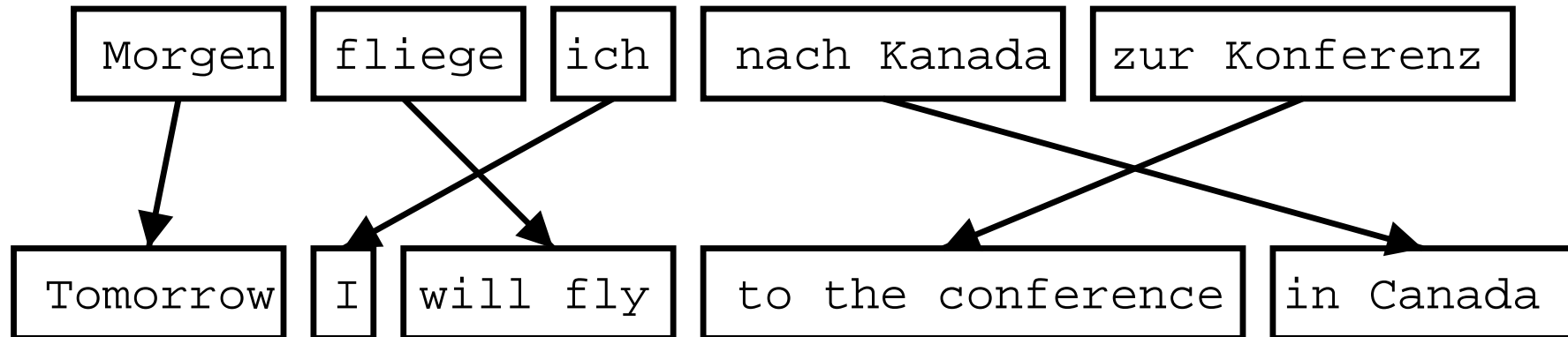
## Growing heuristic

```
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  $\bar{f}_1^I = \bar{f}_1, \dots, \bar{f}_I$

- 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	$\phi(e f)$	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	...	...

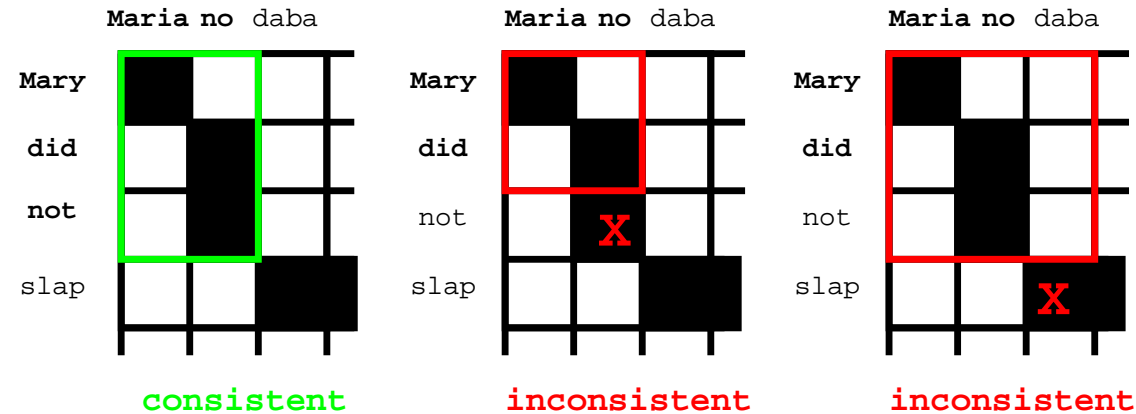
# How to learn the phrase translation table?

- Start with the *word alignment*:

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

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

## Consistent with word alignment

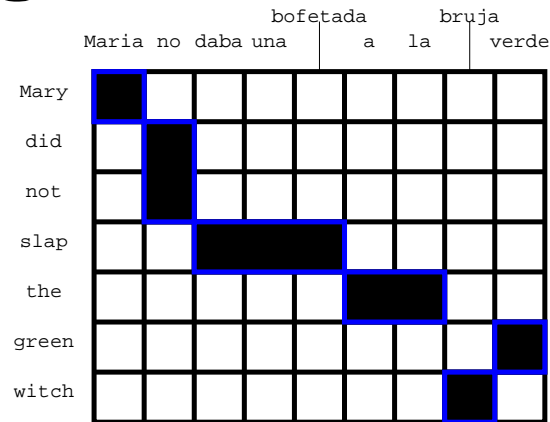


- **Consistent with the word alignment** :=

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

$$(\bar{e}, \bar{f}) \in BP \Leftrightarrow \begin{aligned} &\forall e_i \in \bar{e} : (e_i, f_j) \in A \rightarrow f_j \in \bar{f} \\ \text{AND} &\quad \forall f_j \in \bar{f} : (e_i, f_j) \in A \rightarrow e_i \in \bar{e} \end{aligned}$$

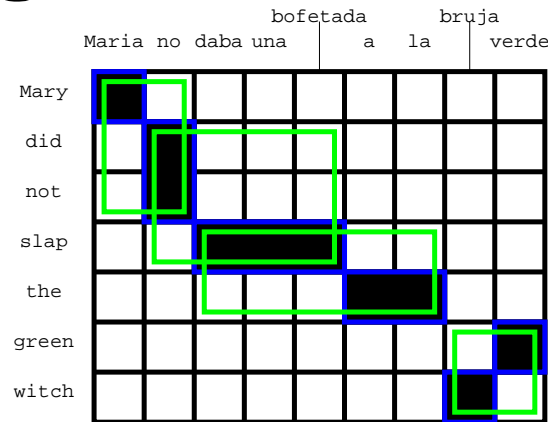
# Word alignment induced phrases



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

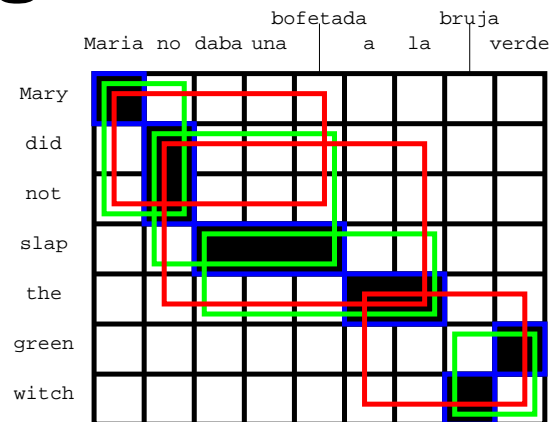


# Word alignment induced phrases



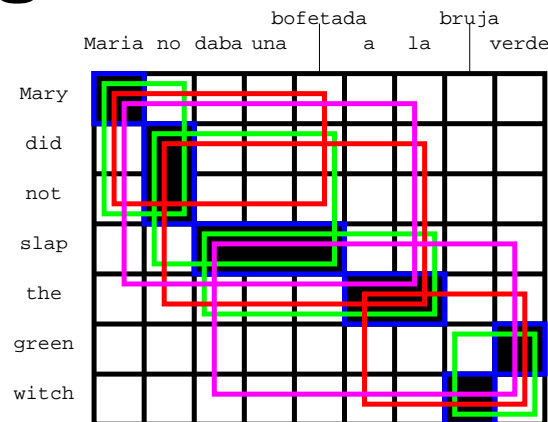
- (Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green),
- (Maria no, Mary did not), (no daba una bofetada, did not slap), (daba una bofetada a la, slap the),
- (bruja verde, green witch)

# Word alignment induced phrases



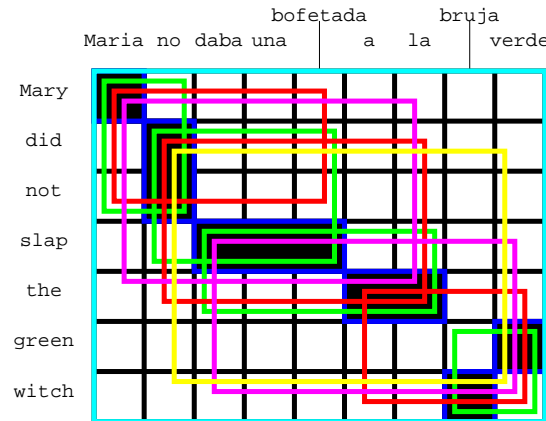
(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green),  
 (Maria no, Mary did not), (no daba una bofetada, did not slap), (daba una bofetada a la, slap the),  
 (bruja verde, green witch), (Maria no daba una bofetada, Mary did not slap),  
 (no daba una 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**  $\phi(\bar{f}|\bar{e})$  over the collected phrase pairs

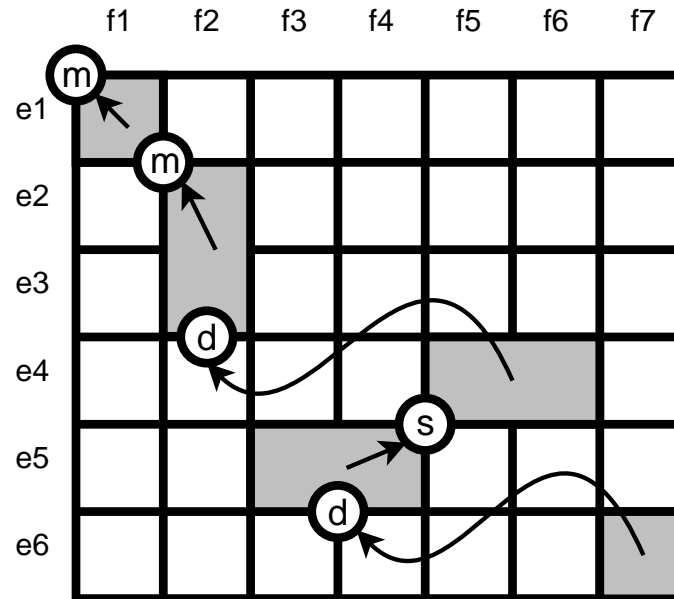
⇒ Possible *choices*

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

# Reordering

- *Monotone* translation
  - do not allow any reordering
  - worse translations
- *Limiting* reordering (to movement over max. number of words) helps
- *Distance-based* reordering cost
  - moving a foreign phrase over  $n$  words: cost  $\omega^n$
- *Lexicalized* reordering model

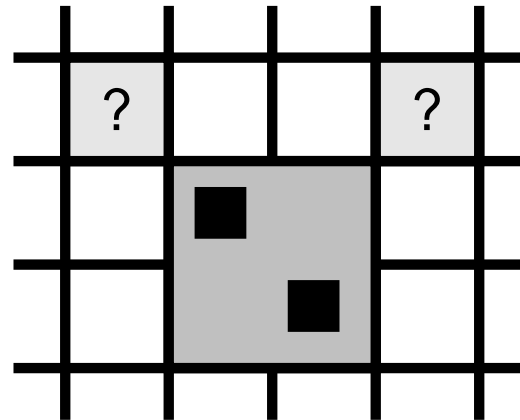
# Lexicalized reordering models



[from Koehn et al., 2005, IWSLT]

- Three **orientation** types: **monotone**, **swap**, **discontinuous**
- Probability  $p(\text{swap}|e, f)$  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]



## 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  $\lambda_i$

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

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

---

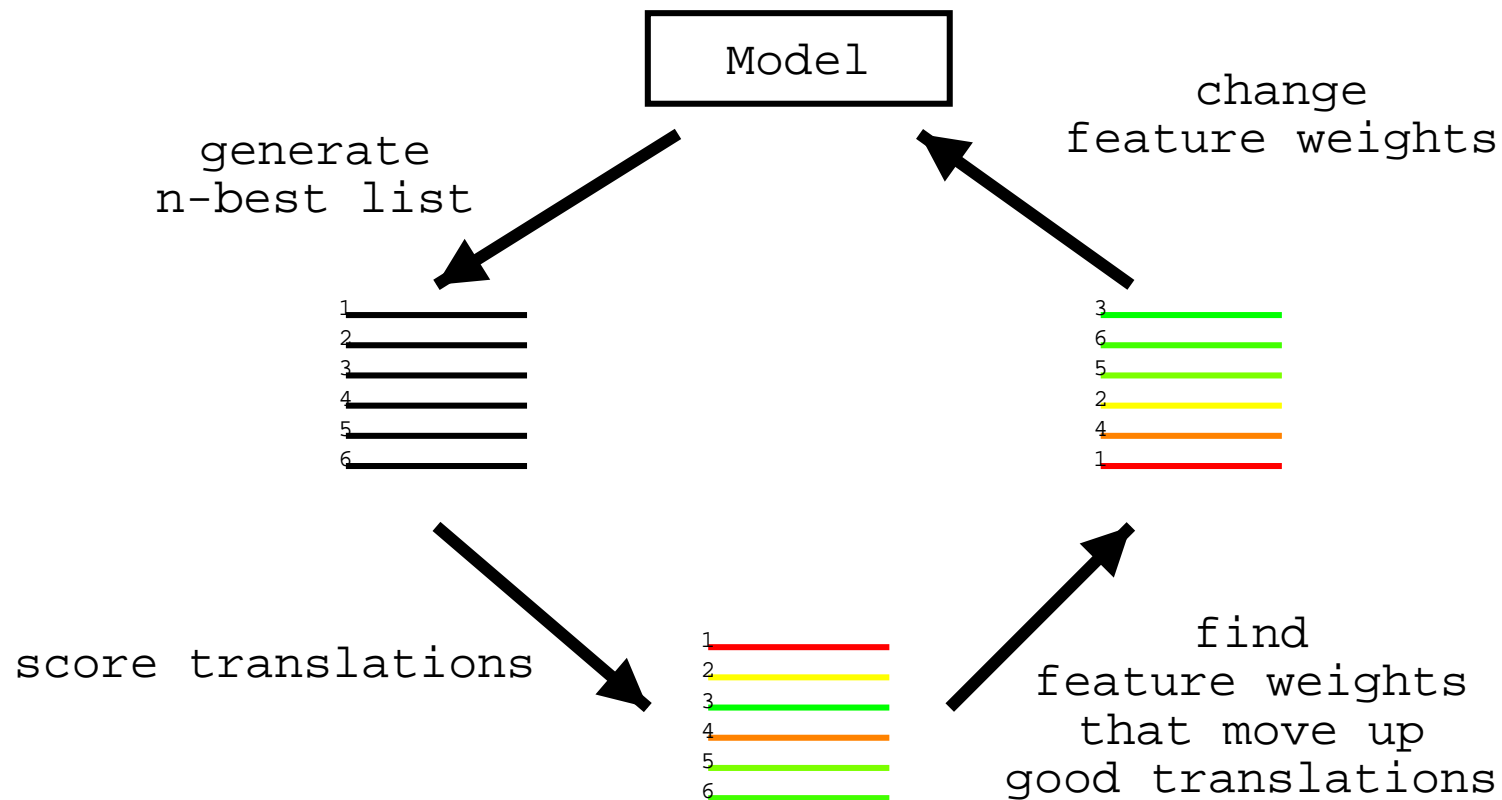
## Knowledge sources

- 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  $\lambda_i$
- 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)

# Learn feature weights



## 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*

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
<b>11 Mary did not slap the green witch .</b>	<b>-17.4</b>	<b>-5.3</b>	<b>-8</b>	<b>0</b>
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
rank translation	feature vector			

## 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