# CS378: Natural Language Processing

# Lecture 23: Machine Translation

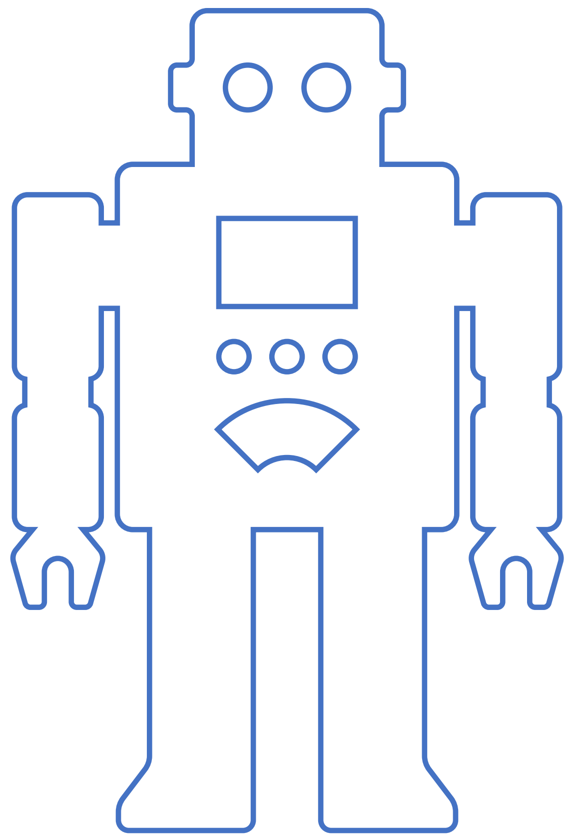
TEXAS
The University of Texas at Austin

Eunsol Choi

slides adapted from Greg Durrett / Yoav Artzi

# Conditional Text Generation: Translation

Yesterday was Sunday.



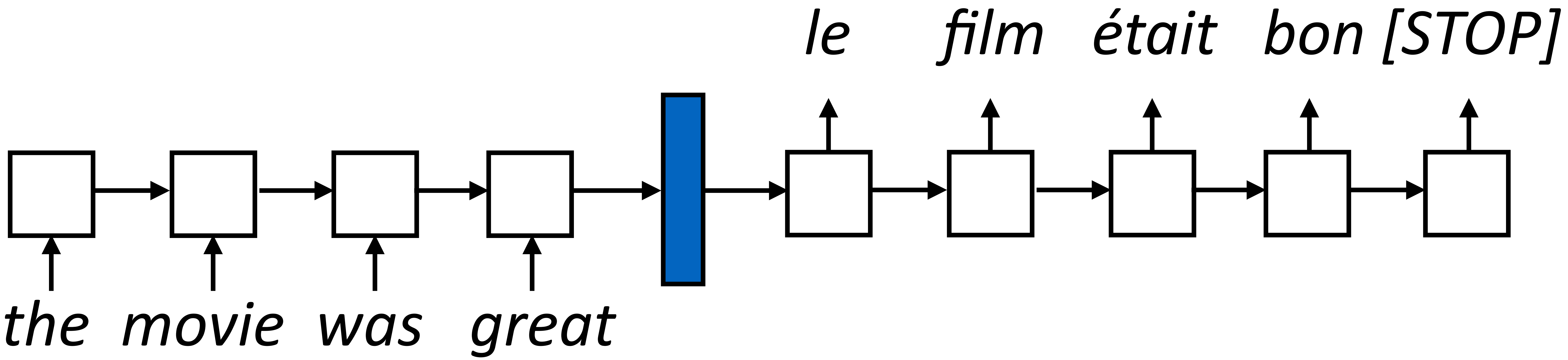
Ontem foi domingo.

Input: Text in {English,…}

Output: Text in {Portuguese,…}

# Machine Translation in 2022

- Variants of encoder-decoder model, with attention mechanism
- We will talk more about it next Tuesday.




- Today: Pre-neural machine translation model

# Machine Translation

**"Il est impossible aux journalistes de rentrer dans les régions tibétaines"**

Bruno Philip, correspondant du "Monde" en Chine, estime que les journalistes de l'AFP qui ont été expulsés de la province tibétaine du Qinghai "n'étaient pas dans l'illégalité".

**Les faits** Le dalaï-lama dénonce l'"enfer" imposé au Tibet depuis sa fuite, en 1959
**Vidéo** Anniversaire de la rébellion

**"It is impossible for journalists to enter Tibetan areas"**

Philip Bruno, correspondent for "World" in China, said that journalists of the AFP who have been deported from the Tibetan province of Qinghai "were not illegal."

**Facts** The Dalai Lama denounces the "hell" imposed since he fled Tibet in 1959
**Video** Anniversary of the Tibetan rebellion: China on guard

- Translate text from one language to another
- Challenges:
  - How to make efficient?
  - Fluency vs. Fidelity

# Machine Translation

- Goal:
  - Conserve the meaning *(and style)* of the original sentence.
  - Sometimes concepts and ambiguities does not transfer easily.

A doctor visited a friend last night. → Un médecin a rendu visite à un ami hier soir.

Une médecin a rendu visite à une amie la nuit dernière.

# Ideal Scenario

- *I have a friend =>* `∃x friend(x,self)` *=> J'ai un ami (friend is male)*

  *J'ai une amie (friend is female)*

  - May need information you didn't think about in your representation
  - Hard for semantic representations to cover everything

# MT in Practice

- Bitext: What can you learn from this?

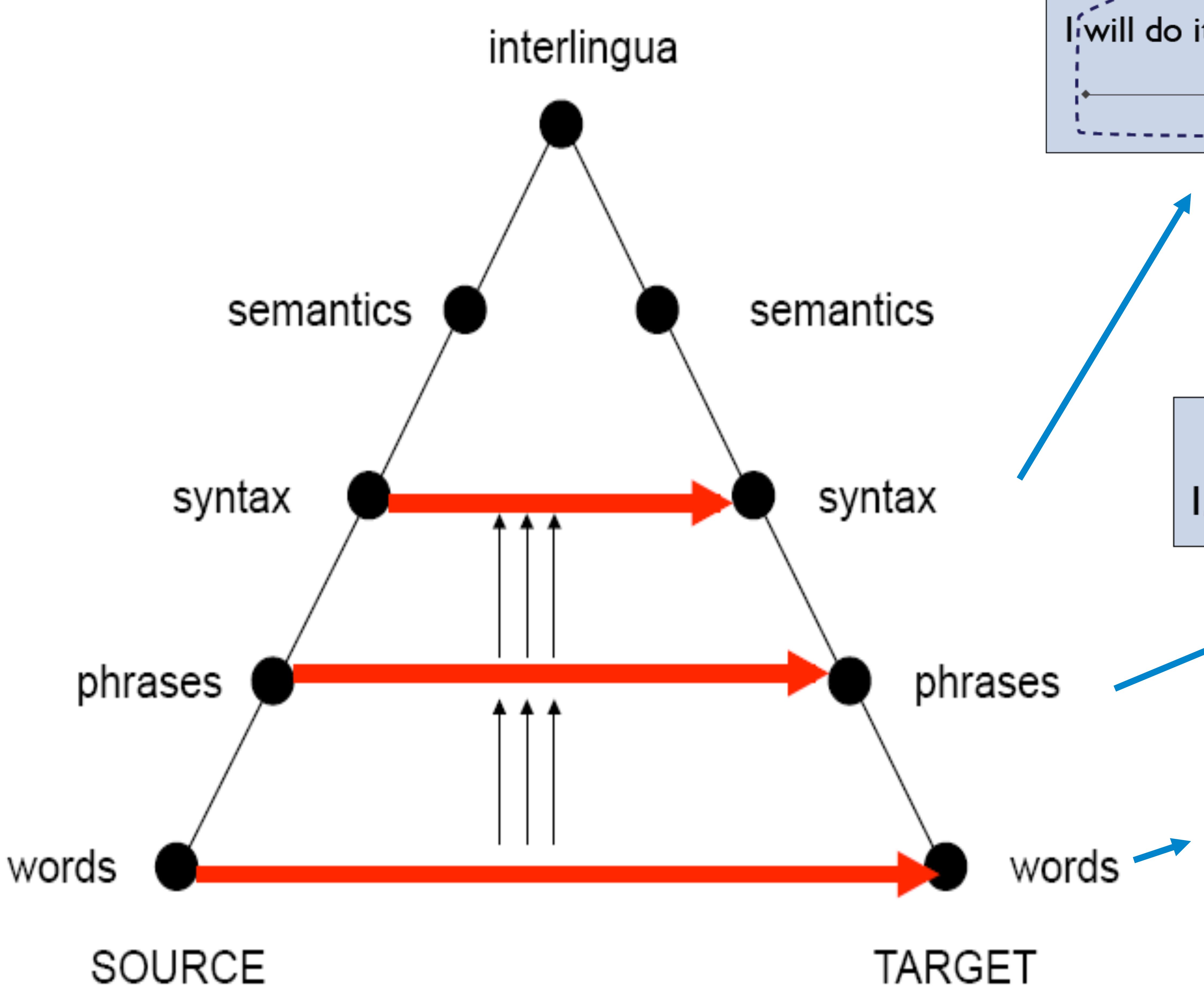
| | |
|---|---|
| Je fais un bureau | I’m making a desk |
| Je fais une soupe | I’m making soup |
| Je fais un bureau | I make a desk |
| Qu’est-ce que tu fais? | What are you making? |

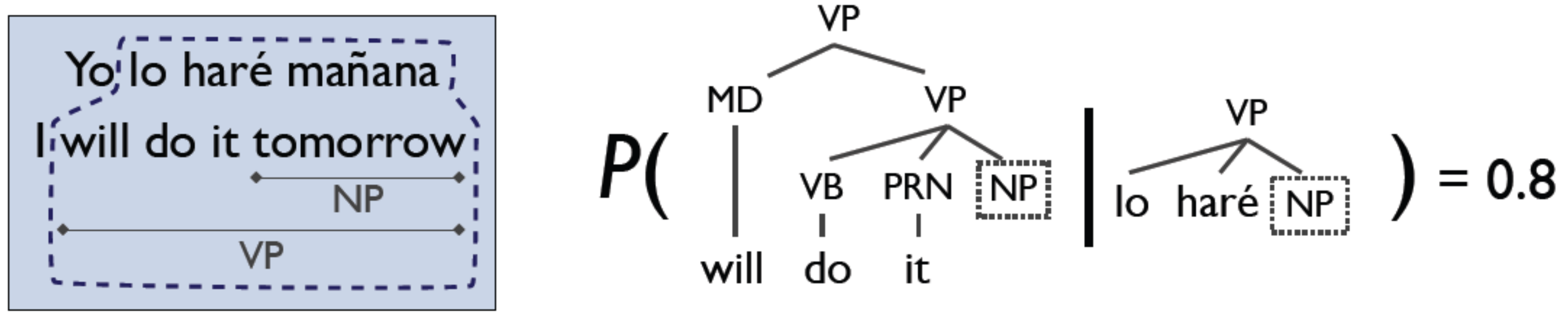
- What makes this hard?

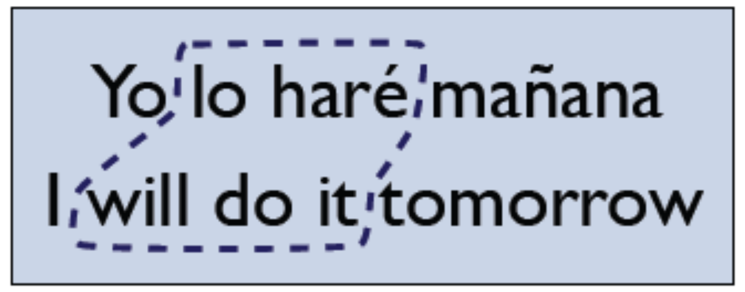
Not word-to-word translation

Multiple translations of a single source (ambiguous)

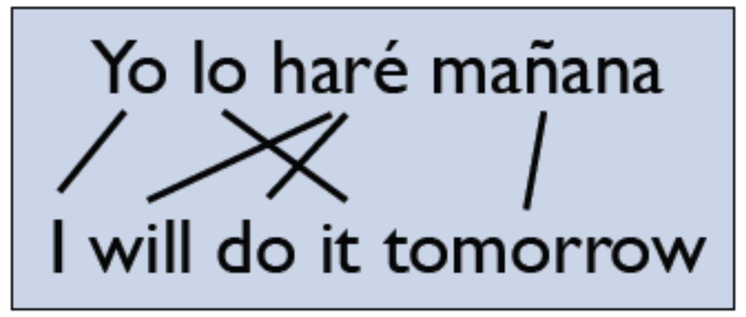
# Levels of Transfer: Vauquois Triangle




| *English* (E) | *P*( E \| lo haré ) |
|---|---|
| will do it | 0.8 |
| will do so | 0.2 |




| *English* (E) | *P*( E \| mañana ) |
|---|---|
| tomorrow | 0.7 |
| morning | 0.3 |

Slide credit: Dan Klein

# Phrase-based MT

- Consider translation as translating pairs of corresponding text from input and output, and then re-order them to finalize the output.
- Key idea: translation works better the bigger chunks you use

A doctor visited a friend last night.

Un médecin a rendu visite à une amie hier soir.

# Phrase-Based MT

- Remember phrases from training data, translate piece-by-piece and stitch those pieces together to translate
  - How to identify phrases? Word alignment over source-target bitext
  - How to stitch together? Language model over target language
  - Decoder takes phrases and a language model and searches over possible translations
- NOT like standard discriminative models
  - Not a single objective

# Brief History: MT

## Workshop on Statistical Machine Translation (WMT)

2019
- Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers) 13 papers
- Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1) 69 papers
- Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2) 42 papers

2018
- Proceedings of the Third Conference on Machine Translation: Research Papers 28 papers
- Proceedings of the Third Conference on Machine Translation: Shared Task Papers 90 papers

2017
- Proceedings of the Second Conference on Machine Translation 82 papers

2016
- Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers 14 papers
- Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers 94 papers

2015
- Proceedings of the Tenth Workshop on Statistical Machine Translation 61 papers

2014
- Proceedings of the Ninth Workshop on Statistical Machine Translation 64 papers

2013
- Proceedings of the Eighth Workshop on Statistical Machine Translation 65 papers

2012
- Proceedings of the Seventh Workshop on Statistical Machine Translation 61 papers

2011
- Proceedings of the Sixth Workshop on Statistical Machine Translation 69 papers

2010
- Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR 64 papers

2009
- Proceedings of the Fourth Workshop on Statistical Machine Translation 42 papers

2008
- Proceedings of the Third Workshop on Statistical Machine Translation 37 papers

2007
- Proceedings of the Second Workshop on Statistical Machine Translation 39 papers

2006
- Proceedings on the Workshop on Statistical Machine Translation 27 papers

Used to be a sub-community inside NLP
A very large overhead to get into the field.

Either you devote your Ph.D into machine translation, or you do not touch it.

# Phrase-Based MT

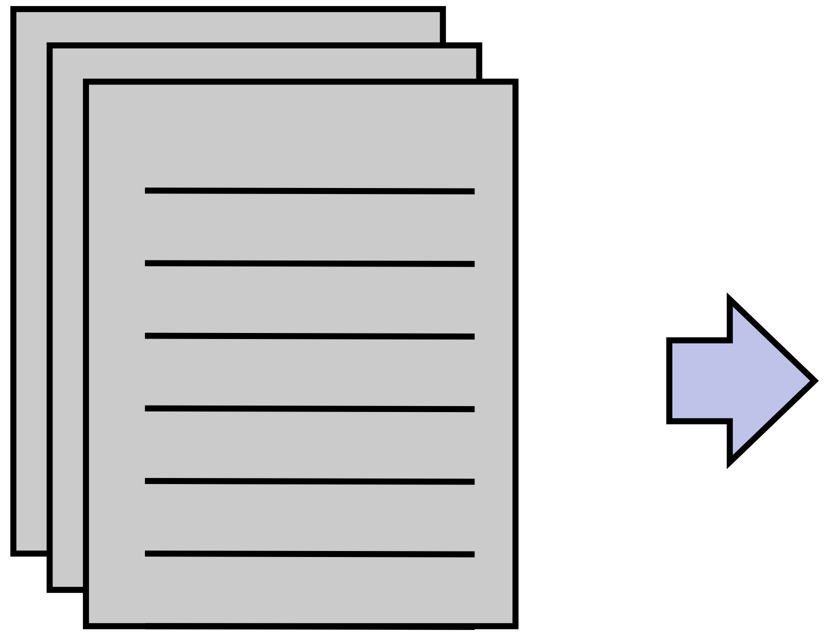
cat ||| chat ||| 0.9
the cat ||| le chat ||| 0.8
dog ||| chien ||| 0.8
house ||| maison ||| 0.6
my house ||| ma maison ||| 0.9
language ||| langue ||| 0.9
…

Phrase table P(f|e)

# Phrase-Based MT

cat ||| chat ||| 0.9
the cat ||| le chat ||| 0.8
dog ||| chien ||| 0.8
house ||| maison ||| 0.6
my house ||| ma maison ||| 0.9
language ||| langue ||| 0.9
…

Phrase table P(f|e)



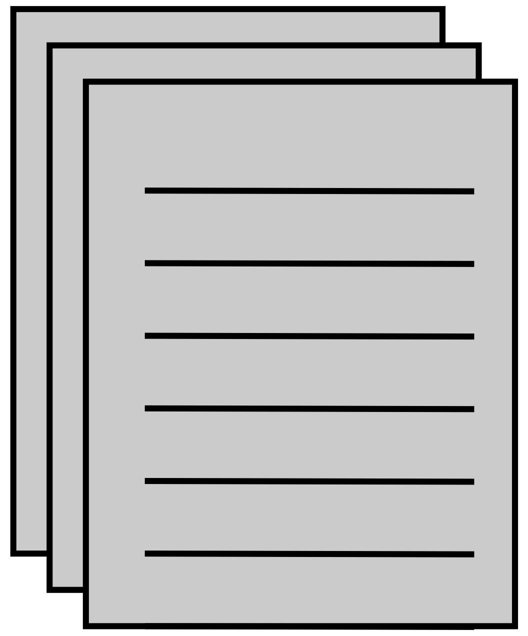
Language model P(e)

Unlabeled English data

# Phrase-Based MT

cat ||| chat ||| 0.9
the cat ||| le chat ||| 0.8
dog ||| chien ||| 0.8
house ||| maison ||| 0.6
my house ||| ma maison ||| 0.9
language ||| langue ||| 0.9
…

Phrase table P(f|e)



Unlabeled English data

Language model P(e)

$$P(e|f) \propto P(f|e)P(e)$$

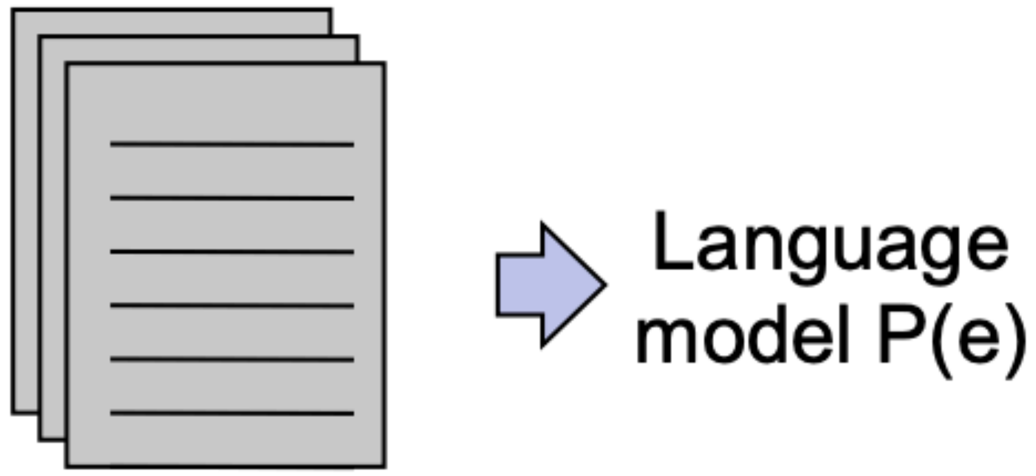
Noisy channel model: combine scores from translation model + language model to translate foreign to English

“Translate faithfully but make fluent English”

# Neural MT

```
cat ||| chat ||| 0.9
the cat ||| le chat ||| 0.8
dog ||| chien ||| 0.8
house ||| maison ||| 0.6
my house ||| ma maison ||| 0.9
language ||| langue ||| 0.9
...
```

Phrase table P(f|e)




Unlabeled English data

- No explicit phrase table (or replaced by a the concept of )

- The notion of language model still remains.

# Evaluating MT

- Fluency: does it sound good in the target language?
- Fidelity/adequacy: does it capture the meaning of the original?
- BLEU score: geometric mean of 1-, 2-, 3-, and 4-gram ***precision*** vs. a reference, multiplied by brevity penalty (BP) which penalizes short translations

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right).$$

- Typically $n = 4$, $w_i = 1/4$

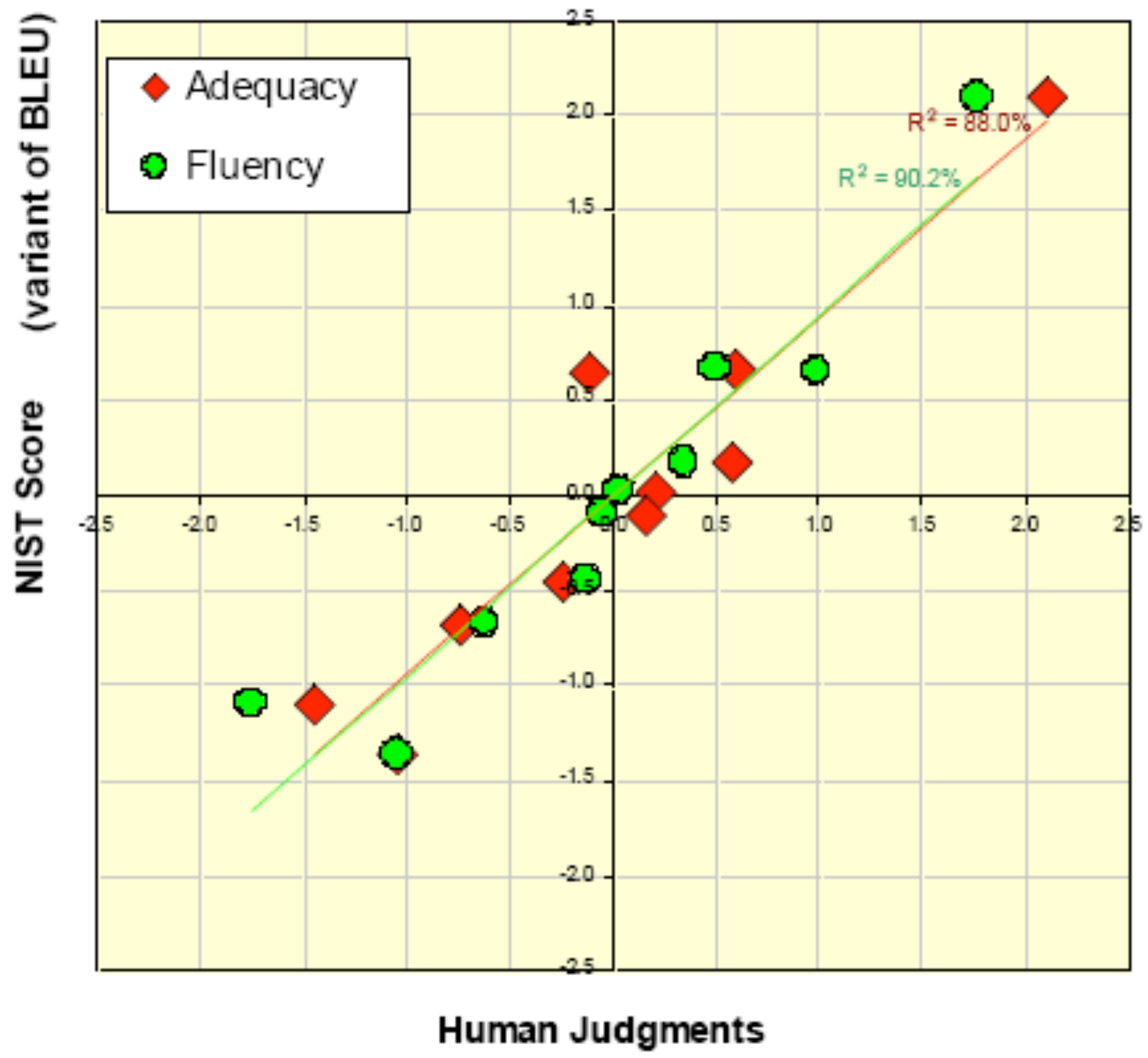
$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}.$$

r = length of reference

c = length of prediction

# BLEU Score

- At a *corpus* level, BLEU correlates pretty well with human judgments
- Better methods with human-in-the-loop
- If you're building real MT systems, you do user studies. In academia, you mostly use BLEU




- Newer learnt metrics (e.g., BLEURT, BERTScore) correlates better with human judgements

slide from G. Doddington (NIST)

# Word Alignment

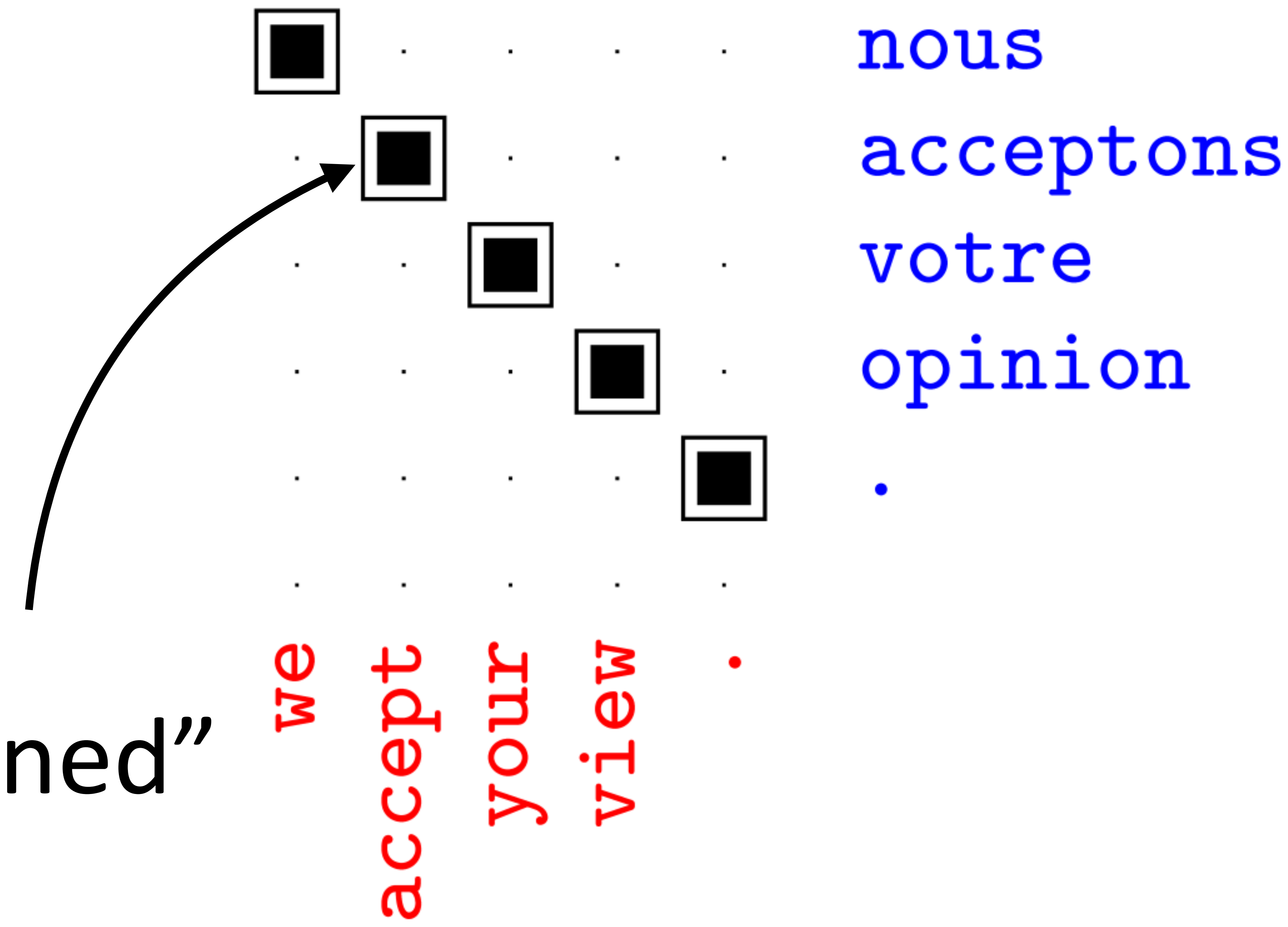
# Word Alignment

- Input: a bitext, pairs of translated sentences

  nous acceptons votre opinion . ||| we accept your view

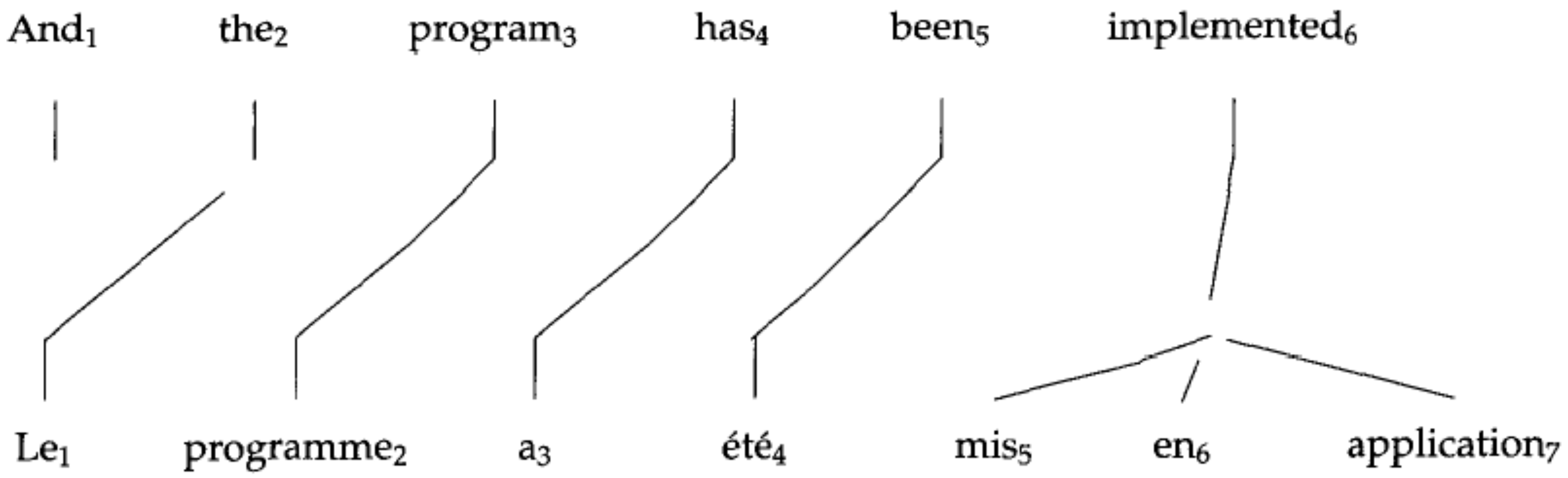
  nous allons changer d'avis ||| we are going to change our minds

- Output: alignments between words in each sentence
  - We will see how to turn these into phrases




"accept and acceptons are aligned"

# 1-to-Many Alignments




- Each output word is generated from a single input word!

# IBM Model 1 [Brown et al. (1993)]

- Translating to English sentence from French sentence.

- English sentence **e** has l words:
  French sentence **f** has m words:

- An alignment **a** identifies which English word each French word originated from.

- Formally, an alignment **a** is:

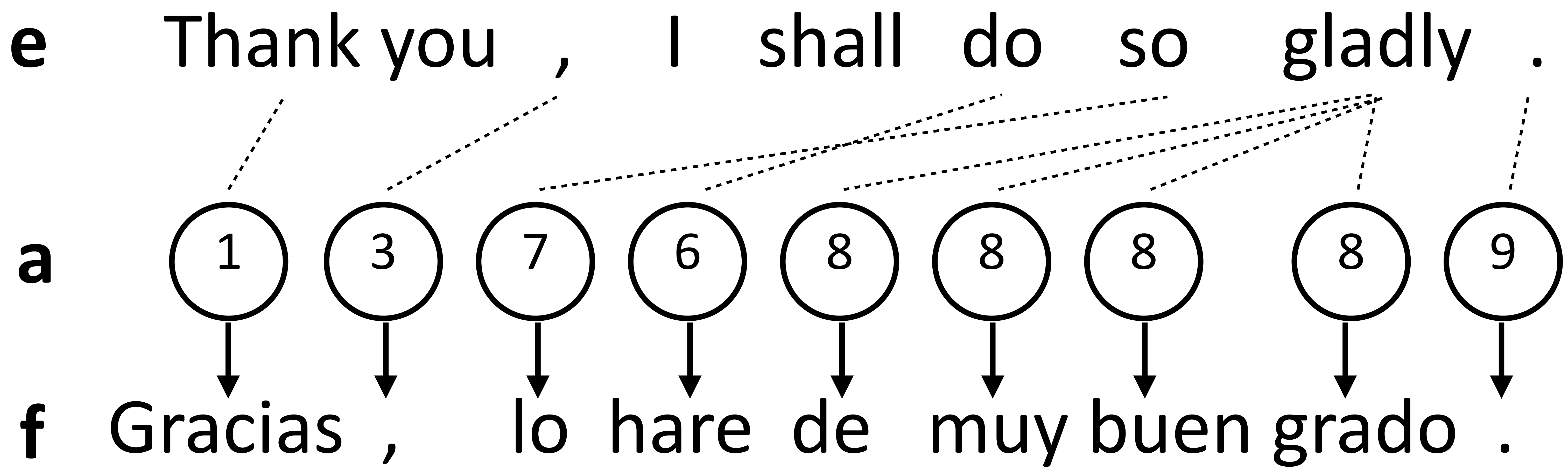
  $$\{a_1, \ldots, a_m\} \quad \text{where} \quad a_j \in 0 \ldots l$$

- How many potential alignments? $(l+1)^m$

# IBM Model 1 (1993)

- Each French word is aligned to *at most* one English word

$$p(f|a,e,m) = \prod_{j=1}^{m} t(f_j|e_{a_j})$$




- Set P(a) uniformly (no prior over good alignments)
- $t(f_j|e_{a_j})$: word translation probability table

# IBM Model 1: alignment

$l = 6, m = 7$

$e =$ And the program has been implemented

$f =$ Le programme a ete mis en application

- One alignment is

$$\{2, 3, 4, 5, 6, 6, 6\}$$

# IBM Model 1: alignment

$l = 6, m = 7$

$e =$ And the program has been implemented

$f =$ Le programme a ete mis en application

- Another (bad!) alignment is

$$\{1, 1, 1, 1, 1, 1, 1\}$$

# IBM Model 1: alignment

$l = 6, m = 7$

$e =$ And the program has been implemented

$f =$ Le programme a ete mis en application

- Another (bad!) alignment is

$$\{1, 1, 1, 1, 1, 1, 1\}$$

# IBM 1 Model

- Models P(**f**|**e**): probability of "French" sentence being generated from "English" sentence according to a model
  - We define two models:

$$p(a|e,m) \qquad\qquad p(f|a,e,m)$$

  - Giving:

$$p(f,a|e,m) = p(a|e,m)p(f|a,e,m)$$

  - Also:

$$p(f|e,m) = \sum_{a\in\mathcal{A}} p(a|e,m)p(f|a,e,m)$$

where $A$ is a set of all possible alignments

# IBM Model 1: Alignment

- In IBM Model 1 all alignments *a* are equally likely:

$$p(a|e, m) = \frac{1}{(1+l)^m}$$

- Reasonable assumption?
  - Simplifying assumption, but it gets things started …

# IBM Model 1: Transition Probability

- Next step: come up with an estimate for

$$p(f|a, e, m)$$

- In Model 1, this is:

$$p(f|a, e, m) = \prod_{j=1}^{m} t(f_j|e_{a_j})$$

# IBM Model 1: Example

$l = 6, m = 7$

$e =$ And the program has been implemented

$f =$ Le programme a ete mis en application

$$a = \{2, 3, 4, 5, 6, 6, 6\}$$

$$\begin{aligned} p(f|a, e) = & t(\text{Le}|\text{the}) \times t(\text{programme}|\text{program}) \\ & \times t(\text{a}|\text{has}) \times t(\text{ete}|\text{been}) \\ & \times t(\text{mis}|\text{implemented}) \times t(\text{en}|\text{implemented}) \\ & \times t(\text{application}|\text{implemented}) = 0.0006804 \end{aligned}$$

$$p(f, a \mid e, 7) = 8.26186E - 10$$

# IBM Model 1: Generative Process

To generate a French string $f$ from an English string $e$:

- Step 1: Pick an alignment $a$ with probability $\frac{1}{(l+1)^m}$
- Step 2: Pick the French words with probability

$$p(f|a,e,m) = \prod_{j=1}^{m} t(f_j|e_{a_j})$$

The final result:

$$p(f,a|e,m) = p(a|e,m) \times p(f|a,e,m) = \frac{1}{(1+l)^m} \prod_{j=1}^{m} t(f_j|e_{a_j})$$

# IBM Model 2:

- Only difference: we now introduce alignment distortion parameters

$$q(i|j, l, m)$$

Probability that *j*'th French word is connected to *i*'th English word, given sentence length of *e* and *f* are *l* and *m*

- Define

$$p(a|e, m) = \prod_{j=1}^{m} q(a_j|j, l, m)$$

$$a = \{a_1, \ldots, a_m\}$$

$$p(f, a|e, m) = \prod_{j=1}^{m} q(a_j|j, l, m) t(f_j|e_{a_j})$$

# IBM Model 2: example

$$
\begin{aligned}
l &= 6 \\
m &= 7 \\
e &= \text{And the program has been implemented} \\
f &= \text{Le programme a ete mis en application} \\
a &= \{2,3,4,5,6,6,6\}
\end{aligned}
$$

$$
\begin{aligned}
p(a \mid e, 7) = \; & \mathbf{q}(2 \mid 1, 6, 7) \times \\
& \mathbf{q}(3 \mid 2, 6, 7) \times \\
& \mathbf{q}(4 \mid 3, 6, 7) \times \\
& \mathbf{q}(5 \mid 4, 6, 7) \times \\
& \mathbf{q}(6 \mid 5, 6, 7) \times \\
& \mathbf{q}(6 \mid 6, 6, 7) \times \\
& \mathbf{q}(6 \mid 7, 6, 7)
\end{aligned}
$$

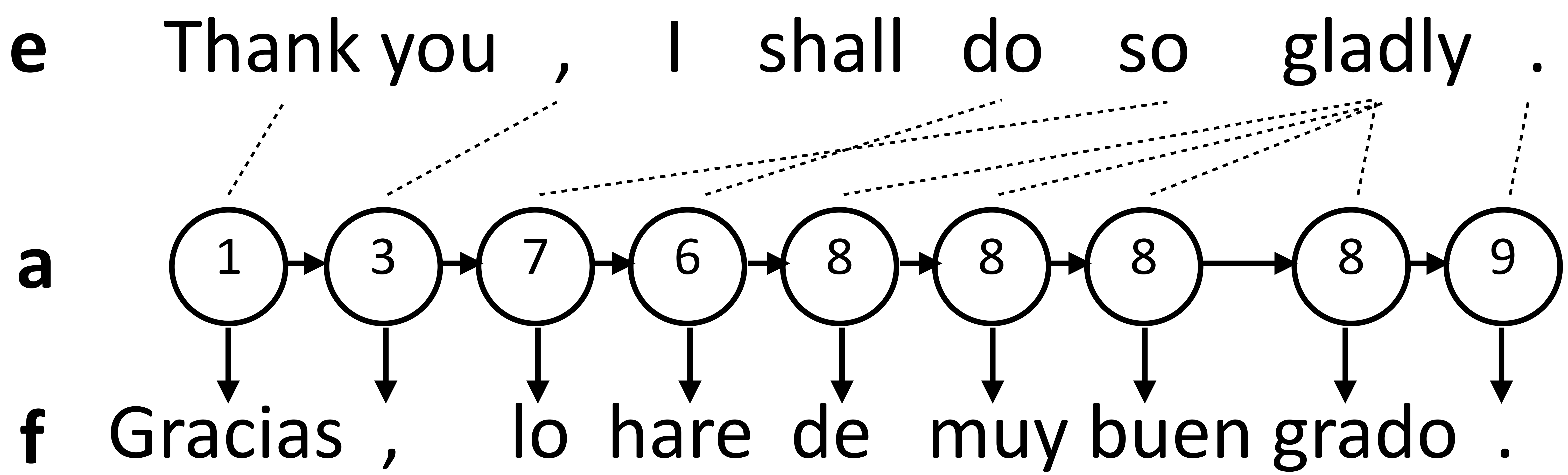
$$
\begin{aligned}
p(f \mid a, e, 7) = \; & \mathbf{t}(Le \mid the) \times \\
& \mathbf{t}(programme \mid program) \times \\
& \mathbf{t}(a \mid has) \times \\
& \mathbf{t}(ete \mid been) \times \\
& \mathbf{t}(mis \mid implemented) \times \\
& \mathbf{t}(en \mid implemented) \times \\
& \mathbf{t}(application \mid implemented)
\end{aligned}
$$

$$p(f, a|e, m) = p(a|e, m) \times p(f|a, e, m) = \prod_{j=1}^{m} q(a_j|j, l, m) t(f_j|e_{a_j})$$

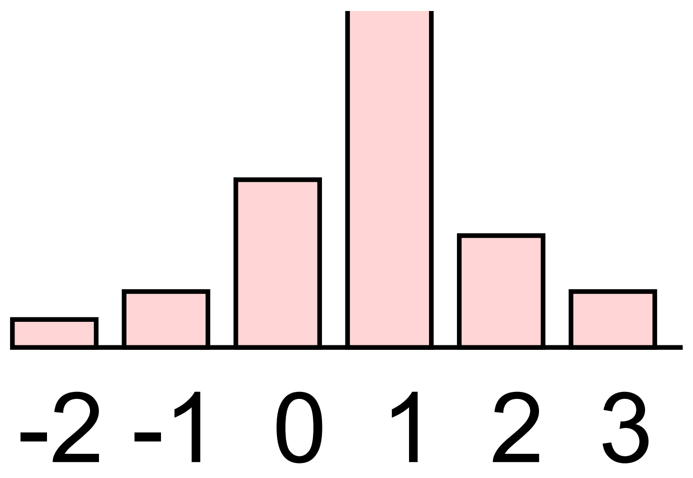
# Further Improvement: HMM for Alignment

- Sequential dependence between a's to capture monotonicity

$$P(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) = \prod_{j=1}^{m} t(f_j \mid e_{a_j}) P(a_j \mid a_{j-1})$$




- Alignment dist parameterized by jump size: $P(a_j - a_{j-1})$ $\longrightarrow$



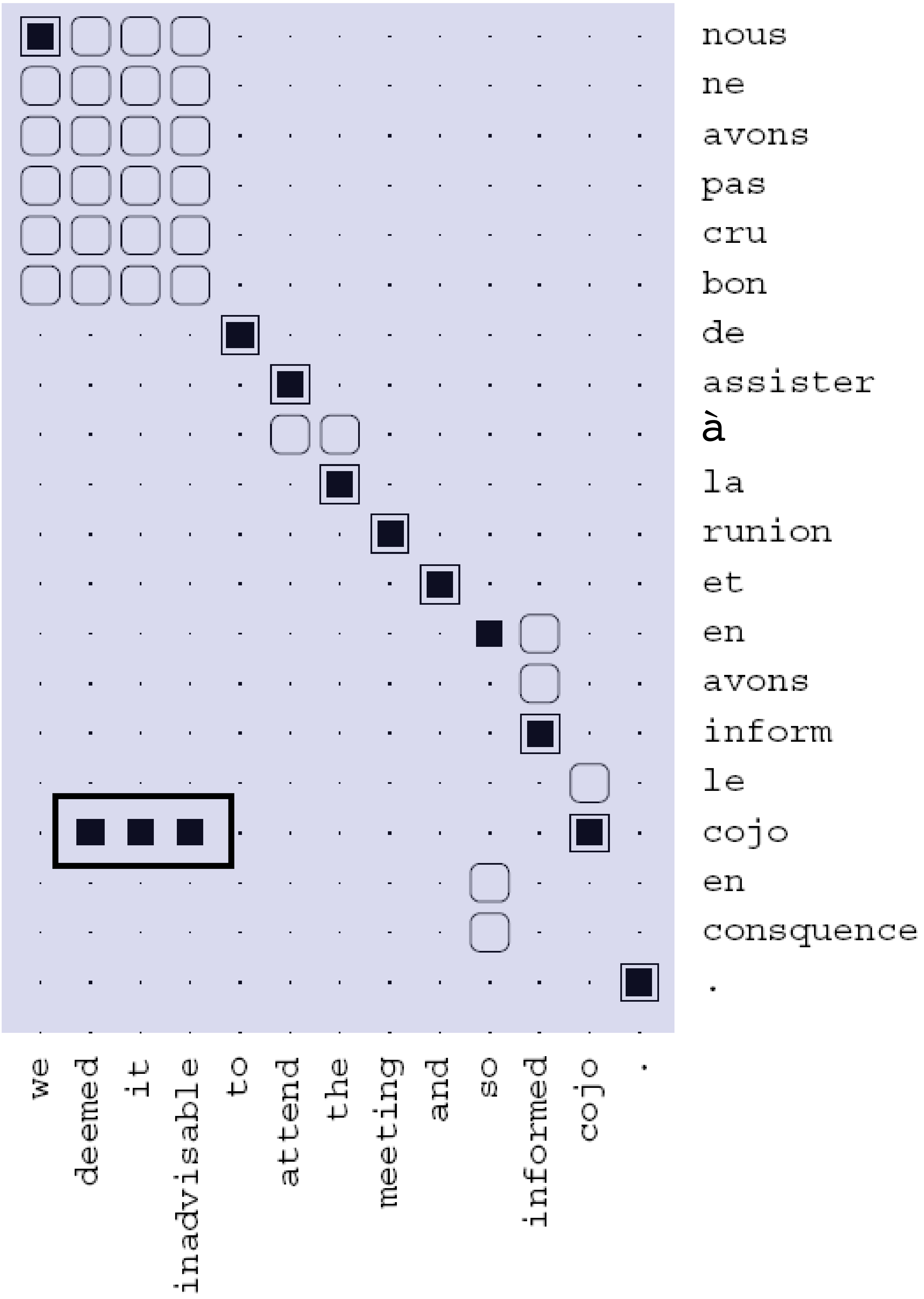

- $t(f_j|e_{a_j})$: same as before

Vogel et al. (1996)

# HMM Model

- Alignments are generally monotonic (along diagonal), for similar language pairs

- Some mistakes, especially when you have rare words



# Evaluating Word Alignment

- "Alignment error rate": use labeled alignments on small corpus

| Model | AER |
| --- | --- |
| Model 1 INT | 19.5 |
| HMM E→F | 11.4 |
| HMM F→E | 10.8 |
| HMM INT | 4.7 |

- Run Model 1 in both directions and intersect them
- Run HMM model in both directions and intersect them

# Phrase Extraction

- Find contiguous sets of aligned words in the two languages that don't have alignments to other words

  d'assister à la reunion et ||| to attend the meeting and

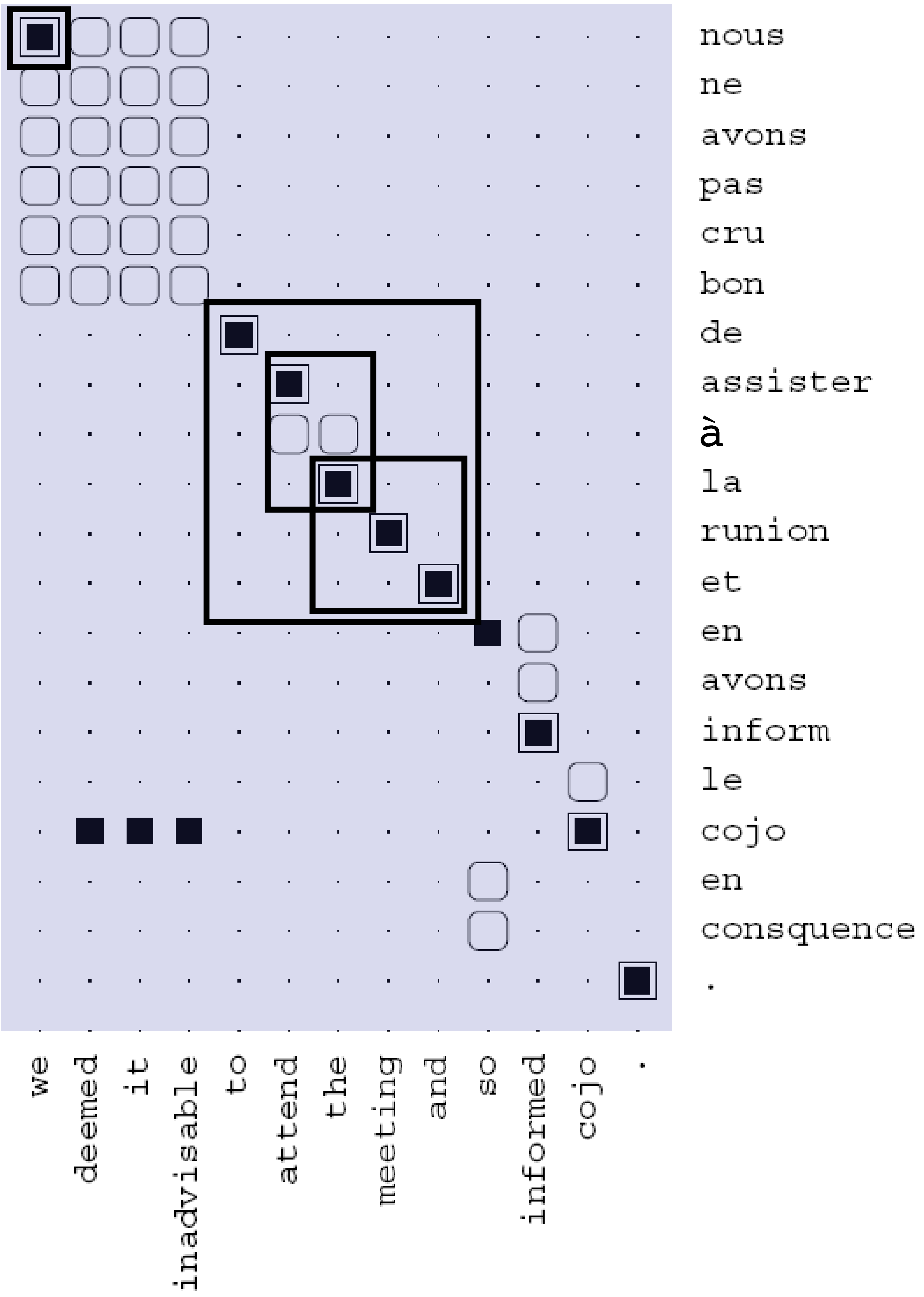
  assister à la reunion ||| attend the meeting

  la reunion and ||| the meeting and

  nous ||| we

  …

- Lots of phrases possible, count across all sentences and score by frequency



# Parameter Estimation: Easy

- Assume alignments are observed in training data
  $e^{(100)}$ = And the program has been implemented
  $f^{(100)}$ = Le programme a ete mis en application
  $a^{(100)}$ = <2,3,4,5,6,6,6>

- Training data is

$$(e^{(k)}, f^{(k)}, a^{(k)}), k = 1 \dots n$$

  Each $e^{(k)}$ is an English sentence, each $f^{(k)}$ is a French sentence, each $a^{(k)}$ is an alignment

- Maximum-likelihood parameter estimates are trivial:

$$t_{ML}(f|e) = \frac{\text{count}(e, f)}{\text{count}(e)} \qquad q_{ML}(j|i, l, m) = \frac{\text{count}(j, i, l, m)}{\text{count}(i, l, m)}$$

# Pseudocode

**Input:** A training corpus $(f^{(k)}, e^{(k)}, a^{(k)})$ for $k = 1 \ldots n$, where $f^{(k)} = f_1^{(k)} \ldots f_{m_k}^{(k)}$, $e^{(k)} = e_1^{(k)} \ldots e_{l_k}^{(k)}$, $a^{(k)} = a_1^{(k)} \ldots a_{m_k}^{(k)}$.

**Algorithm:**

- Set all counts $c(\ldots) = 0$
- For $k = 1 \ldots n$

# Pseudocode

**Input:** A training corpus $(f^{(k)}, e^{(k)}, a^{(k)})$ for $k = 1 \ldots n$, where $f^{(k)} = f_1^{(k)} \ldots f_{m_k}^{(k)}$, $e^{(k)} = e_1^{(k)} \ldots e_{l_k}^{(k)}$, $a^{(k)} = a_1^{(k)} \ldots a_{m_k}^{(k)}$.

**Algorithm:**

- Set all counts $c(\ldots) = 0$
- For $k = 1 \ldots n$
  - For $i = 1 \ldots m_k$, For $j = 0 \ldots l_k$,

$$
\begin{aligned}
c(e_j^{(k)}, f_i^{(k)}) &\leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k, i, j) \\
c(e_j^{(k)}) &\leftarrow c(e_j^{(k)}) + \delta(k, i, j) \\
c(j|i, l, m) &\leftarrow c(j|i, l, m) + \delta(k, i, j) \\
c(i, l, m) &\leftarrow c(i, l, m) + \delta(k, i, j)
\end{aligned}
$$

where $\delta(k, i, j) = 1$ if $a_i^{(k)} = j$, 0 otherwise.

Observed count of word $f_i$ to be aligned to $e_j$ in the data.

**Output:** $t_{ML}(f|e) = \frac{c(e,f)}{c(e)}$, $q_{ML}(j|i,l,m) = \frac{c(j|i,l,m)}{c(i,l,m)}$