Phrase-Based MT

Machine Translation
Lecture 7

Instructor: Chris Callison-Burch
TAs: Mitchell Stern, Justin Chiu

Website: mt-class.org/penn

Translational Equivalence

Er hat die Prüfung ***bestanden****, jedoch nur knapp*

He **insisted on** the test, but just barely.

He **passed** the test, but just barely.

How do lexical translation models deal with contextual information?

Translational Equivalence

*Er hat die Prüfung **bestanden**, jedoch nur knapp*

He **insisted on** the test, but just barely.

He **passed** the test, but just barely.

F	E	prob
bestanden	**insisted**	0.06
	were	0.06
	existed	0.04
	was	0.04
	been	0.04
	passed	0.03
	consist	0.01

Translational Equivalence

*Er hat die Prüfung **bestanden**, jedoch nur knapp*

He **insisted on** the test, but just barely.

He **passed** the test, but just barely.

Lexical Translation

What is wrong with this?

How can we improve this?

Translation model

- What are the atomic units?
    - Lexical translation: **words**
    - Phrase-based translation: **phrases**
- **Standard model used by Google, Microsoft ...**
- Benefits
    - many-to-many translation
    - use of local context in translation
- Downsides
    - Where do phrases comes from?

Translation model

- With a **latent variable**, we introduce a decomposition into **phrases** which translate **independently**:

$$p(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) = p(\mathbf{a}) \prod_{\langle \bar{\mathbf{e}}, \bar{\mathbf{f}} \rangle \in \mathbf{a}} p(\bar{\mathbf{f}} \mid \bar{\mathbf{e}})$$

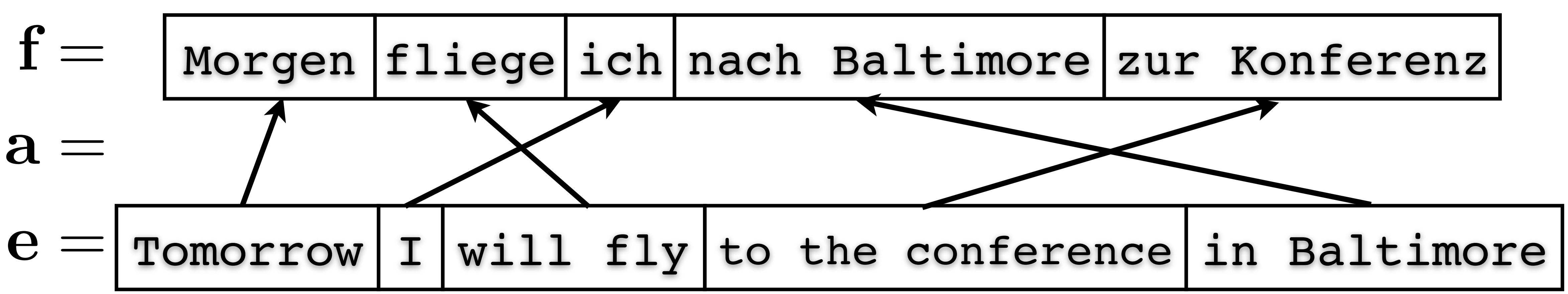
$\mathbf{f} =$ `Morgen fliege ich nach Baltimore zur Konferenz`

$\mathbf{e} =$ `Tomorrow I will fly to the conference in Baltimore`

Translation model

- With a **latent variable**, we introduce a decomposition into **phrases** which translate **independently**:

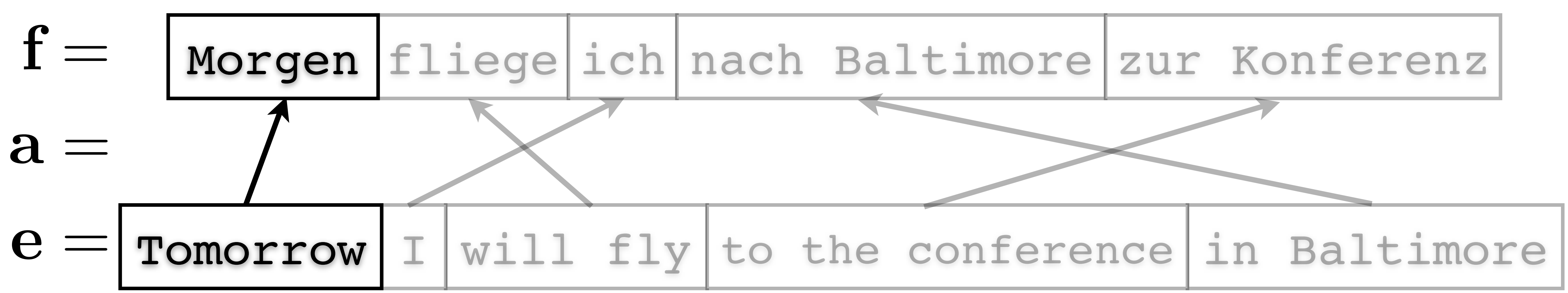
$$p(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) = p(\mathbf{a}) \prod_{\langle \bar{\mathbf{e}}, \bar{\mathbf{f}} \rangle \in \mathbf{a}} p(\bar{\mathbf{f}} \mid \bar{\mathbf{e}})$$



Translation model

- With a **latent variable**, we introduce a decomposition into **phrases** which translate **independently**:

$$p(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) = p(\mathbf{a}) \prod_{\langle \bar{\mathbf{e}}, \bar{\mathbf{f}} \rangle \in \mathbf{a}} p(\bar{\mathbf{f}} \mid \bar{\mathbf{e}})$$

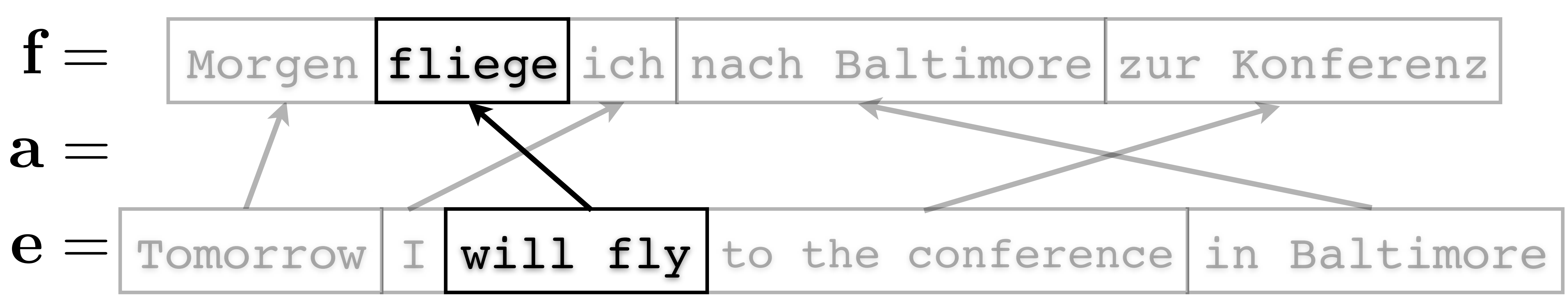


p(Morgen|Tomorrow)

Translation model

- With a **latent variable**, we introduce a decomposition into **phrases** which translate **independently**:

$$p(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) = p(\mathbf{a}) \prod_{\langle \bar{\mathbf{e}}, \bar{\mathbf{f}} \rangle \in \mathbf{a}} p(\bar{\mathbf{f}} \mid \bar{\mathbf{e}})$$

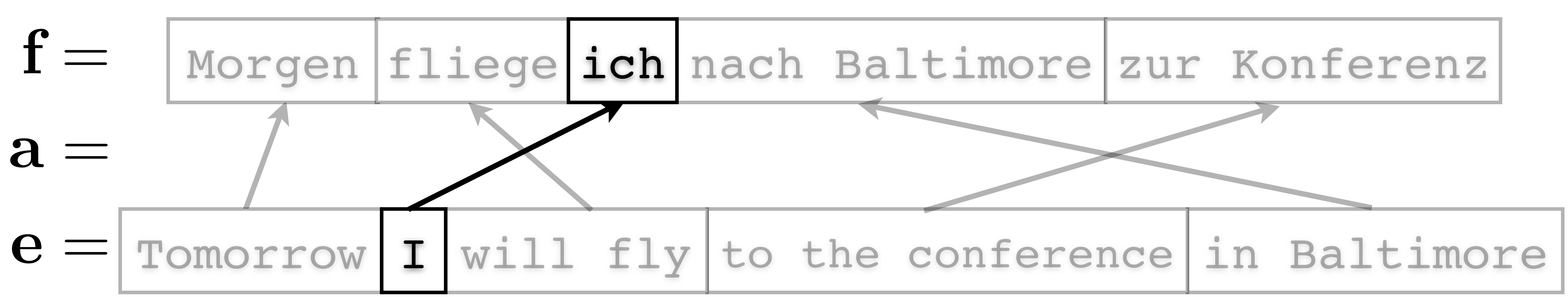


p(Morgen|Tomorrow) x *p*(fliege|will fly)

Translation model

- With a **latent variable**, we introduce a decomposition into **phrases** which translate **independently**:

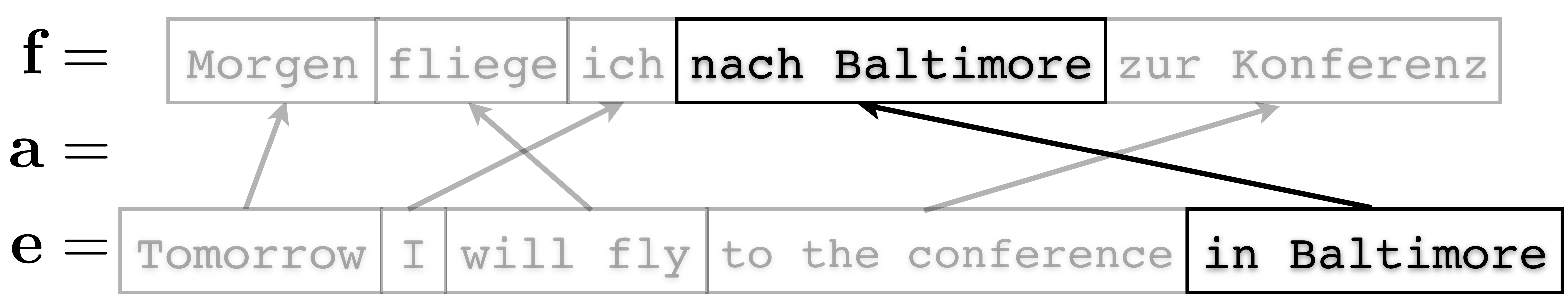
$$p(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) = p(\mathbf{a}) \prod_{\langle \bar{\mathbf{e}}, \bar{\mathbf{f}} \rangle \in \mathbf{a}} p(\bar{\mathbf{f}} \mid \bar{\mathbf{e}})$$



p(Morgen|Tomorrow) x p(fliege|will fly) x p(ich|I)

- With a **latent variable**, we introduce a decomposition into **phrases** which translate **independently**:

$$p(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) = p(\mathbf{a}) \prod_{\langle \bar{\mathbf{e}}, \bar{\mathbf{f}} \rangle \in \mathbf{a}} p(\bar{\mathbf{f}} \mid \bar{\mathbf{e}})$$



p(Morgen|Tomorrow) x *p*(fliege|will fly) x *p*(ich|I) x ...

Translation model

- With a **latent variable**, we introduce a decomposition into **phrases** which translate **independently**:

$$p(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) = p(\mathbf{a}) \prod_{\langle \bar{\mathbf{e}}, \bar{\mathbf{f}} \rangle \in \mathbf{a}} p(\bar{\mathbf{f}} \mid \bar{\mathbf{e}})$$

Marginalize to get p(f|e):

$$p(\mathbf{f} \mid \mathbf{e}) = \sum_{\mathbf{a} \in \mathcal{A}} p(\mathbf{a}) \prod_{\langle \bar{\mathbf{e}}, \bar{\mathbf{f}} \rangle \in \mathbf{a}} p(\bar{\mathbf{f}} \mid \bar{\mathbf{e}})$$

Phrases

- Contiguous strings of words
- Phrases are not necessarily syntactic constituents
- Usually have maximum limits
- Phrases subsume words (individual words are phrases of length 1)

Linguistic Phrases

- Model is not limited to linguistic phrases (NPs, VPs, PPs, CPs...)
- Non-constituent phrases are useful

es gibt   there is | there are

- Is a "good" phrase more likely to be
  [P NP]   or   [governor P]
  Why? How would you figure this out?

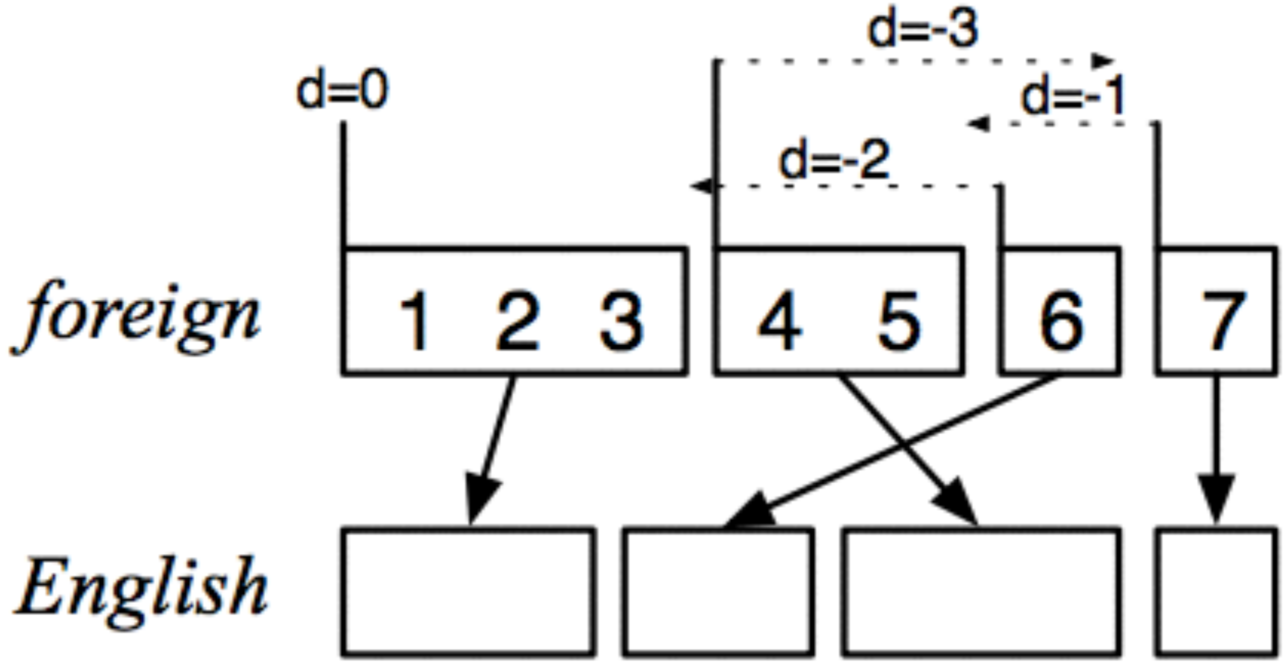
Phrase Tables

$\bar{\mathbf{f}}$	$\bar{\mathbf{e}}$	$p(\bar{\mathbf{f}} \mid \bar{\mathbf{e}})$
das Thema	the issue	0.41
	the point	0.72
	the subject	0.47
	the thema	0.99
es gibt	there is	0.96
	there are	0.72
morgen	tomorrow	0.9
fliege ich	will I fly	0.63
	will fly	0.17
	I will fly	0.13

p(a)

- Two responsibilities
  - Divide the source sentence into phrases
    - Standard approach: uniform distribution over all possible segmentations
    - How many segmentations are there?
  - Reorder the phrases
    - Standard approach: Markov model on phrases (parameterized with log-linear model)

Reordering Model



phrase	translates	movement	distance
1	1–3	start at beginning	0
2	6	skip over 4–5	+2
3	4–5	move back over 4–6	-3
4	7	skip over 6	+1

Scoring function: $d(x) = \alpha^{|x|}$ — exponential with distance

Learning Phrases

- Latent segmentation variable
- Latent phrasal inventory
- Parallel data
  - EM?

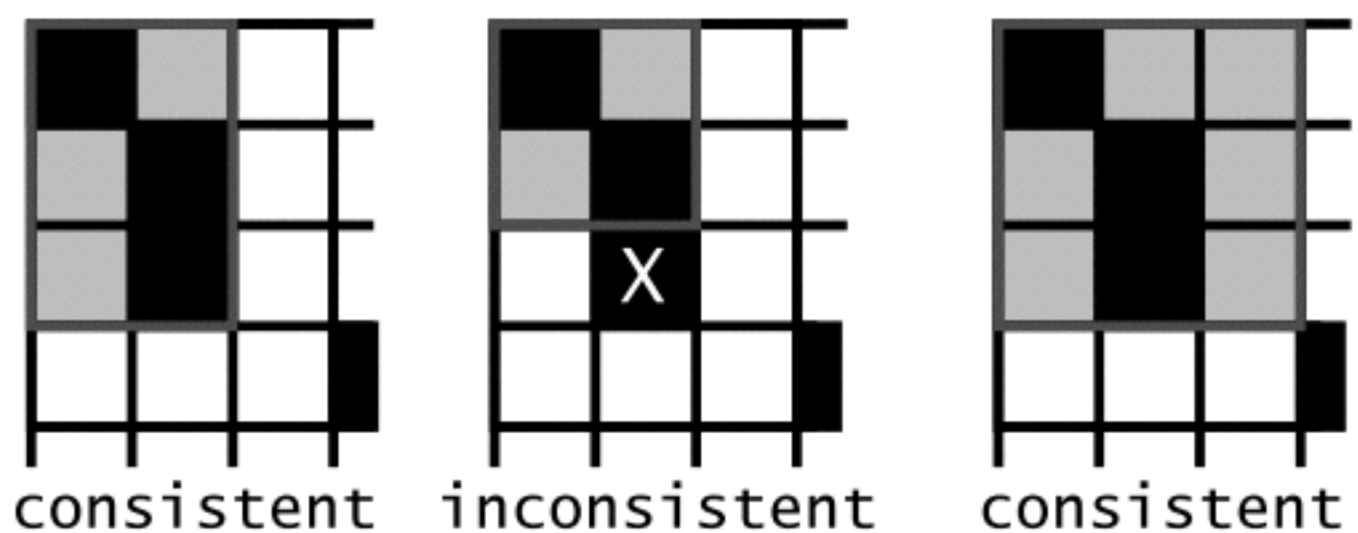
Computational problem: summing over all segmentations and alignments is #P-complete

Modeling problem: MLE has a degenerate solution.

Learning Phrases

- Three stages
  - word alignment
  - extraction of phrases
  - estimation of phrase probabilities

Consistent Phrases



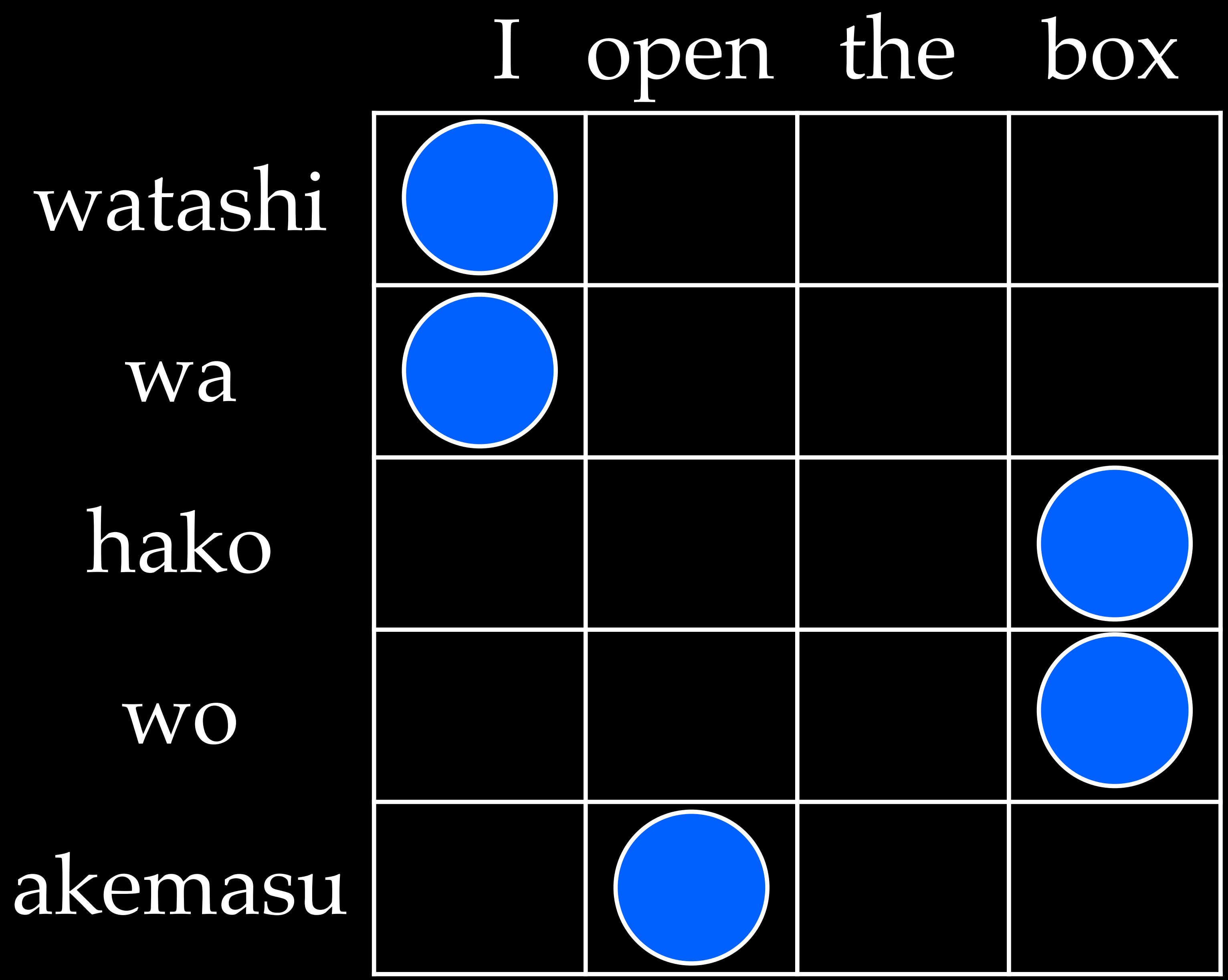
ok

violated
one alignment
point outside

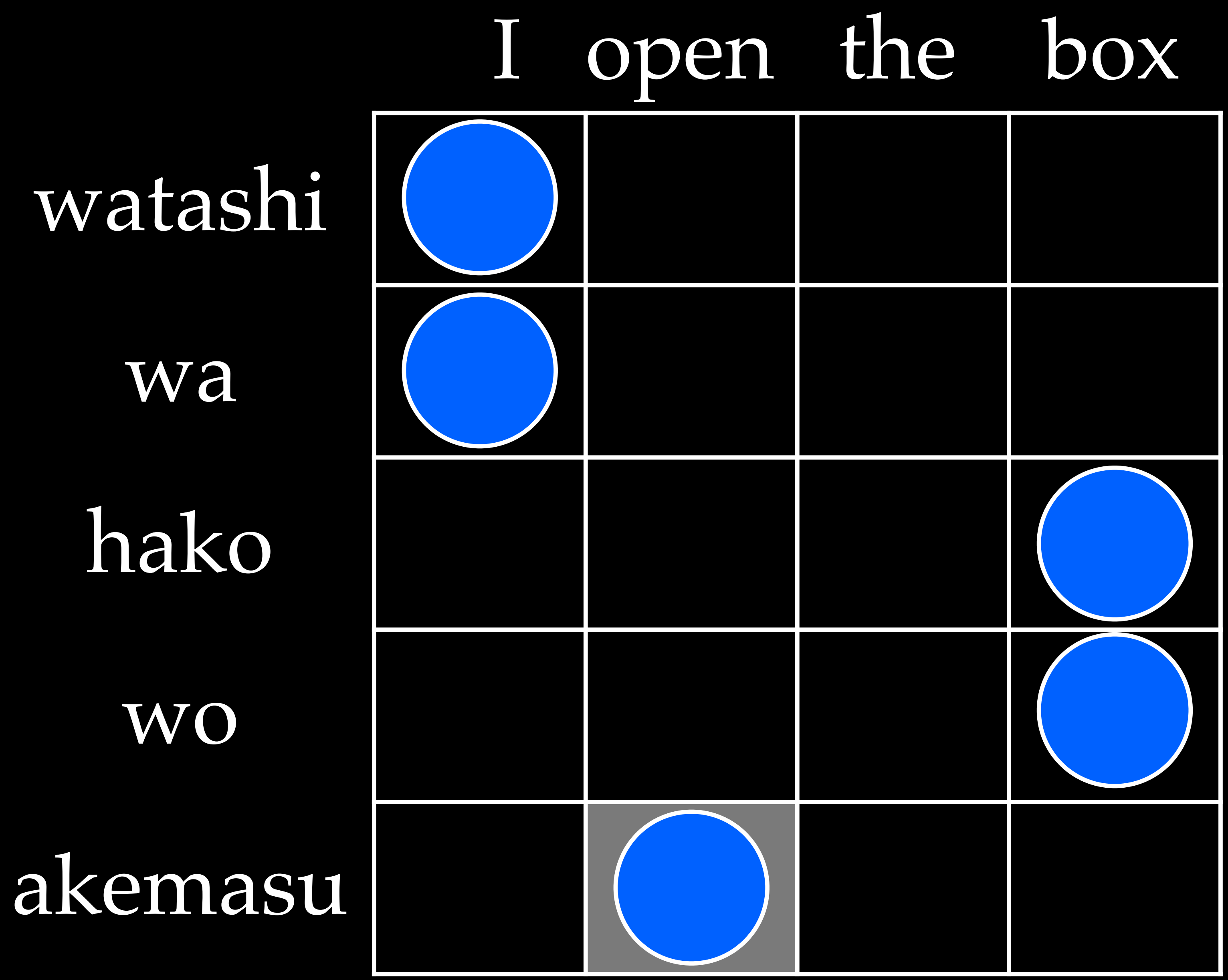
ok
unaligned
word is fine

All words of the phrase pair have to align to each other.

Phrase Extraction

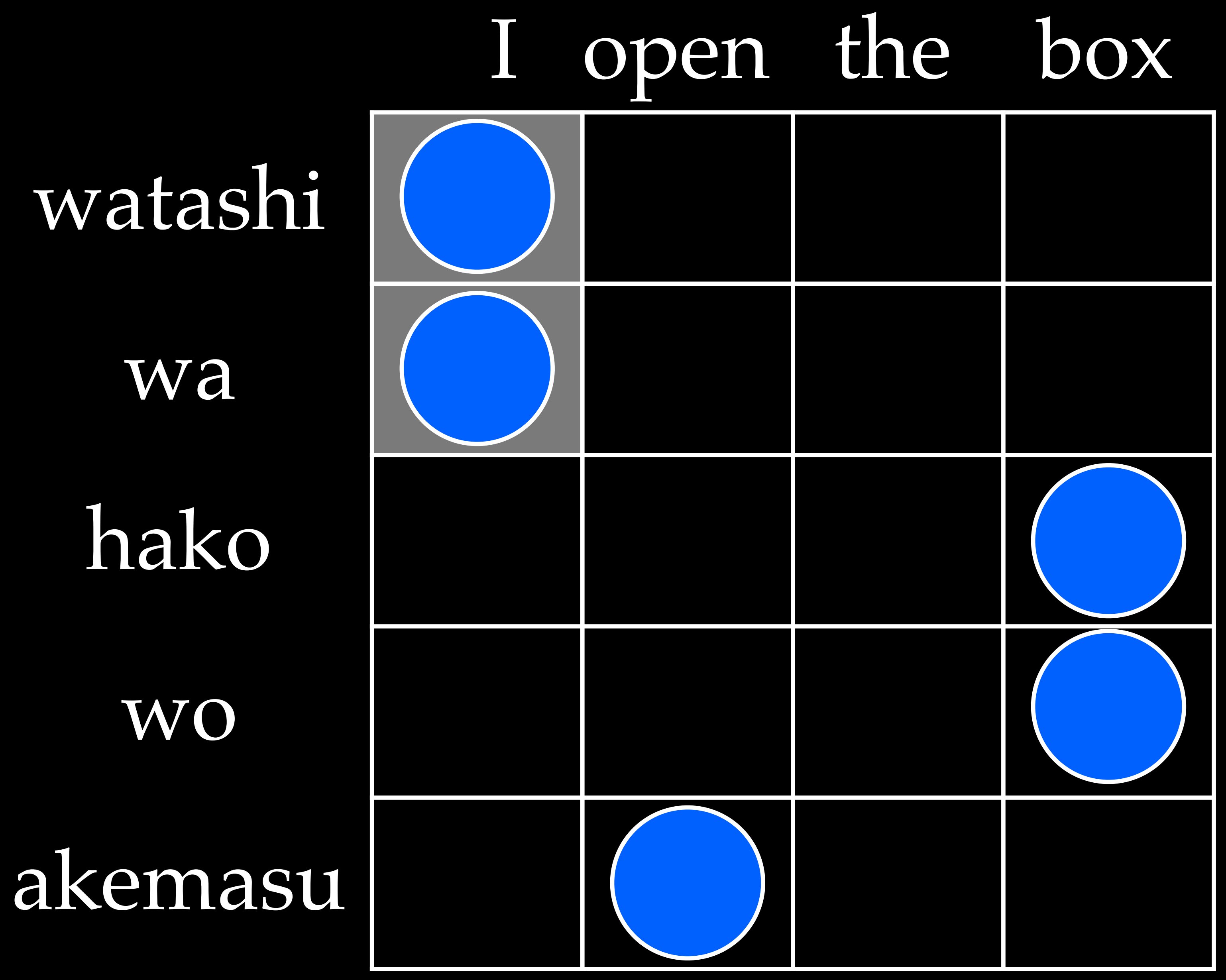


Phrase Extraction



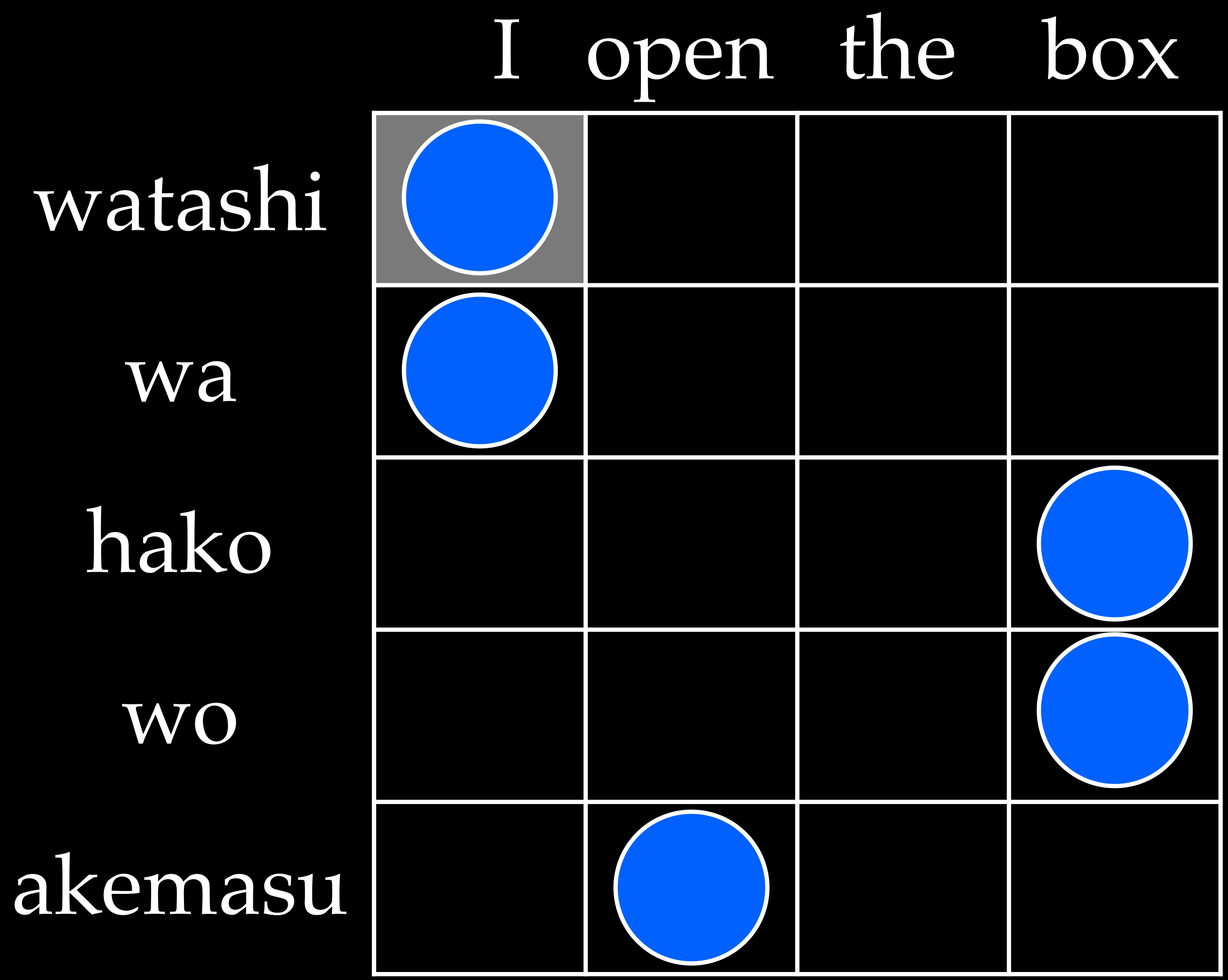
akemasu / open

Phrase Extraction



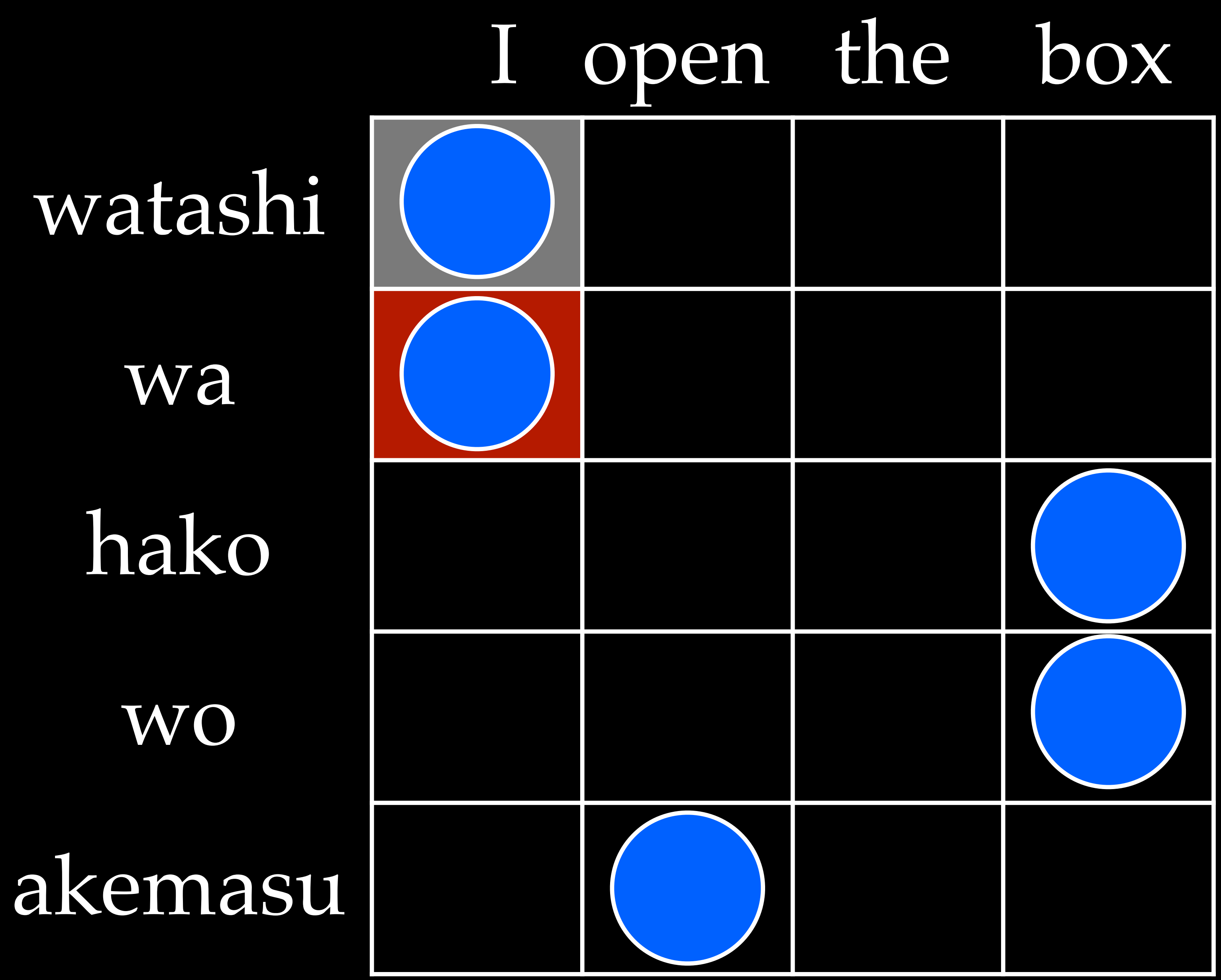
watashi wa / I

Phrase Extraction



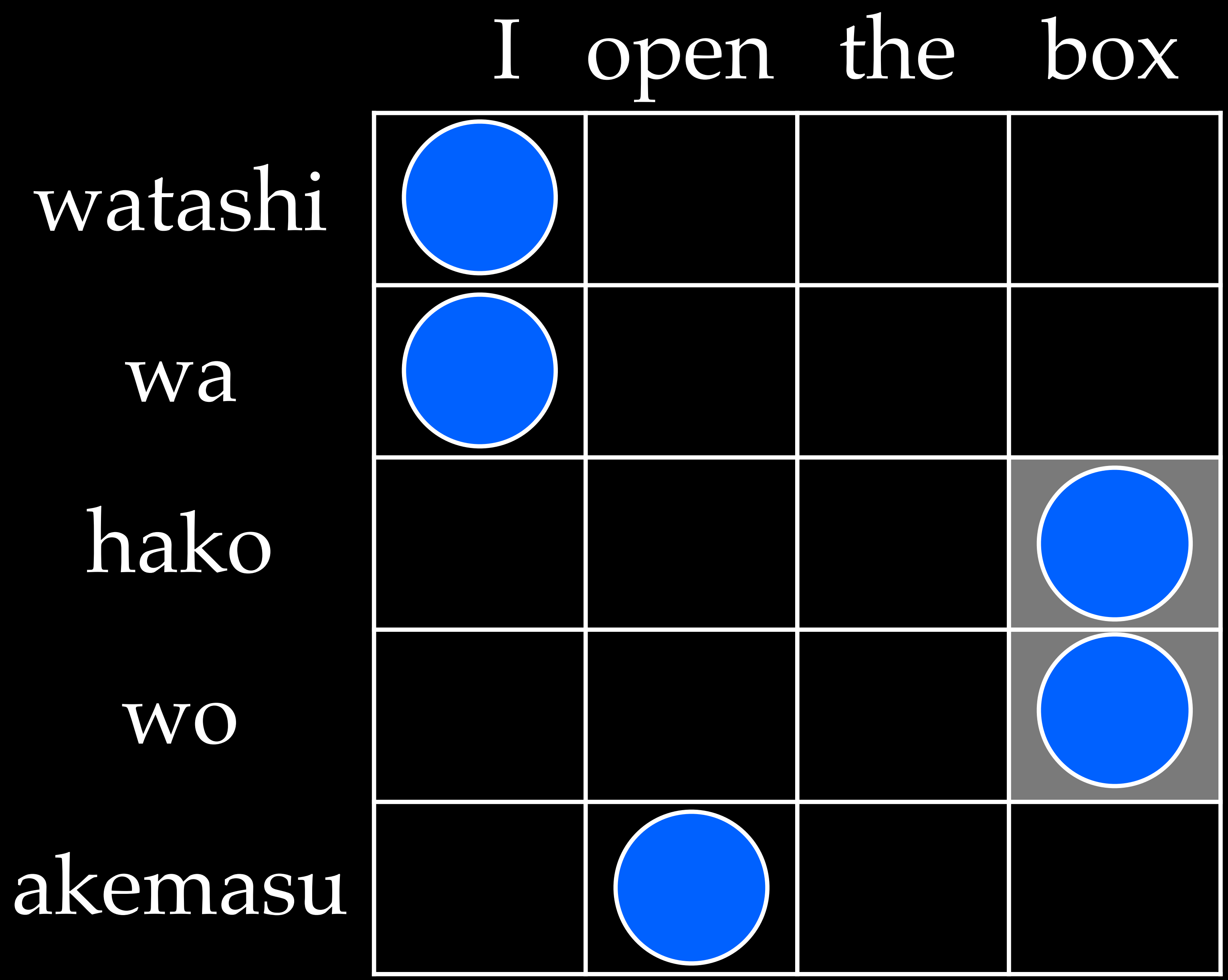
watashi / I

Phrase Extraction



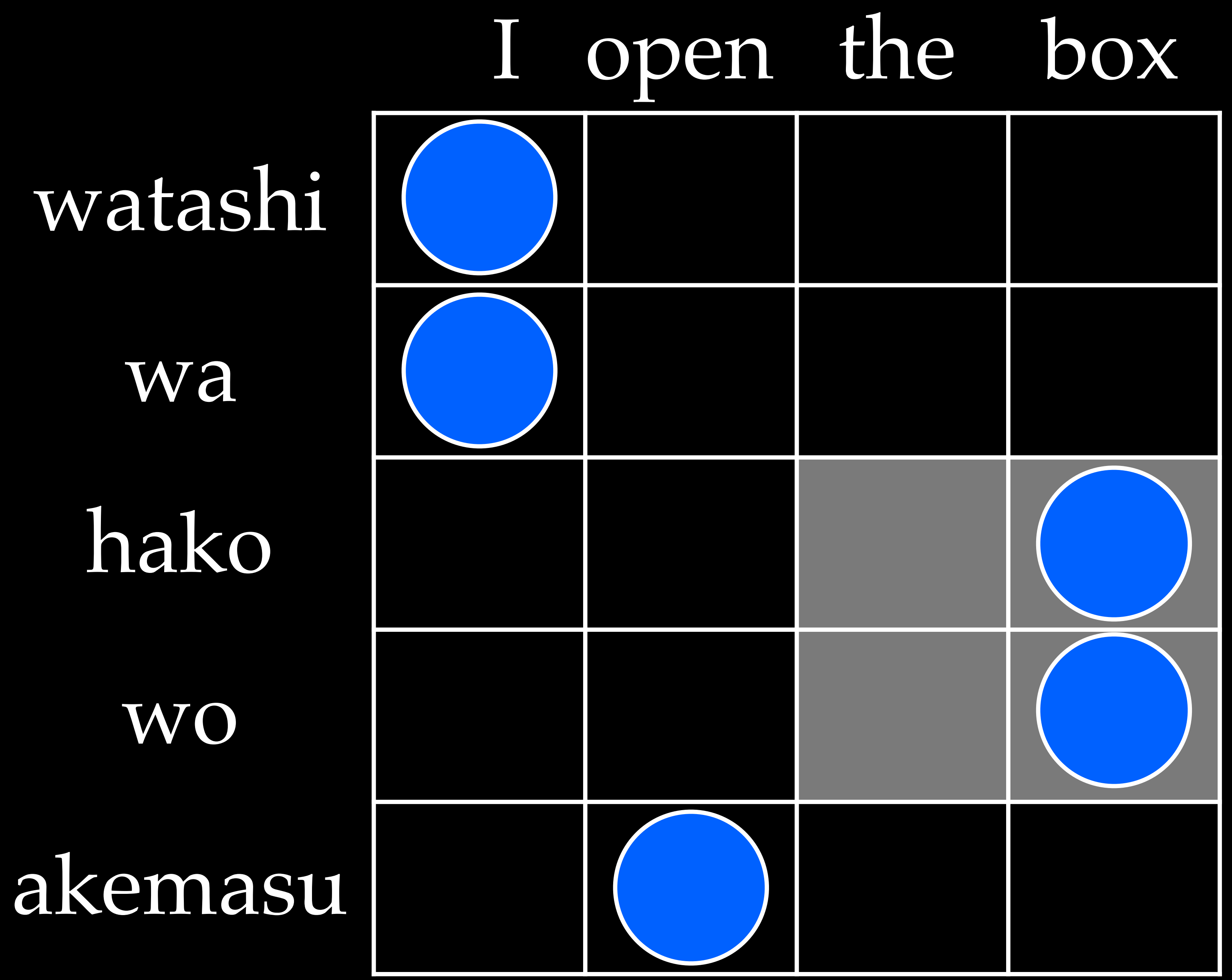
~~watashi / I~~

Phrase Extraction



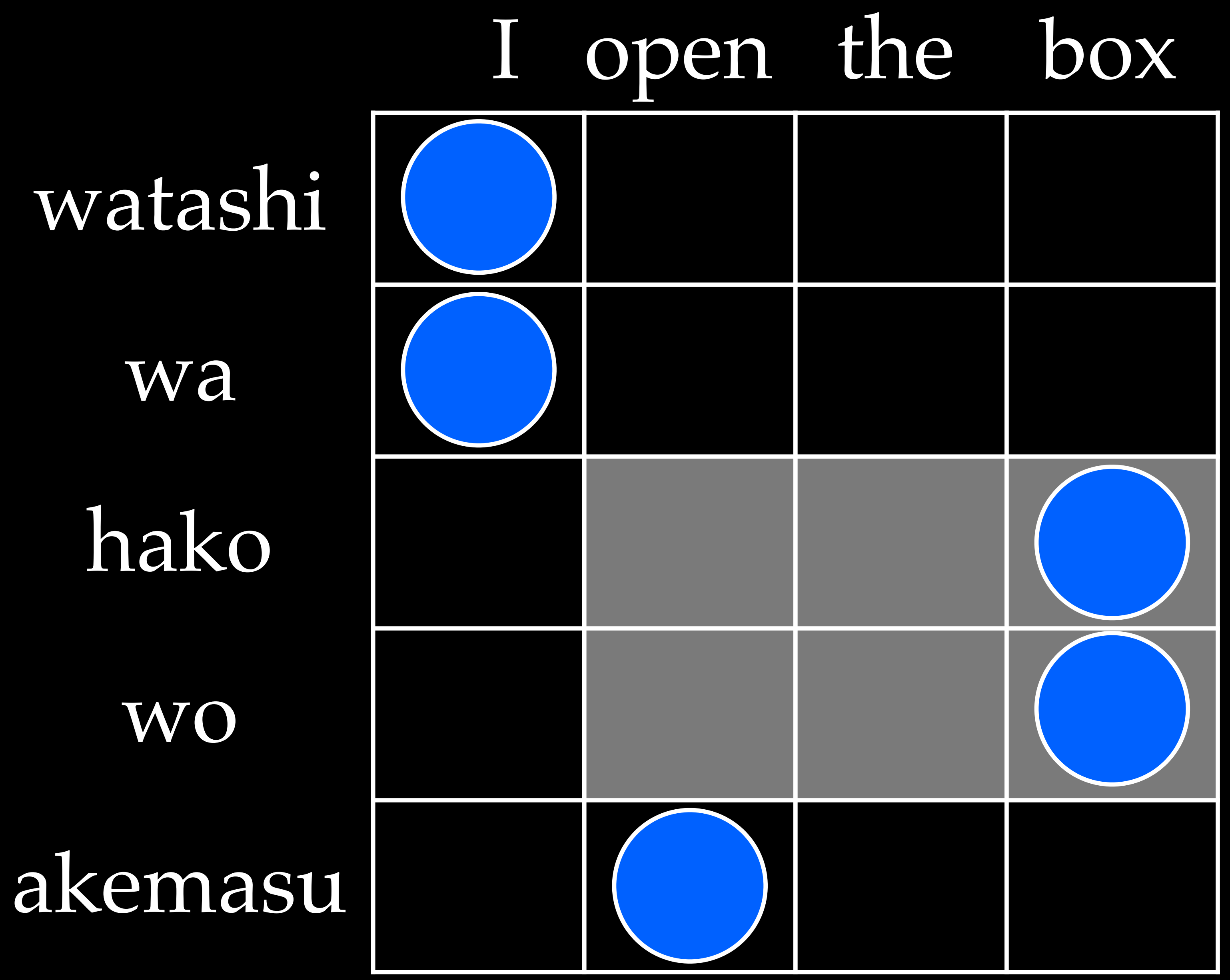
hako wo / box

Phrase Extraction



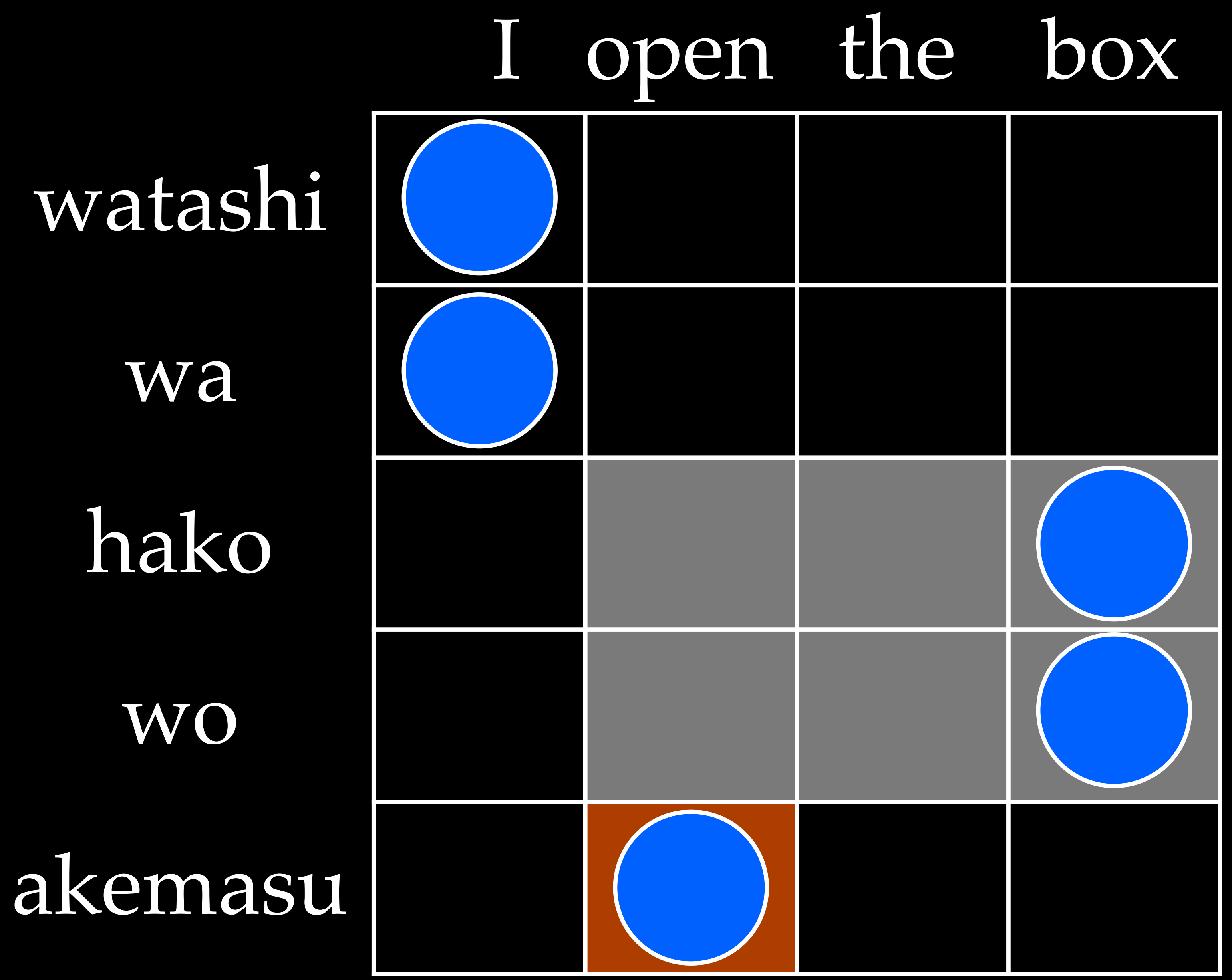
hako wo / the box

Phrase Extraction



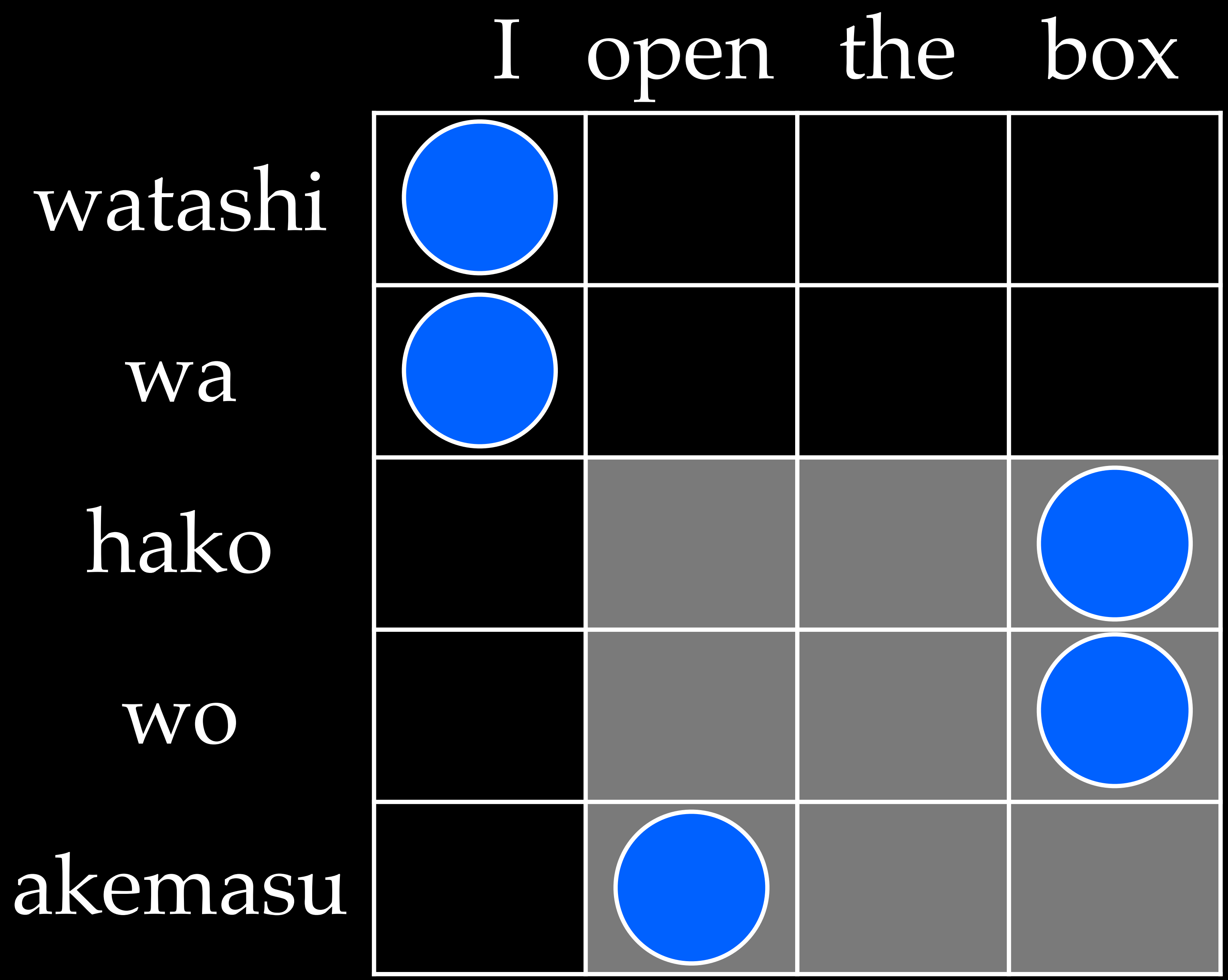
hako wo / open the box

Phrase Extraction



hako wo / open the box

Phrase Extraction



hako wo akemasu / open the box

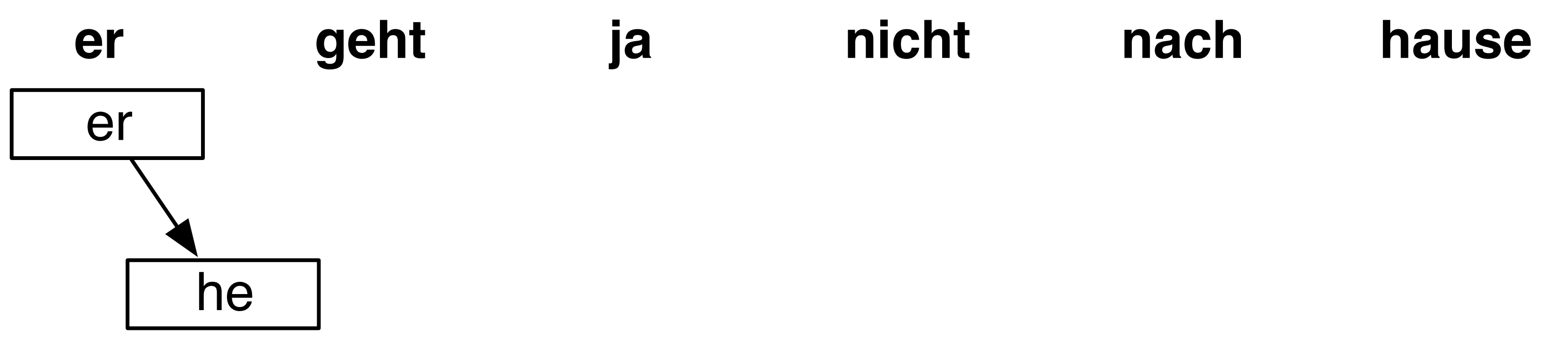
Translation Process

- Task: translate this sentence from German into English

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

Translation Process

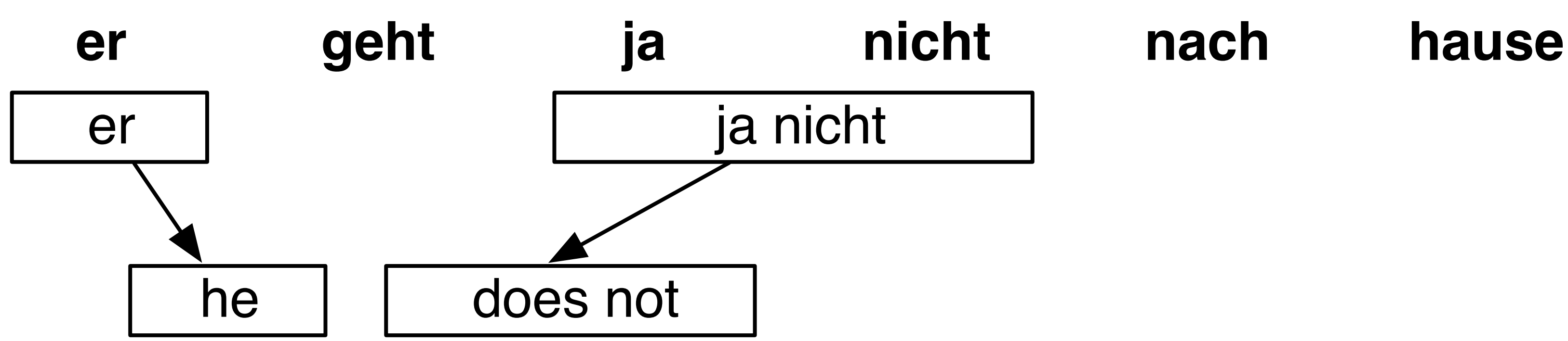
- Task: translate this sentence from German into English



- Pick phrase in input, translate

Translation Process

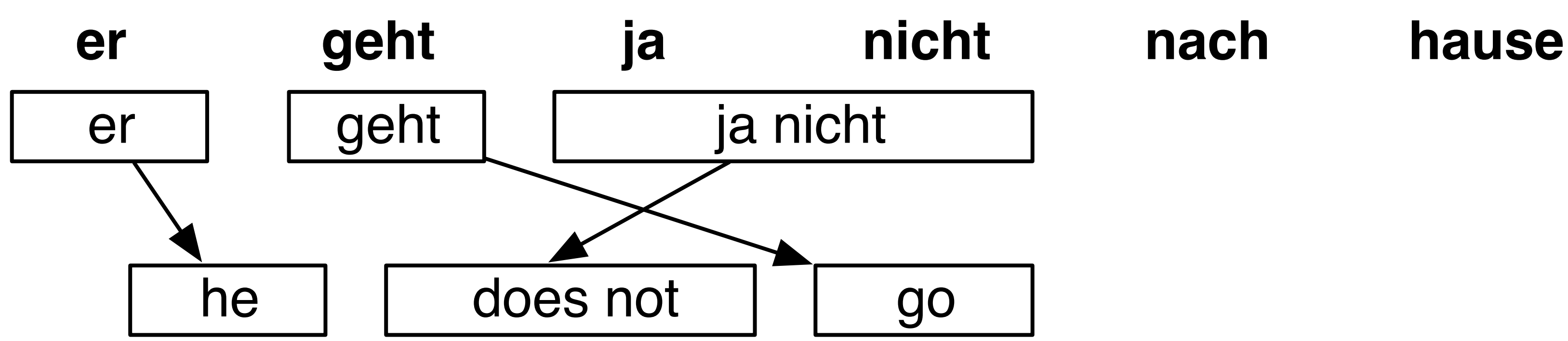
- Task: translate this sentence from German into English



- 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 Process

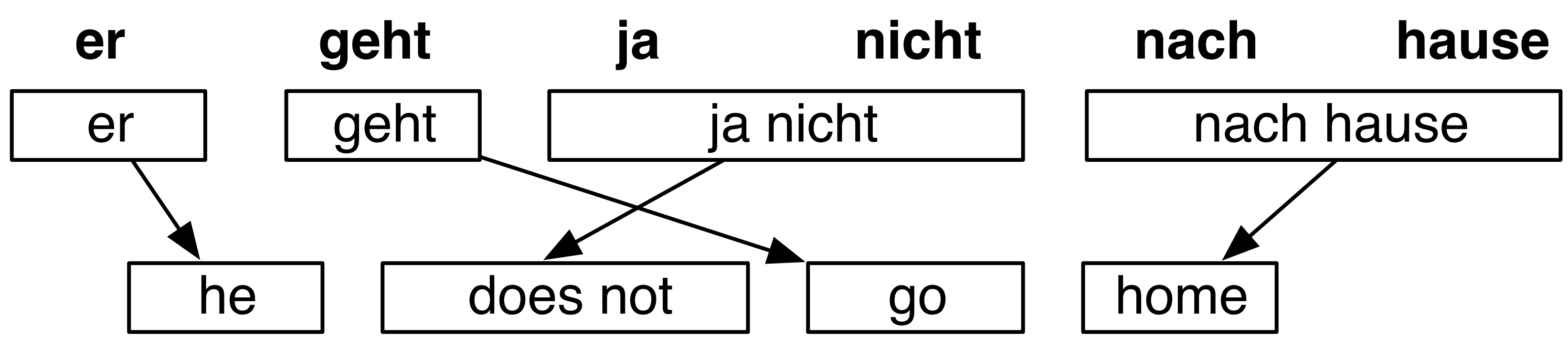
- Task: translate this sentence from German into English



- Pick phrase in input, translate

Translation Process

- Task: translate this sentence from German into English



- Pick phrase in input, translate

Computing Translation Probability

- Probabilistic model for phrase-based translation:

$$\mathbf{e}_{\text{best}} = \text{argmax}_{\mathbf{e}} \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i)\ d(start_i - end_{i-1} - 1)\ p_{\text{LM}}(\mathbf{e})$$

- Score is computed incrementally for each partial hypothesis

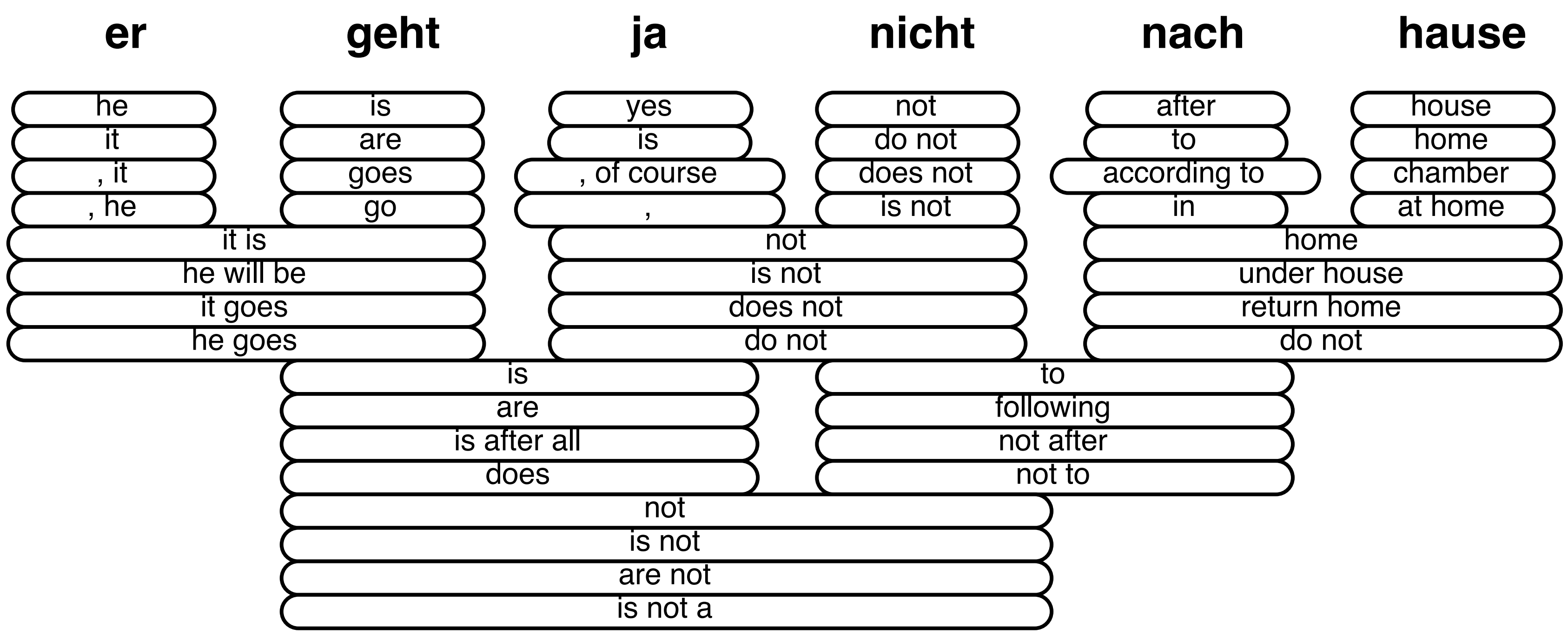
- Components

Phrase translation Picking phrase $\bar{f}_i$ to be translated as a phrase $\bar{e}_i$
→ look up score $\phi(\bar{f}_i|\bar{e}_i)$ from phrase translation table

Reordering Previous phrase ended in end_{i-1}, current phrase starts at $start_i$
→ compute $d(start_i - end_{i-1} - 1)$

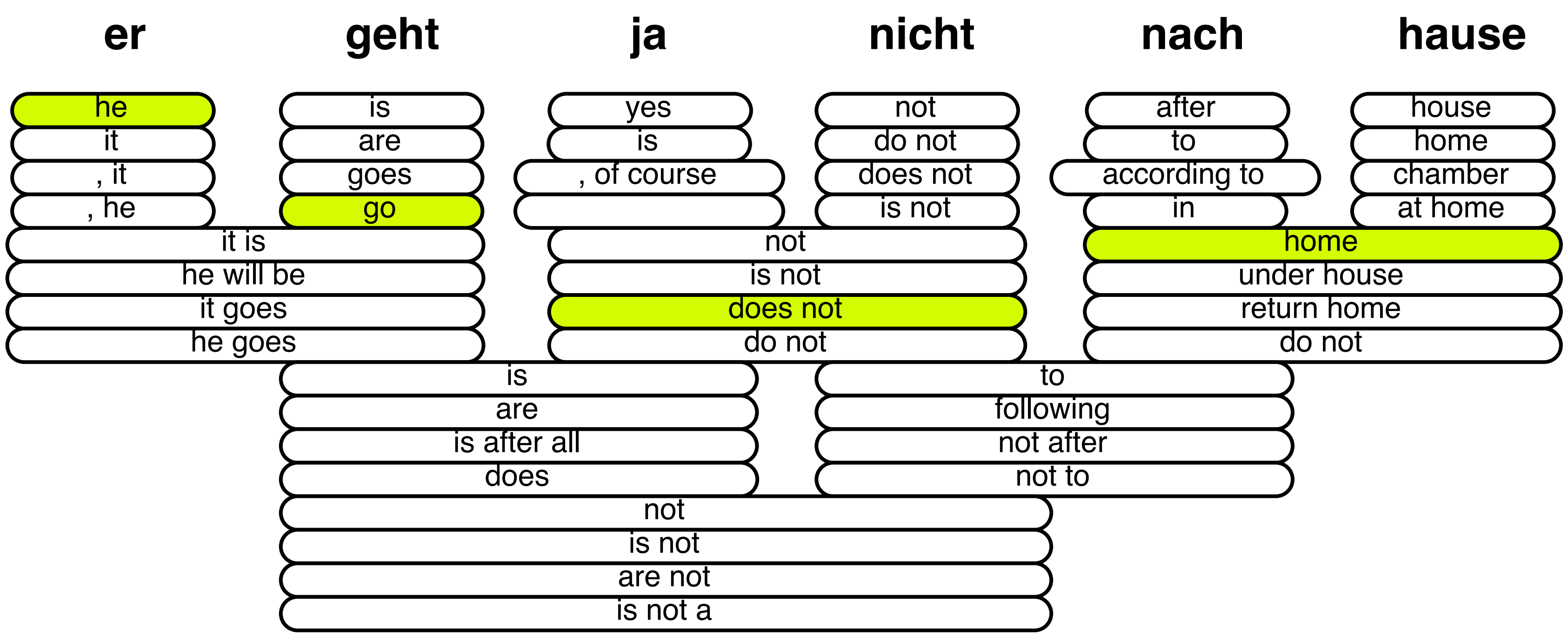
Language model For n-gram model, need to keep track of last $n-1$ words
→ compute score $p_{\text{LM}}(w_i|w_{i-(n-1)}, ..., w_{i-1})$ for added words w_i

Translation Options



- Many translation options to choose from
  - in Europarl phrase table: 2727 matching phrase pairs for this sentence
  - by pruning to the top 20 per phrase, 202 translation options remain

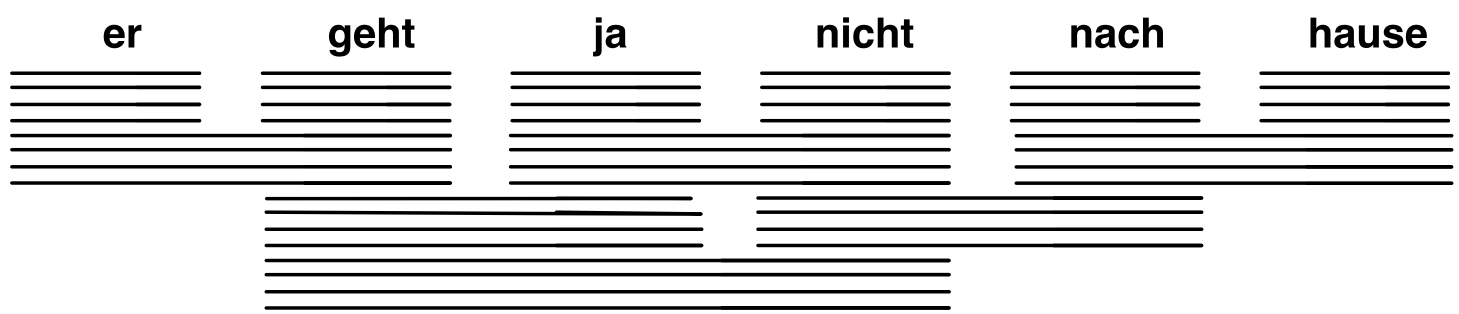
Translation Options



- The machine translation decoder does not know the right answer
  - picking the right translation options
  - arranging them in the right order

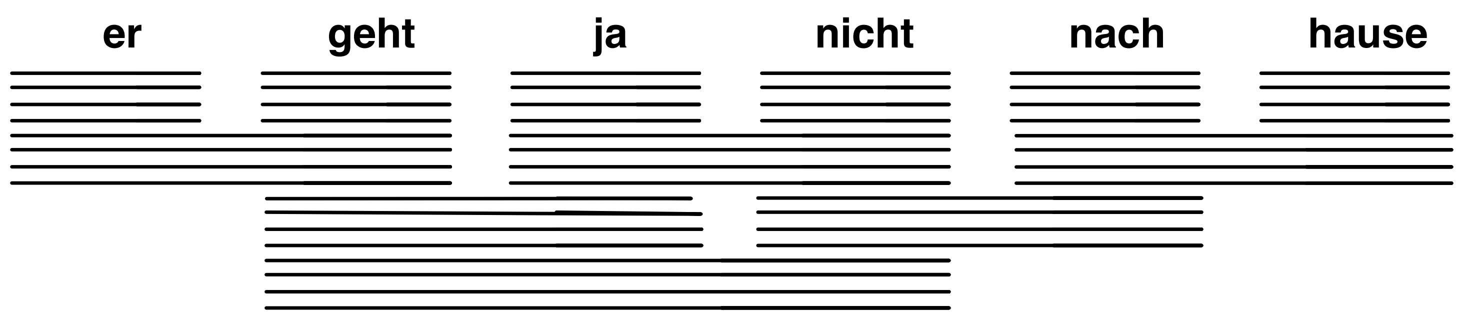
$\longrightarrow$ Search problem solved by heuristic beam search

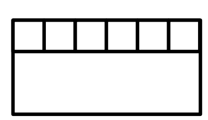
Decoding: Precompute Translation Options



consult phrase translation table for all input phrases

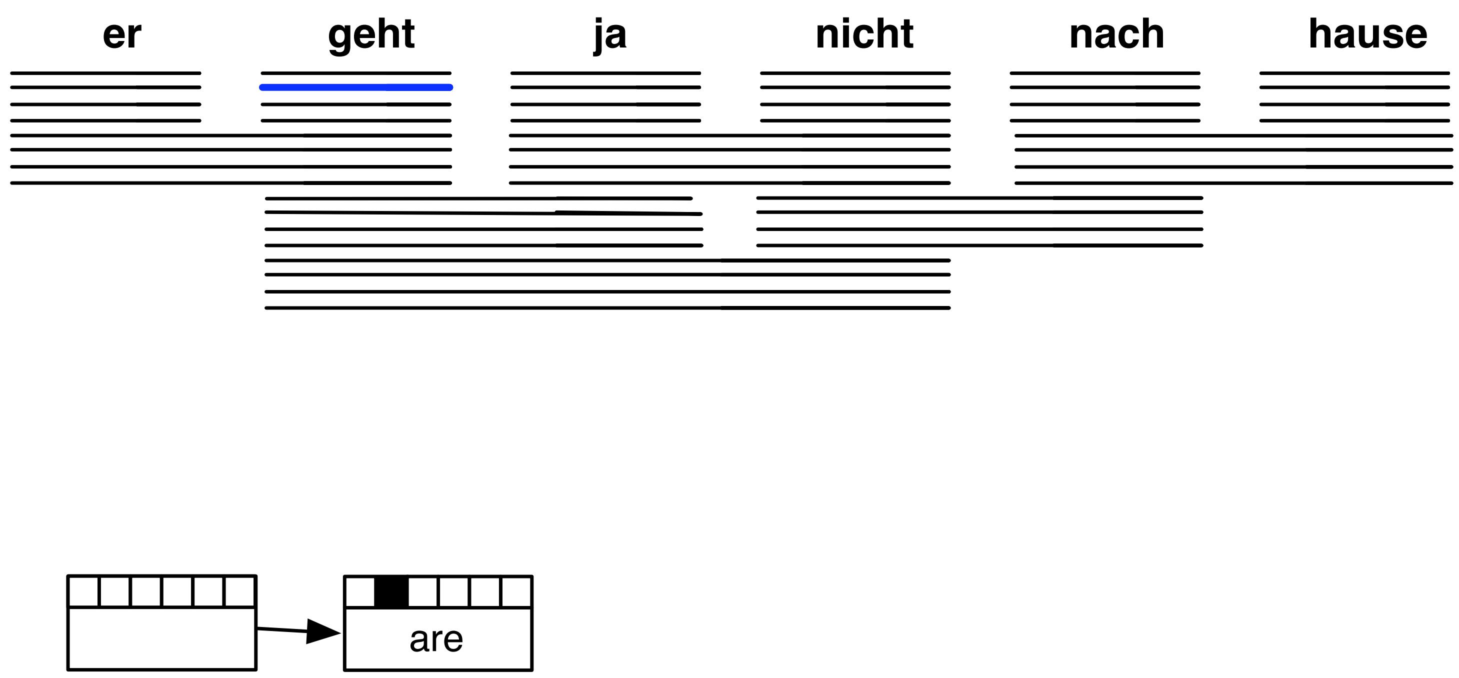
Decoding: Start with Initial Hypothesis





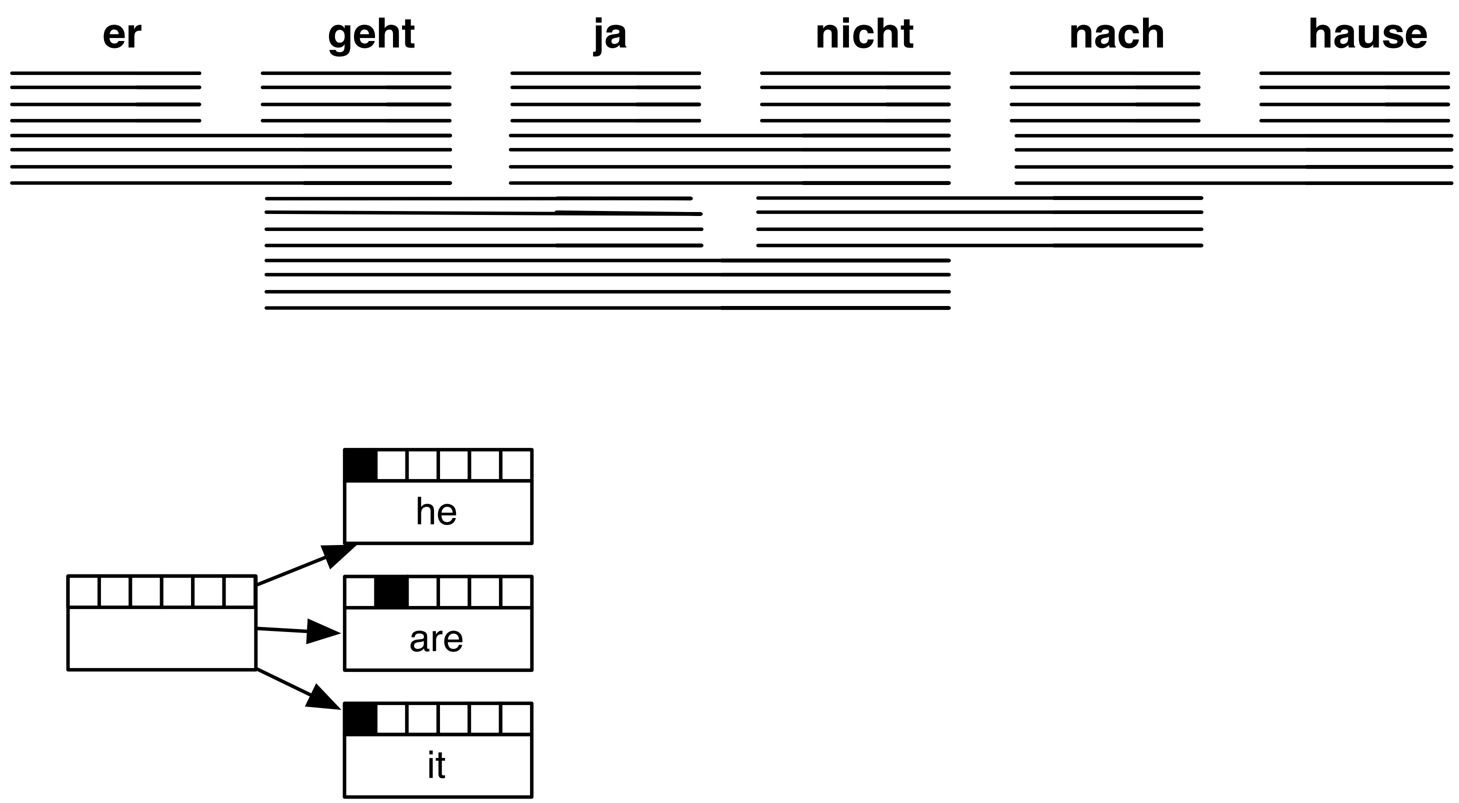
initial hypothesis: no input words covered, no output produced

Decoding: Hypothesis Expansion



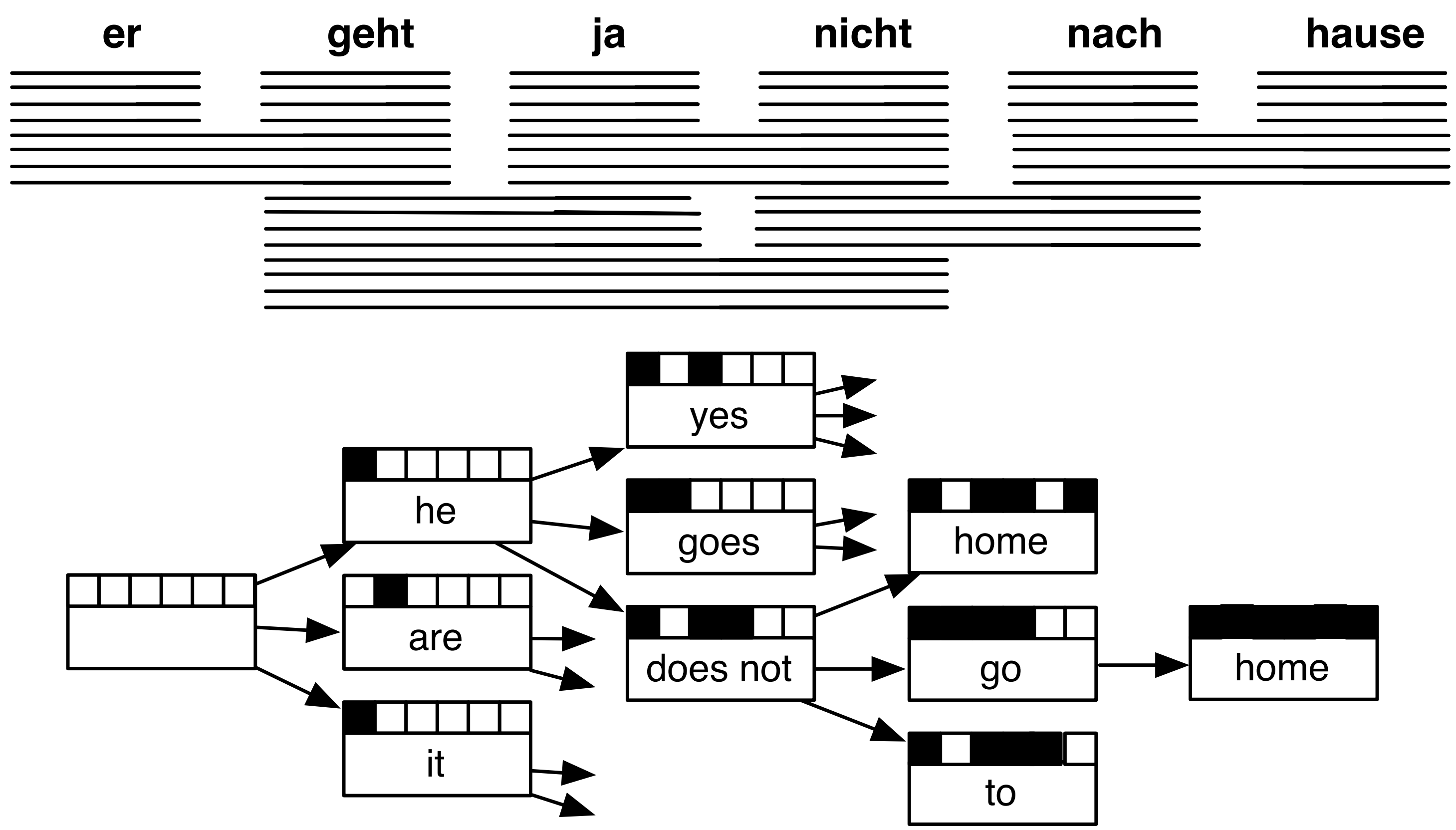
pick any translation option, create new hypothesis

Decoding: Hypothesis Expansion



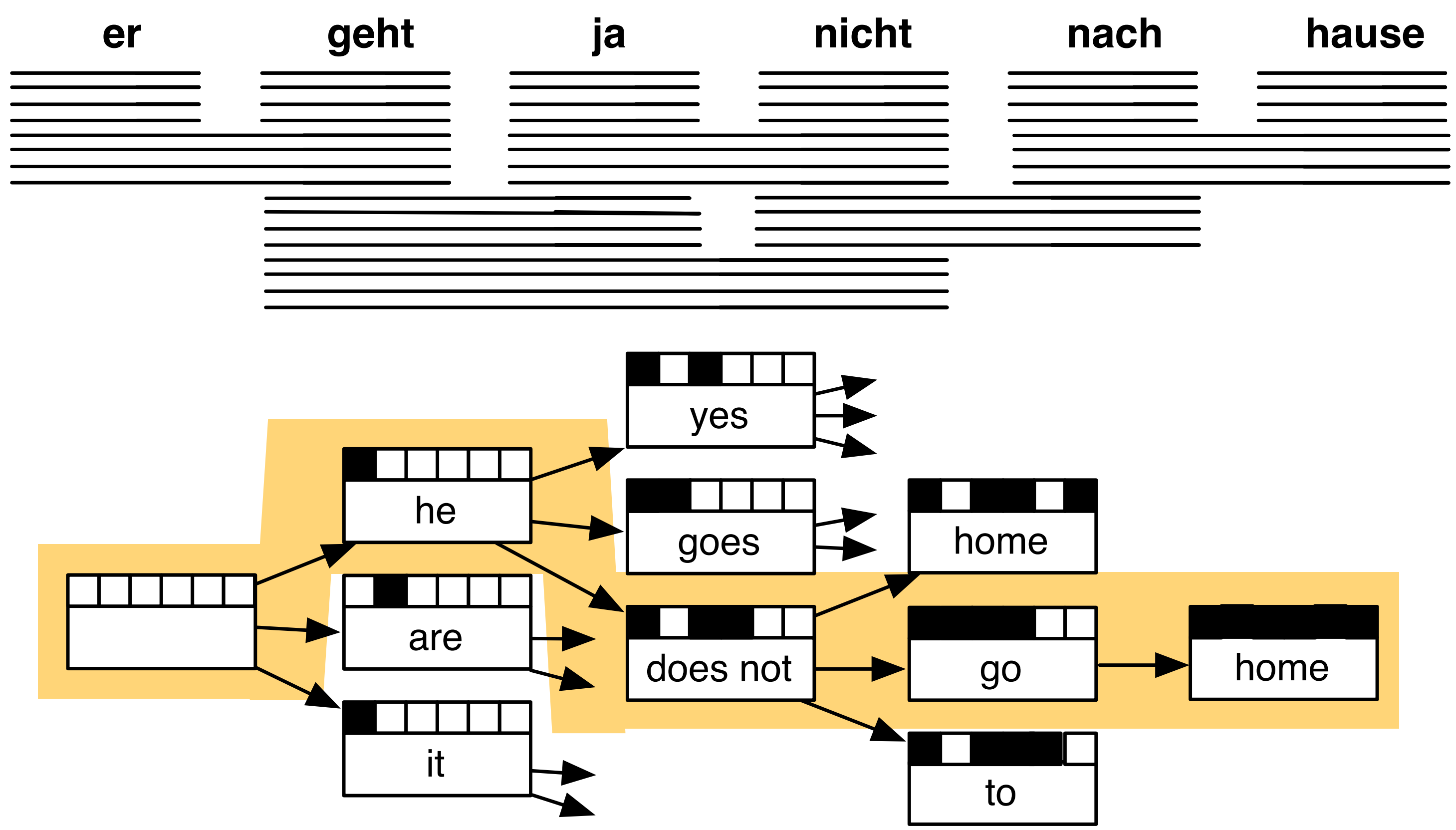
create hypotheses for all other translation options

Decoding: Hypothesis Expansion



also create hypotheses from created partial hypothesis

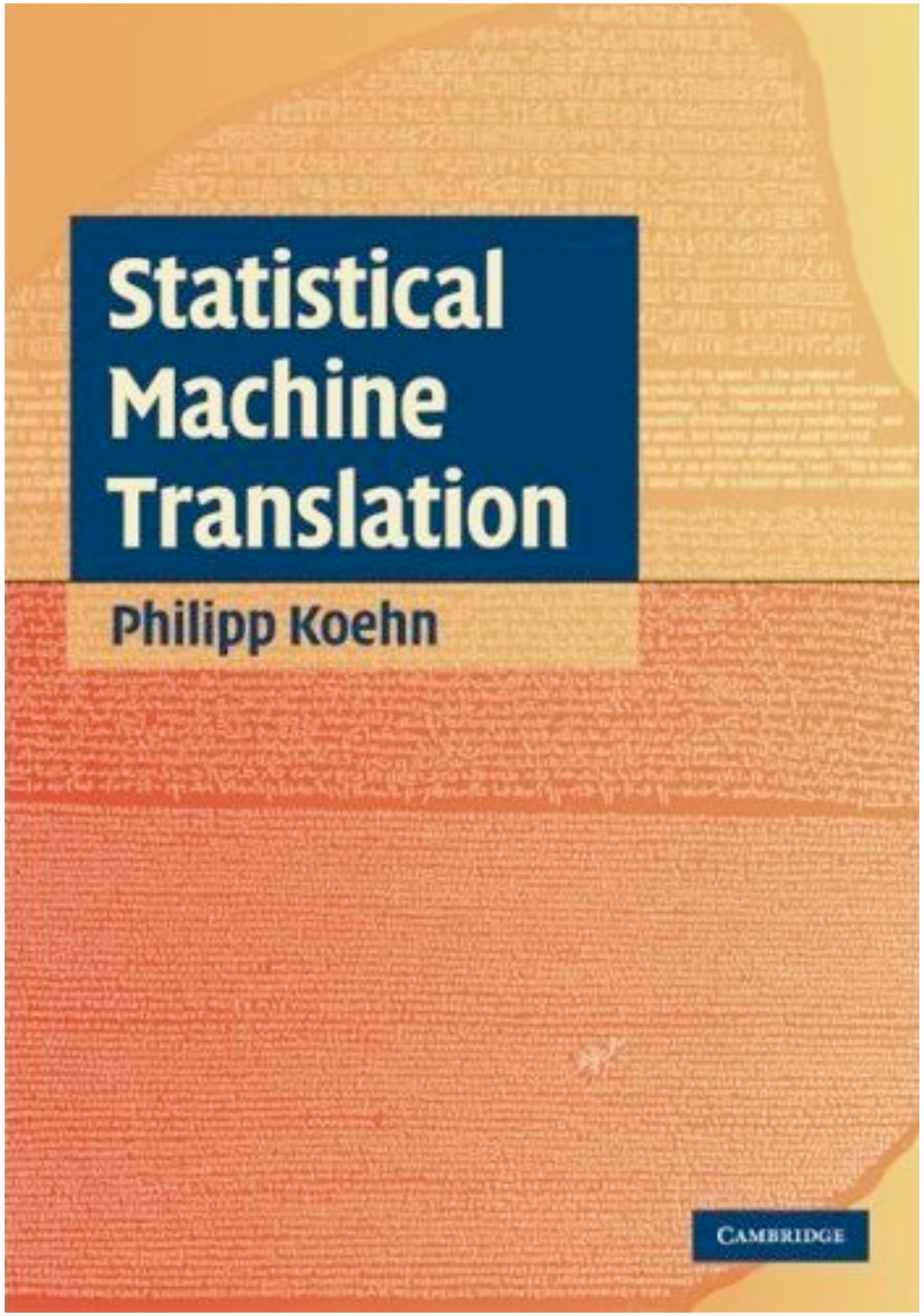
Decoding: Find Best Path



backtrack from highest scoring complete hypothesis

Reading

- Read Chapter 5 and 6 from the textbook



Announcements

- HW2 will be released soon
- HW2 due **Thursday Feb 19th** at 11:59pm