Machine Translation 2: Phrase-Based Translation

Computational Linguistics

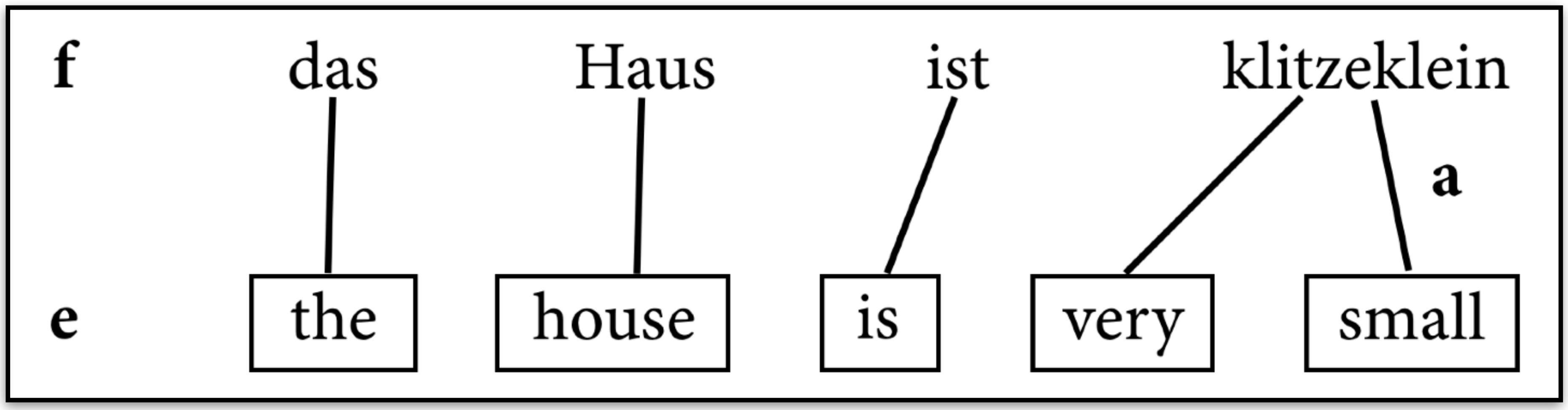
Alexander Koller

7 January 2020

slides contain material from mt-class.org

Where were we?

- Last time: Word alignments.



- Today: Actual machine translation.
  - Input: “Das Haus ist klitzeklein.”
  - Output: “The house is very small.”

Translation quality

- We can measure quality of a translation in two dimensions:
  - *Adequacy*: How accurately does translation represent the meaning of the original?
  - *Fluency:* Is the translation a good string of the target language ("good English")?

- How can we select a fluent translation?

Fluency

Israeli officials are responsible for airport security.

Israel is in charge of the security at this airport.

The security work for this airport is the responsibility of the Israel government.

Israeli side was in charge of the security of this airport.

Israel is responsible for the airport's security.

Israel is responsible for safety work at this airport.

Israel presides over the security of the airport.

Israel took charge of the airport security.

The safety of this airport is taken charge of by Israel.

This airport's security is the responsibility of the Israeli security officials.

(from Koehn book)

Noisy Channel Model

- We can model fluency with a *language model* P(**e**) of the target language.
  - Can estimate from lots of monolingual data!
  - Use e.g. n-gram models (with smoothing).

- Noisy Channel Model:

$$P(\mathbf{e} \mid \mathbf{f}) = \frac{P(\mathbf{f} \mid \mathbf{e}) \cdot P(\mathbf{e})}{P(\mathbf{f})}$$

$$\propto \underbrace{P(\mathbf{f} \mid \mathbf{e})}_{\text{translation model (for adequacy)}} \cdot \underbrace{P(\mathbf{e})}_{\text{language model (for fluency)}}$$

Word-based translation model

- Could derive model for word-by-word translation, e.g. from IBM Model 1:

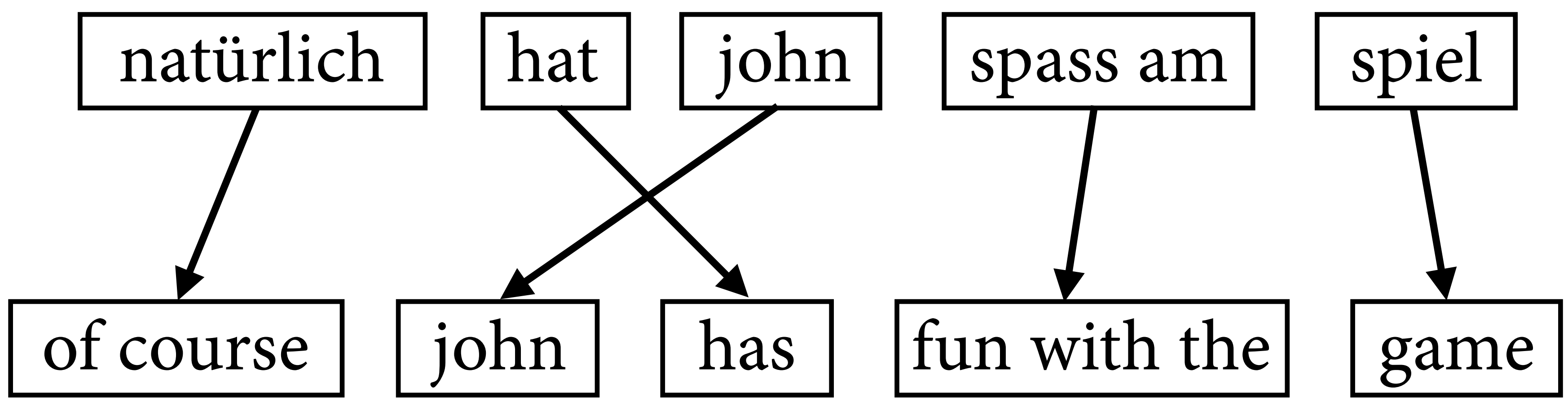
$$P(\mathbf{f} \mid \mathbf{e}) = \sum_{\mathbf{a}} P(\mathbf{f}, \mathbf{a} \mid \mathbf{e})$$

$$\propto \prod_{j=1}^{l_f} \sum_{i=1}^{l_e} P(f_j \mid e_i)$$

- (This would be a terrible translation model.)

Phrase-based translation

- But want to translate entire *phrases* (i.e. substrings):
  - translation of one word can consist of multiple words
  - context of word in phrase can help disambiguate



- Note: these "phrases" need not be linguistically meaningful constituents.

Phrase-based translation model

number of phrases

$$P(\mathbf{f} \mid \mathbf{e}) = \prod_{i=1}^{I} \phi(\bar{f}_i \mid \bar{e}_i) \cdot d(\text{start}_i - \text{end}_{i-1} - 1)$$

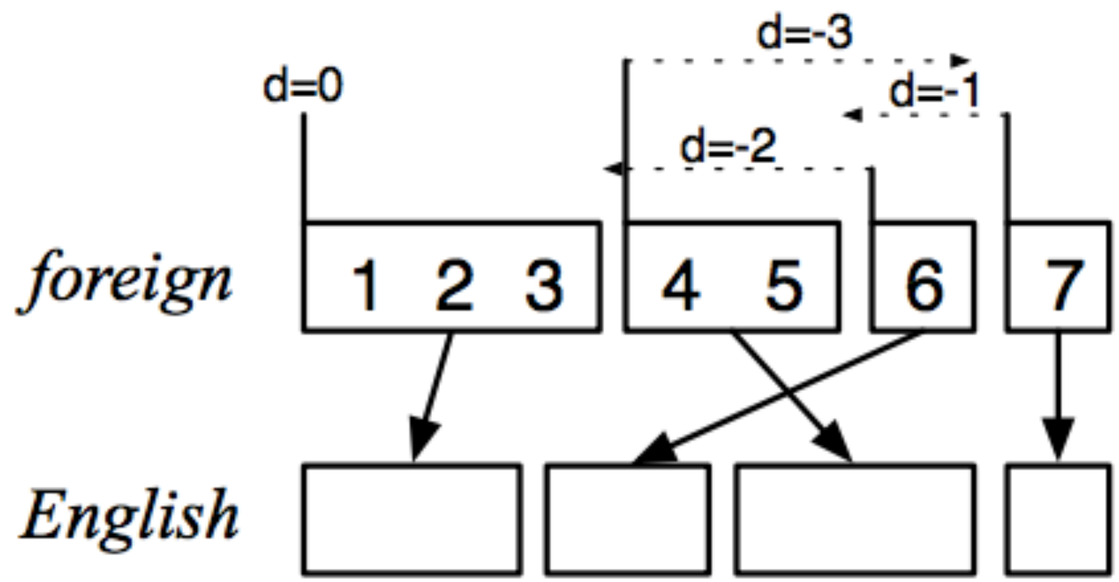
phrase translation probability

distance-based reordering model

(the whole thing gets multiplied by P(**e**) later)

Reordering Model

Let's assume a simple model for reordering for now.



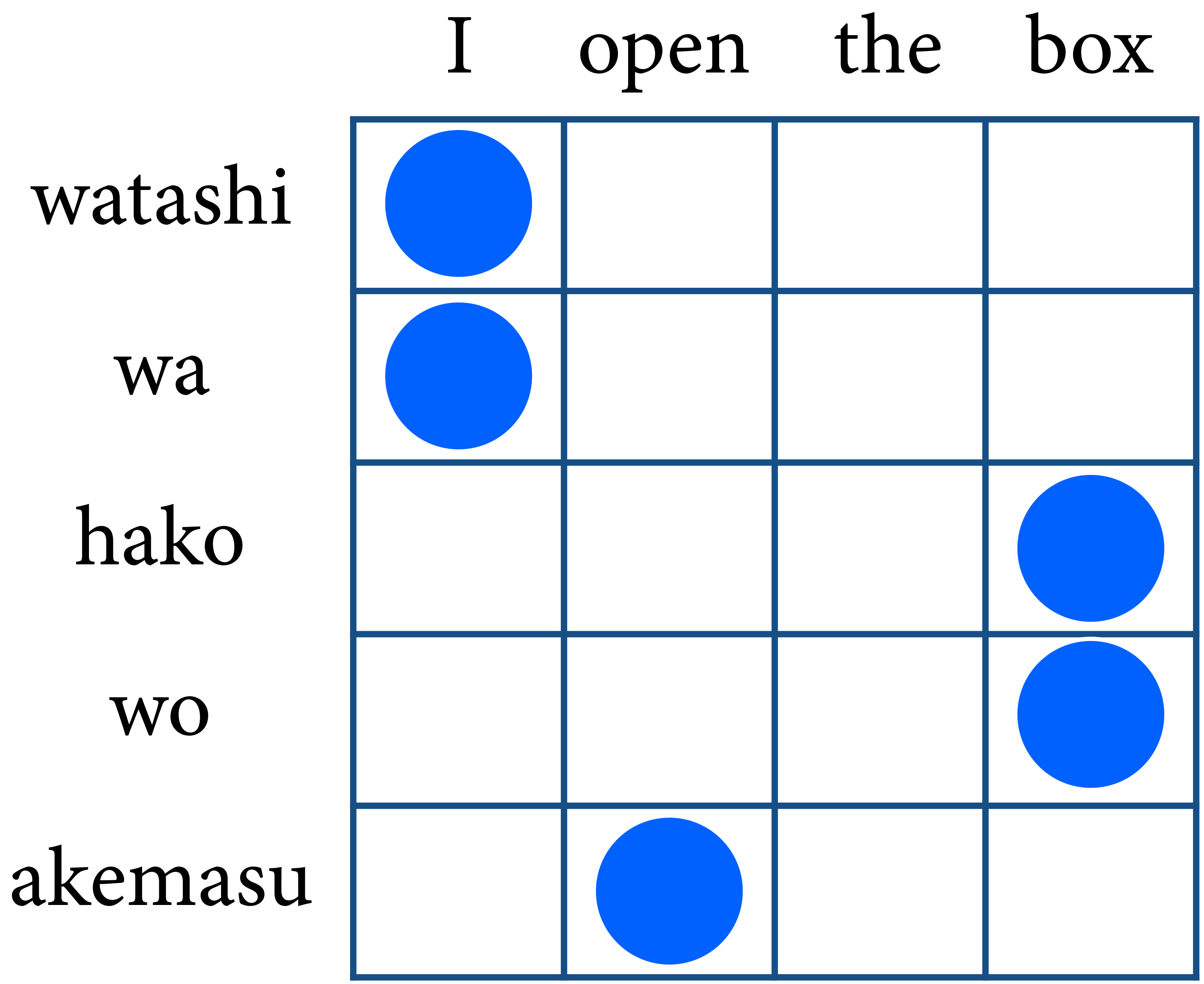
phrase	translates	movement	distance
1	1–3	start at beginning	0
2	6	skip over 4–5	+2
3	4–5	move back over 4–6	-3
4	7	skip over 6	+1

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

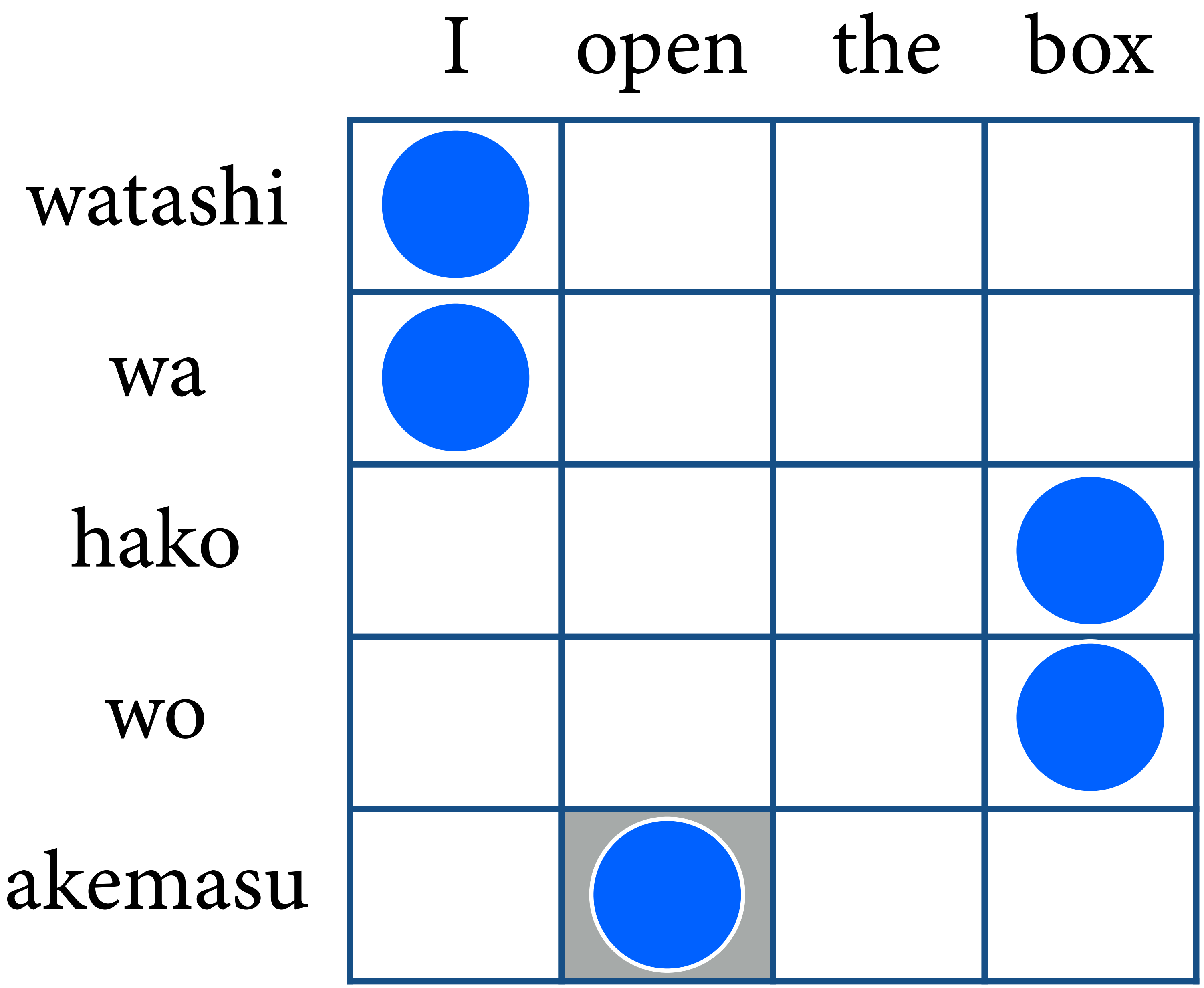
Learning phrase translations

- Extend word alignments to phrase alignments.
- Collect all phrase pairs from the parallel corpus (both big and small — we want *all* phrase pairs).
- Estimate phrase translation probabilities P(**f** | **e**) using maximum likelihood estimation (plus smoothing).

Phrase Extraction

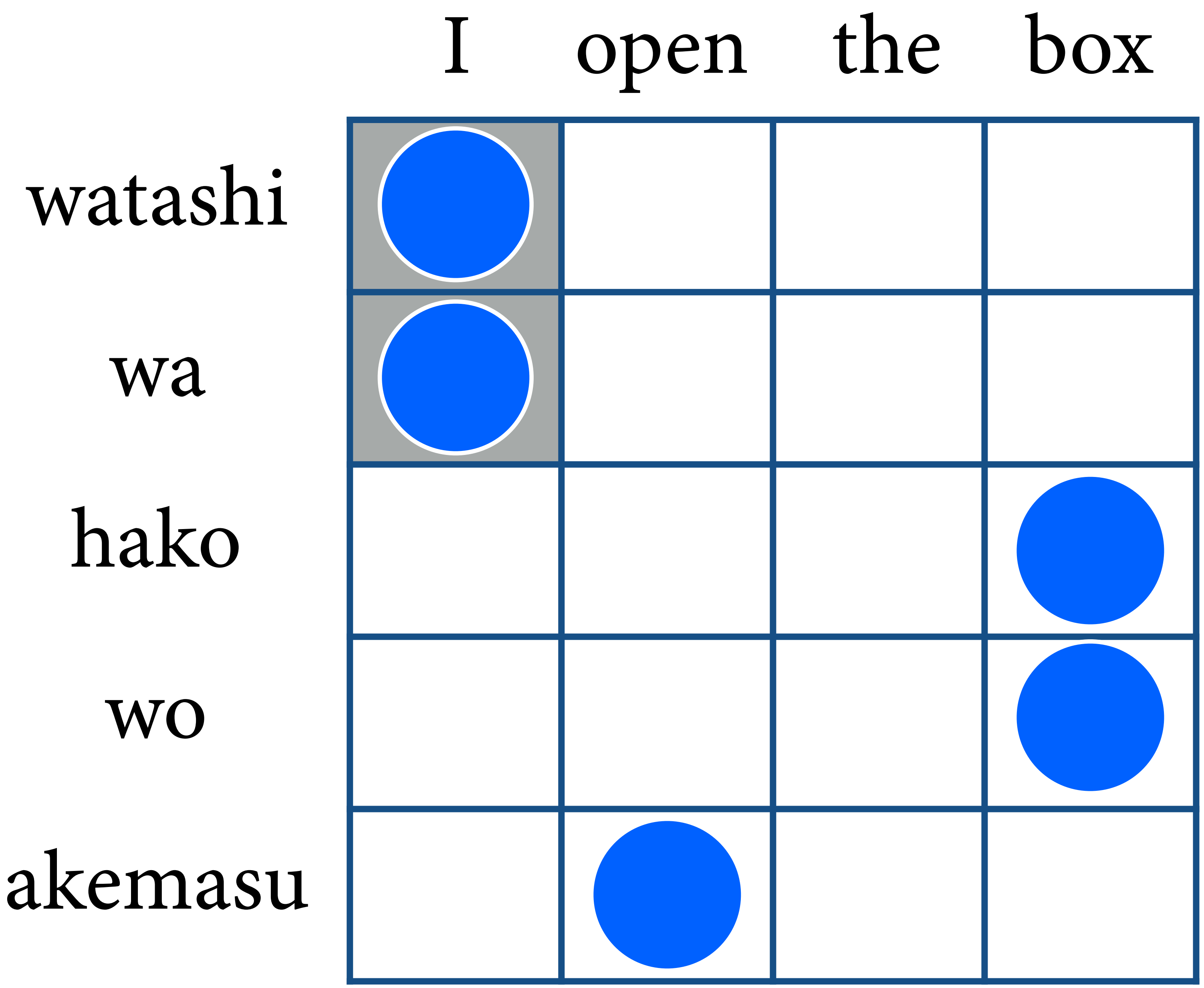


Phrase Extraction



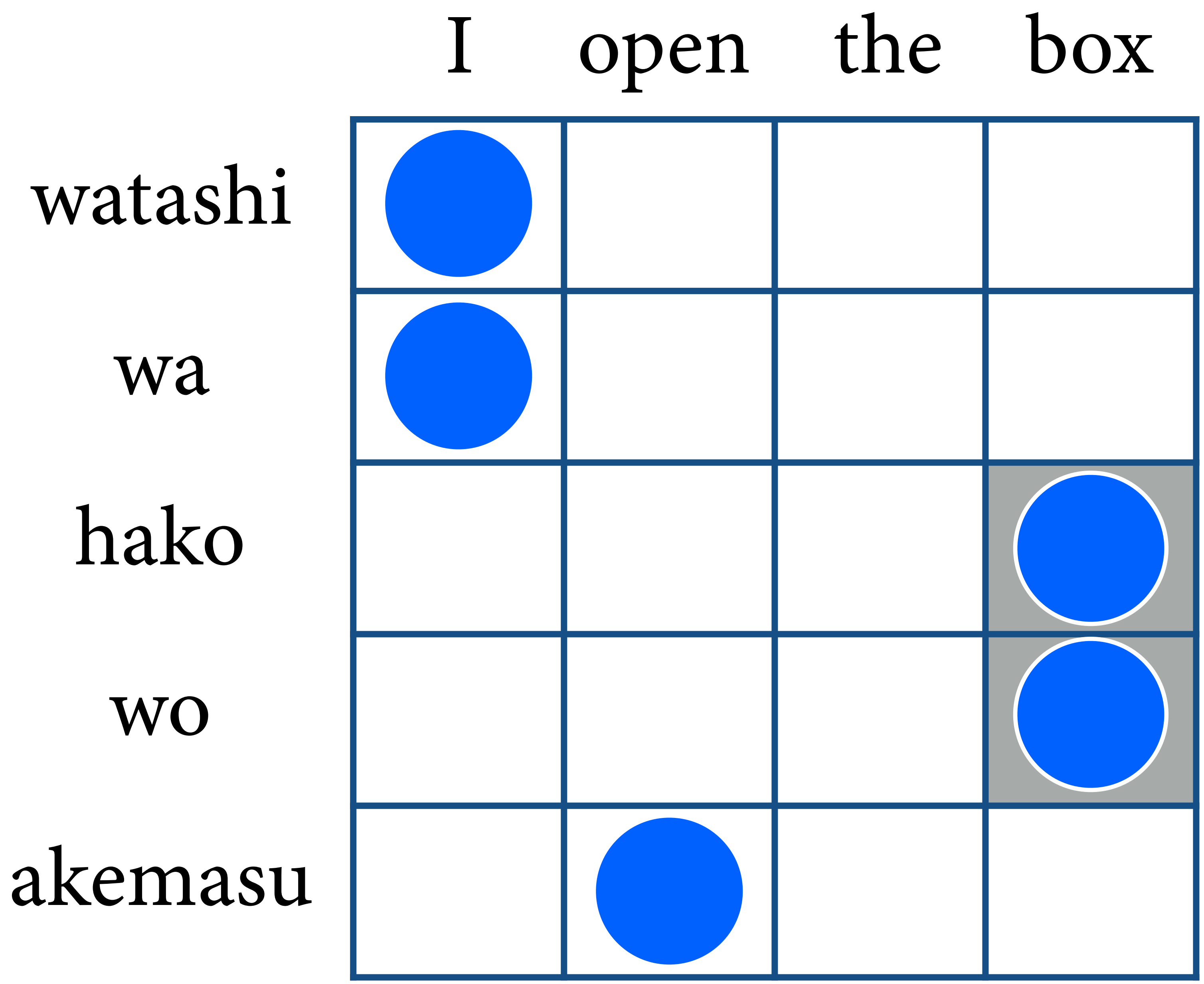
akemasu / open

Phrase Extraction



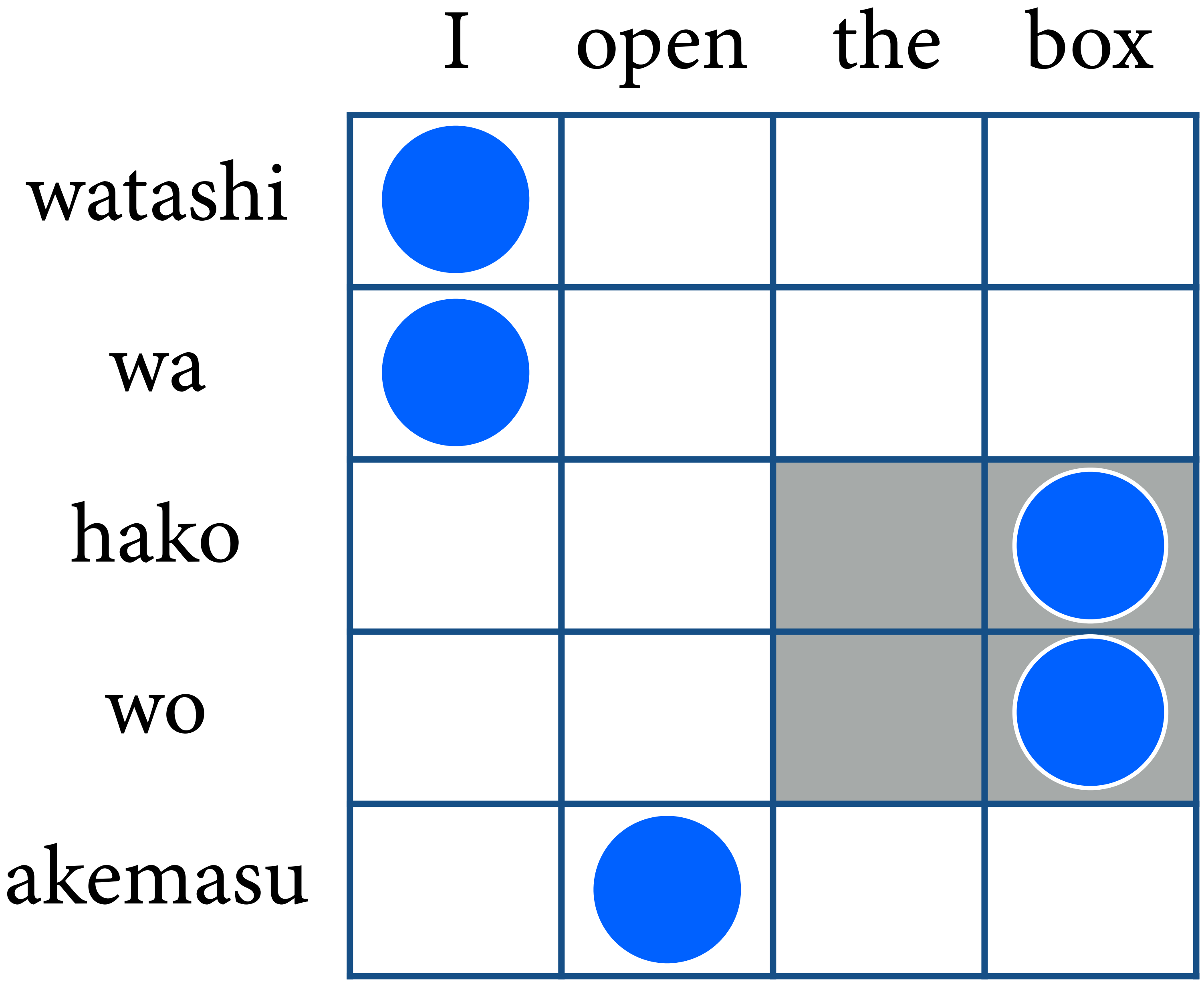
watashi wa / I

Phrase Extraction



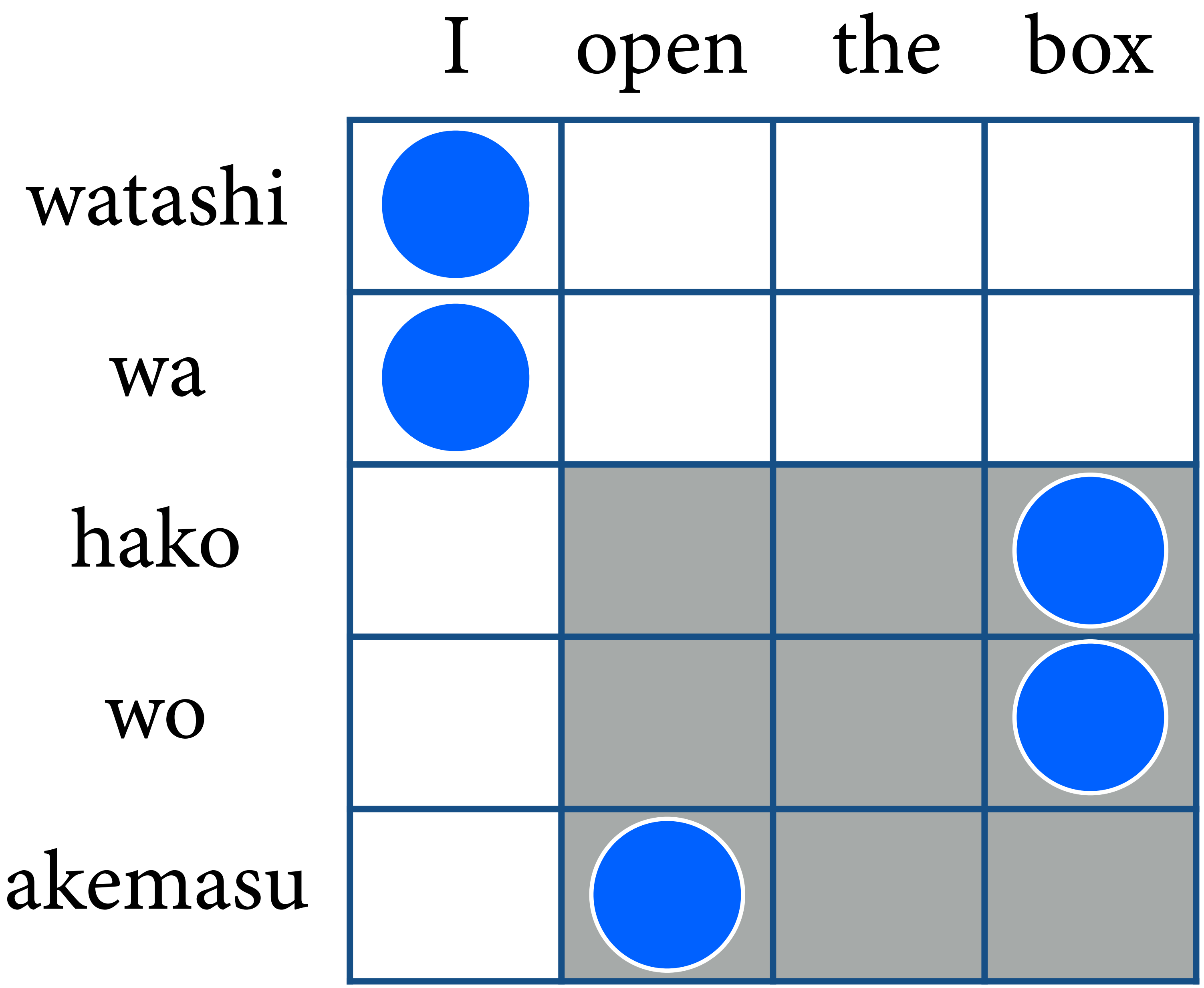
hako wo / box

Phrase Extraction



hako wo / the box

Phrase Extraction



hako wo akemasu / open the box

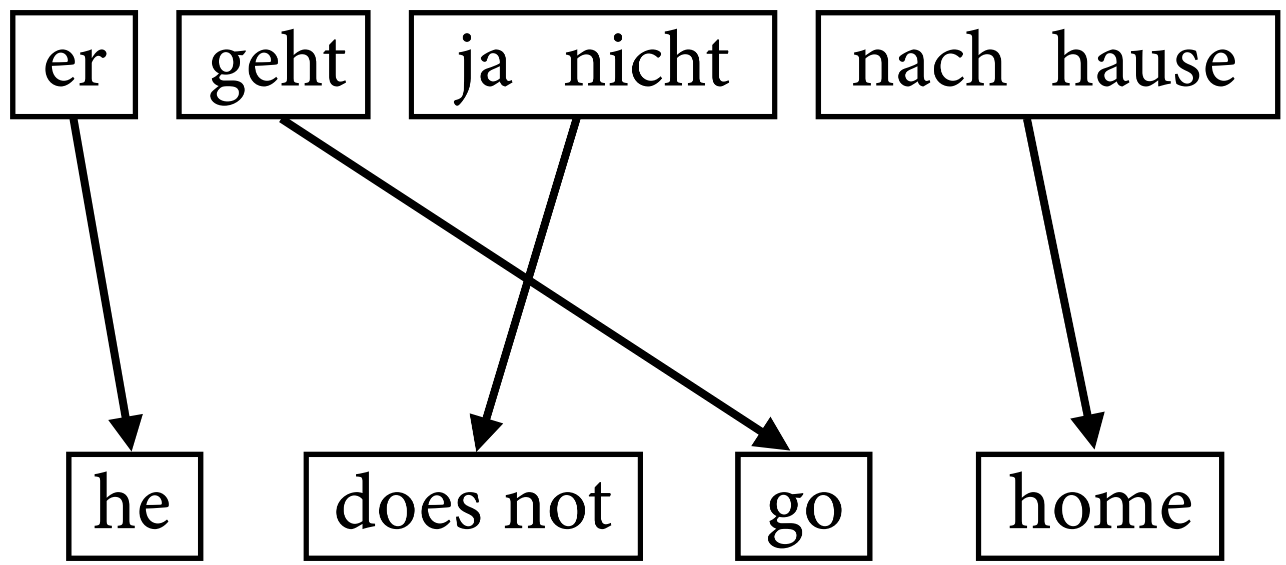
Decoding

- We now have:
  - noisy channel P(**e** | **f**) ∝ P(**f** | **e**) * P(**e**)
  - language model P(**e**)
  - phrase-based translation model

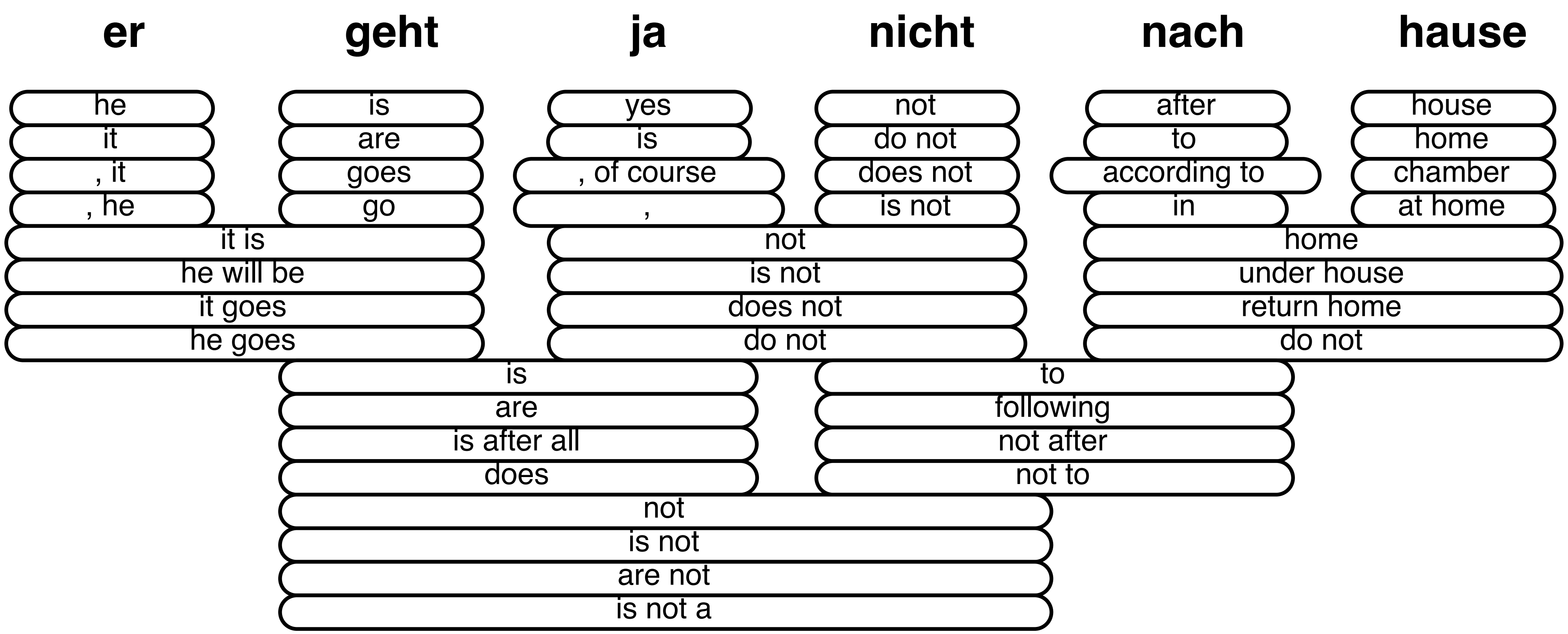
$$P(\mathbf{f} \mid \mathbf{e}) = \prod_{i=1}^{I} \phi(\bar{f}_i \mid \bar{e}_i) \cdot d(\mathrm{start}_i - \mathrm{end}_{i-1} - 1)$$

- We need to solve the *decoding* problem:
  for a given **f**, compute argmax$_{\mathbf{e}}$ P(**e** | **f**).

Basic idea



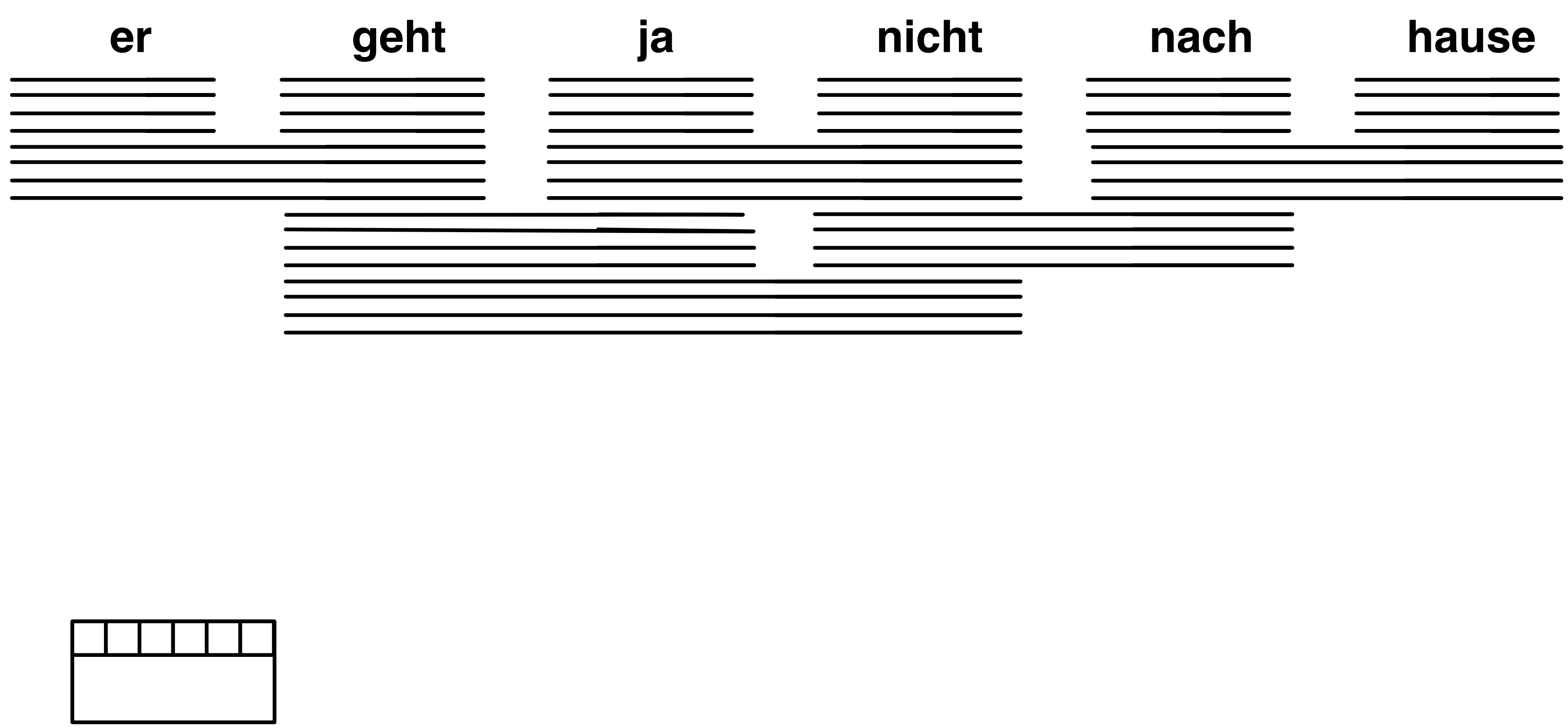
More realistically



- 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

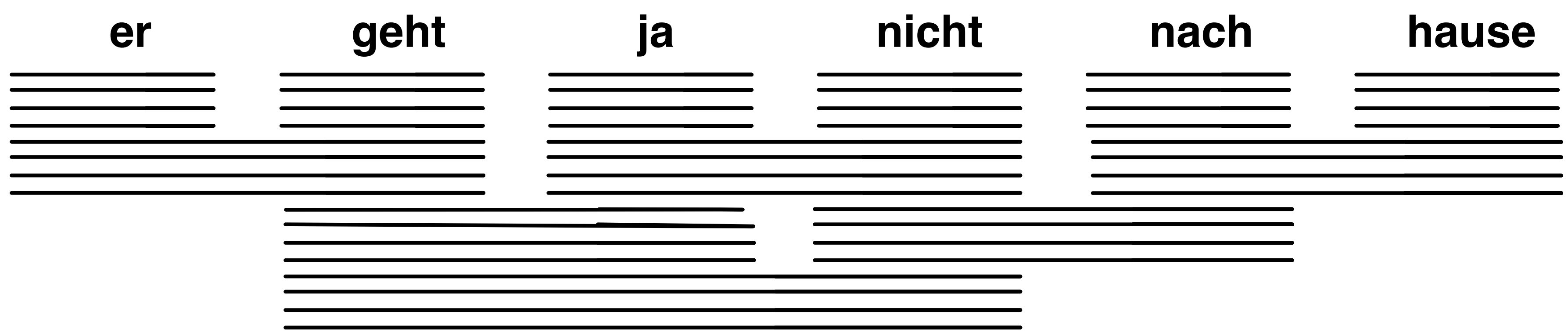
Decoding as Search

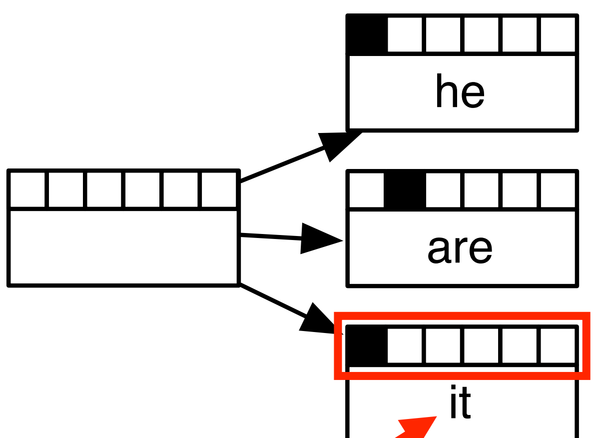
start with empty hypothesis (no words translated)



Decoding as Search

expand hypotheses by next English word





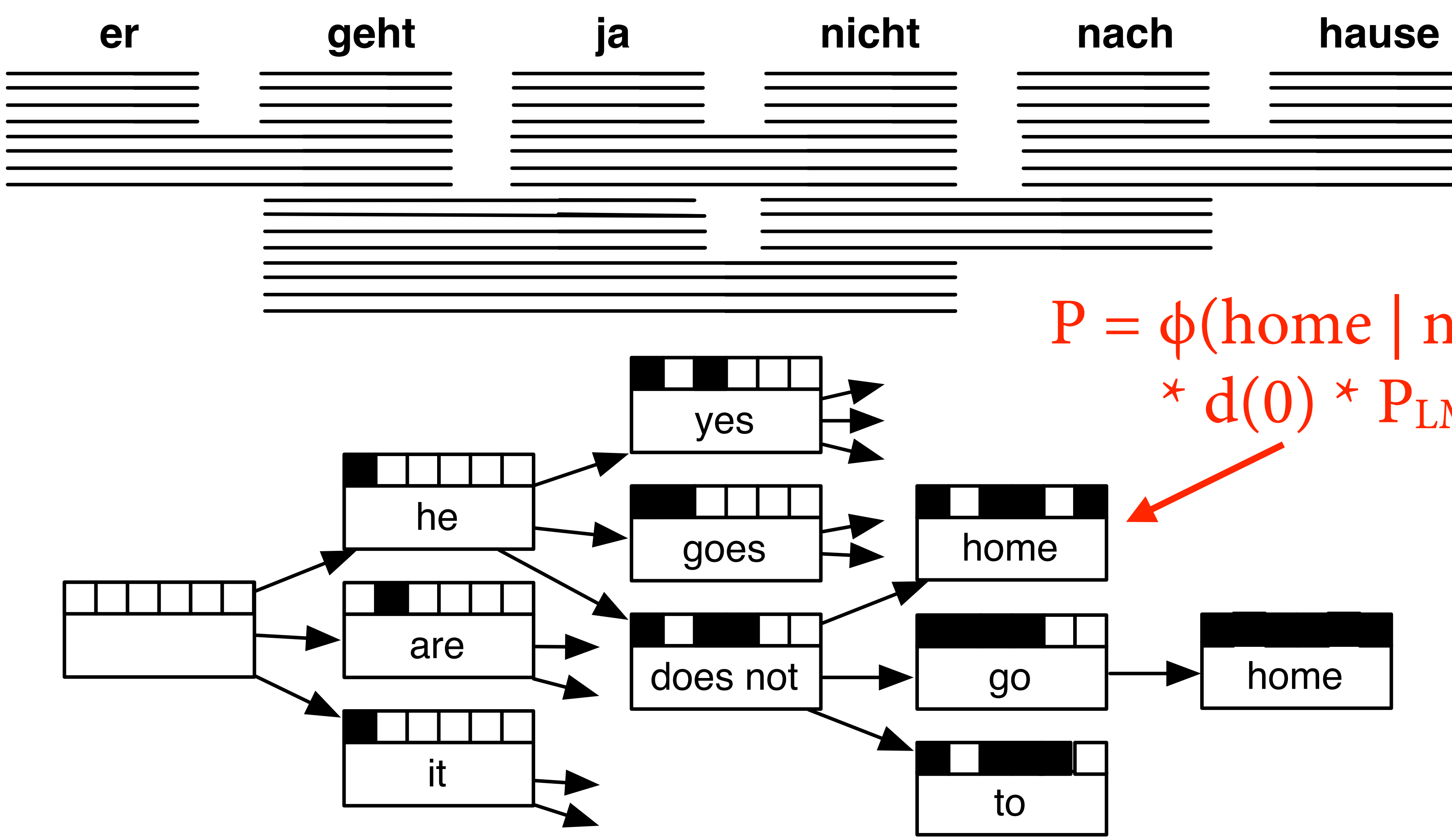
which Foreign words covered?

next English word

$P = \phi(\text{it} \mid \text{er}) * d(0) * P_{LM}(\text{it})$

Decoding as Search

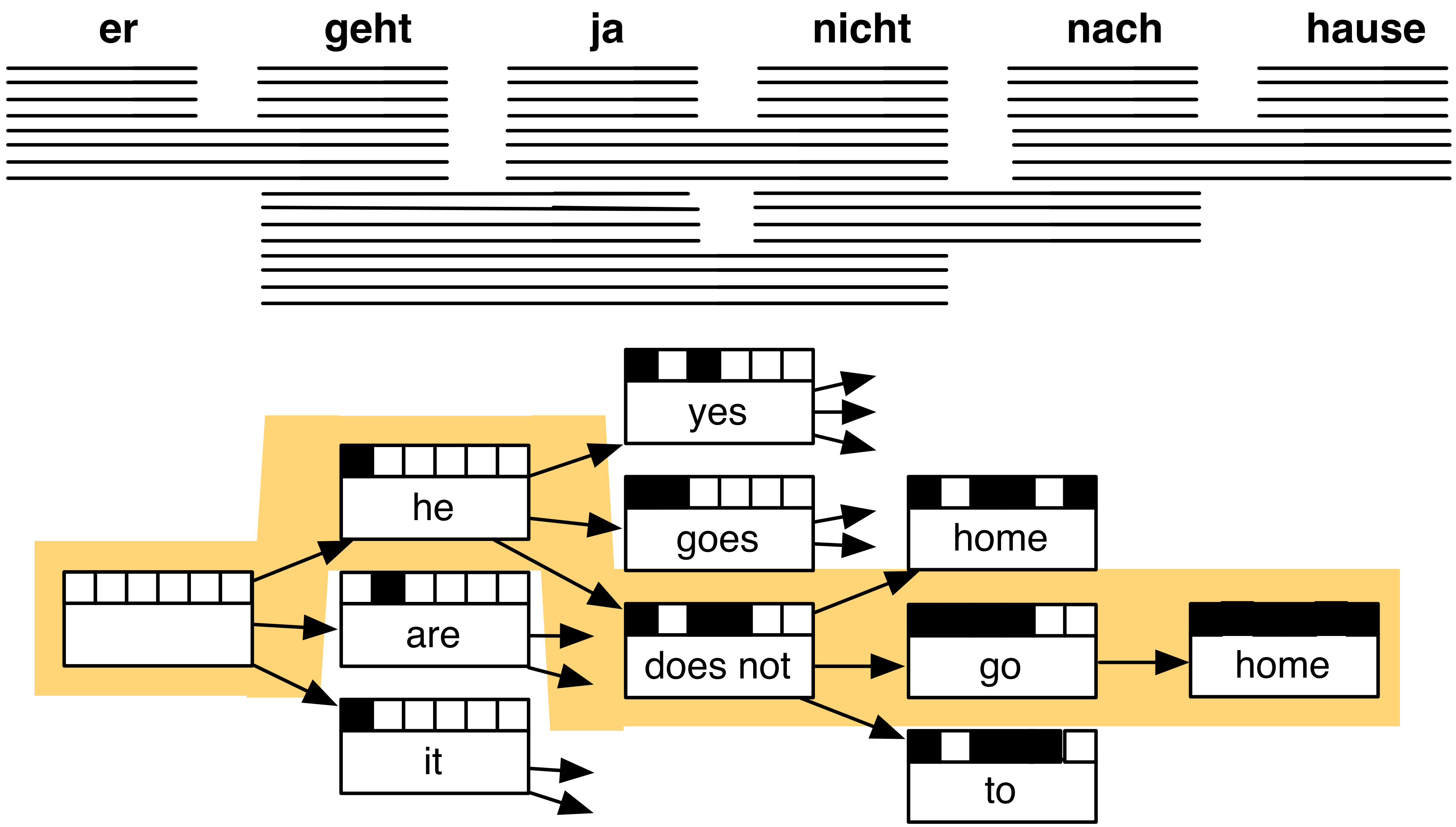
continue expanding hypotheses



$$P = \phi(\text{home} \mid \text{nach Hause}) * d(0) * P_{LM}(\text{home} \mid \text{not})$$

Decoding as Search

backtrack from highest-scoring complete hypothesis



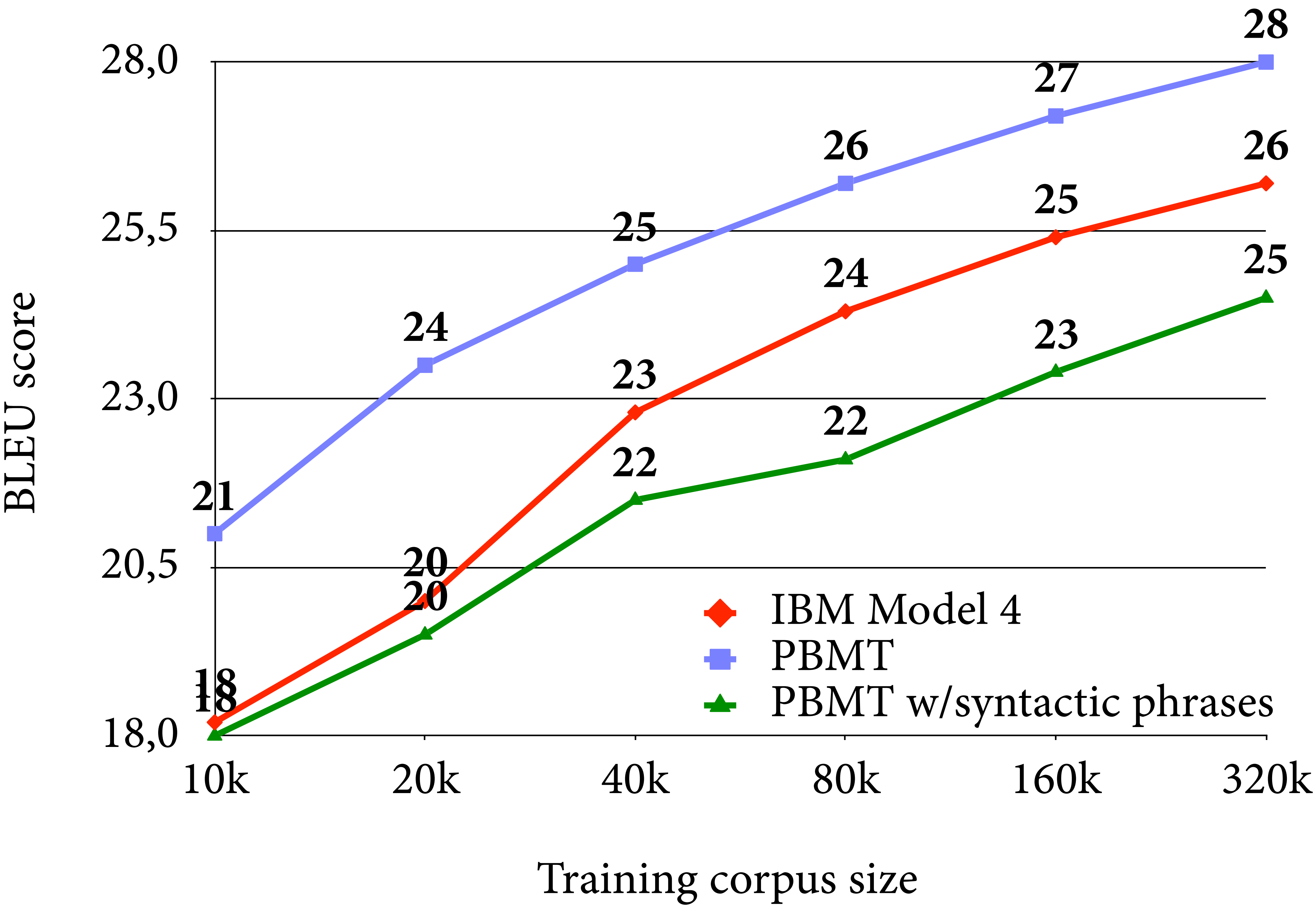
Computational issues

- Search space is huge.
  - exponential in sentence length (because of free reordering)
  - in fact, finding best translation is NP-complete
- Need heuristics to deal with complexity.
  - beam search: *stack decoding*
  - A* search

Putting linguistics in SMT

- Word-based, phrase-based SMT very naive from a linguistics perspective.
- Can we do better by putting linguistics into SMT? (At least a bit of syntax?)
- Received wisdom before 2005: phrase-based translation with lots of data much better; syntax hurts.

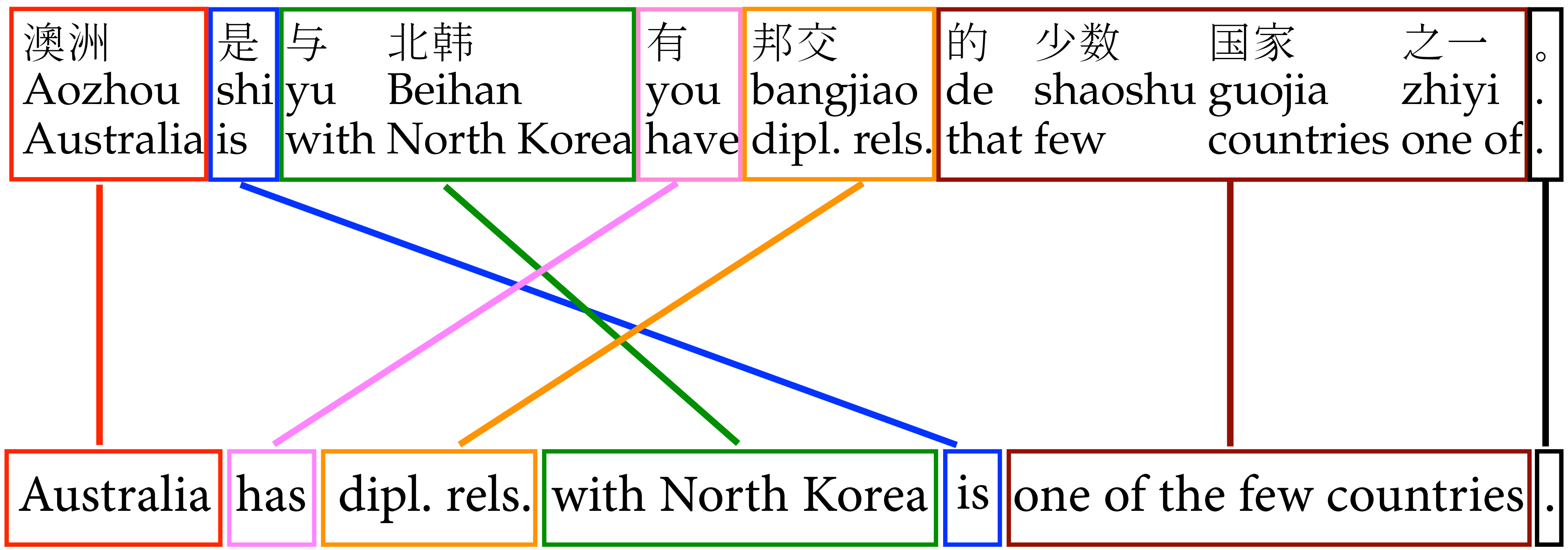
Syntax can hurt



Koehn et al. (2003)

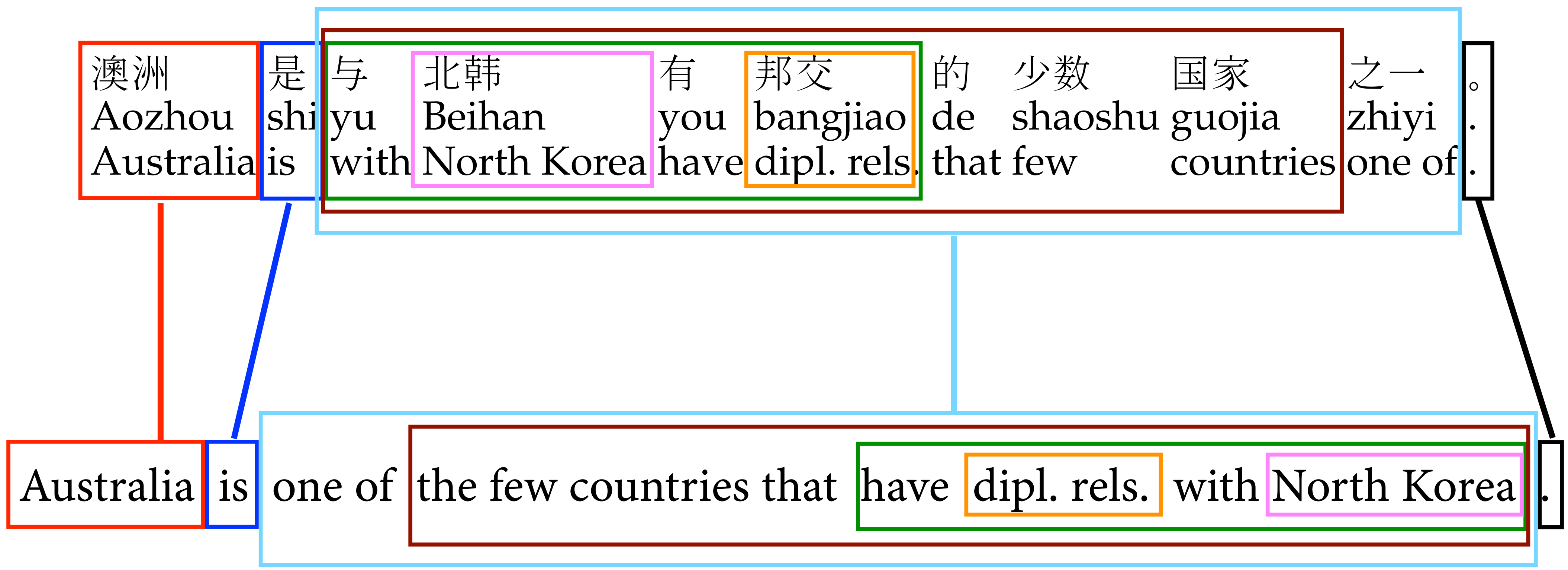
Chinese-English reordering

(output of phrase-based system ATS)



"Australia is one of the few countries that have diplomatic relations with North Korea."

Syntax-based reordering



⟨yu $\boxed{1}$ you $\boxed{2}$, have $\boxed{2}$ with $\boxed{1}$⟩

⟨$\boxed{1}$ de $\boxed{2}$, the $\boxed{2}$ that $\boxed{1}$⟩

⟨$\boxed{1}$ zhiyi, one of $\boxed{1}$⟩

"Australia is one of the few countries that have diplomatic relations with North Korea."

Syntax-based translation

- Idea: Learn *synchronous* syntax rules that capture syntactic reordering between the two languages.

- Then much less unsystematic reordering necessary.

- We need to figure out:
  - how to represent translation rules
  - how to extract translation rules from data
  - how to define probability model (skipped here)
  - how to do decoding

(Chiang 2005, 2007)

Synchronous CFG

S → X① / X①

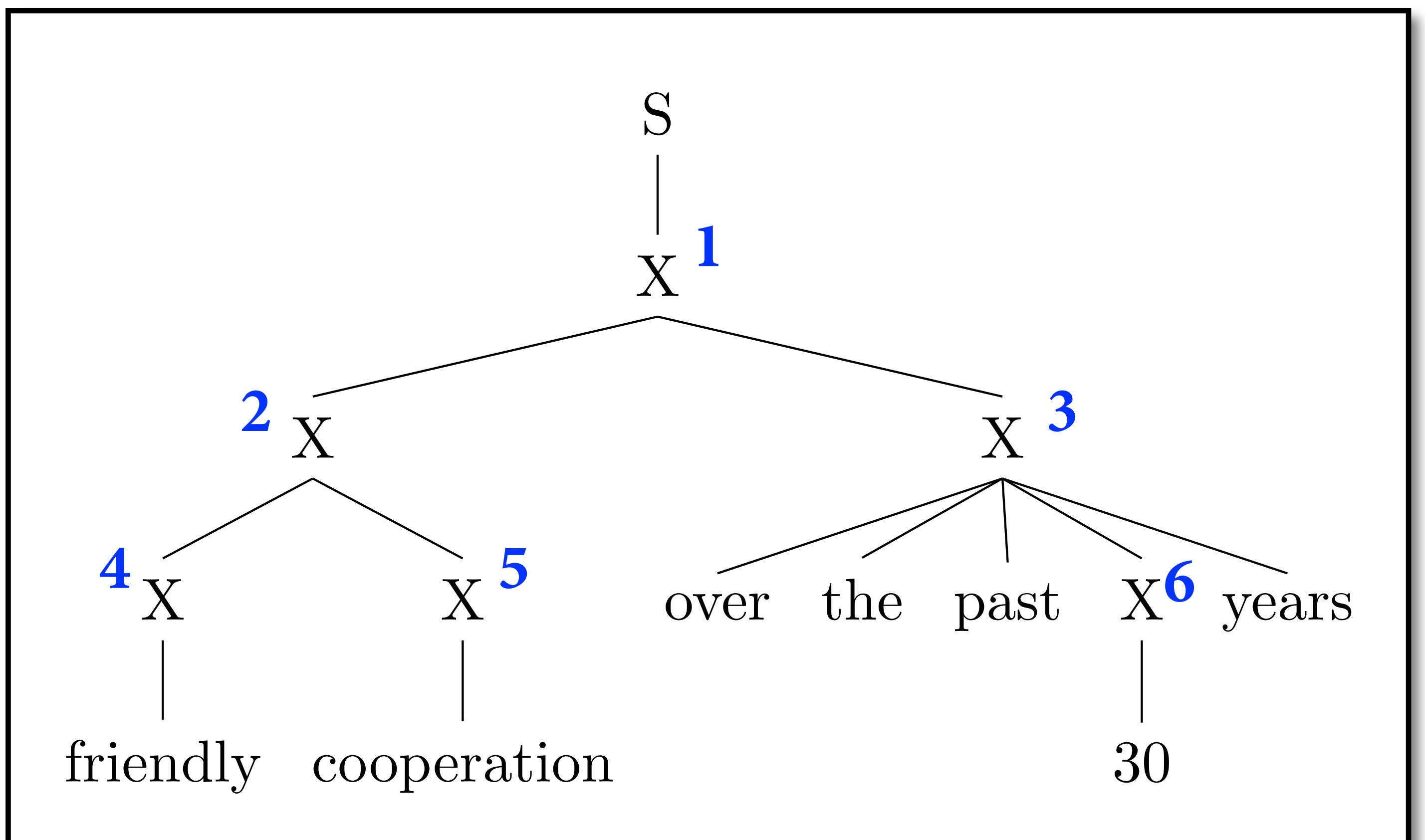
X → X① de X② / X② X①

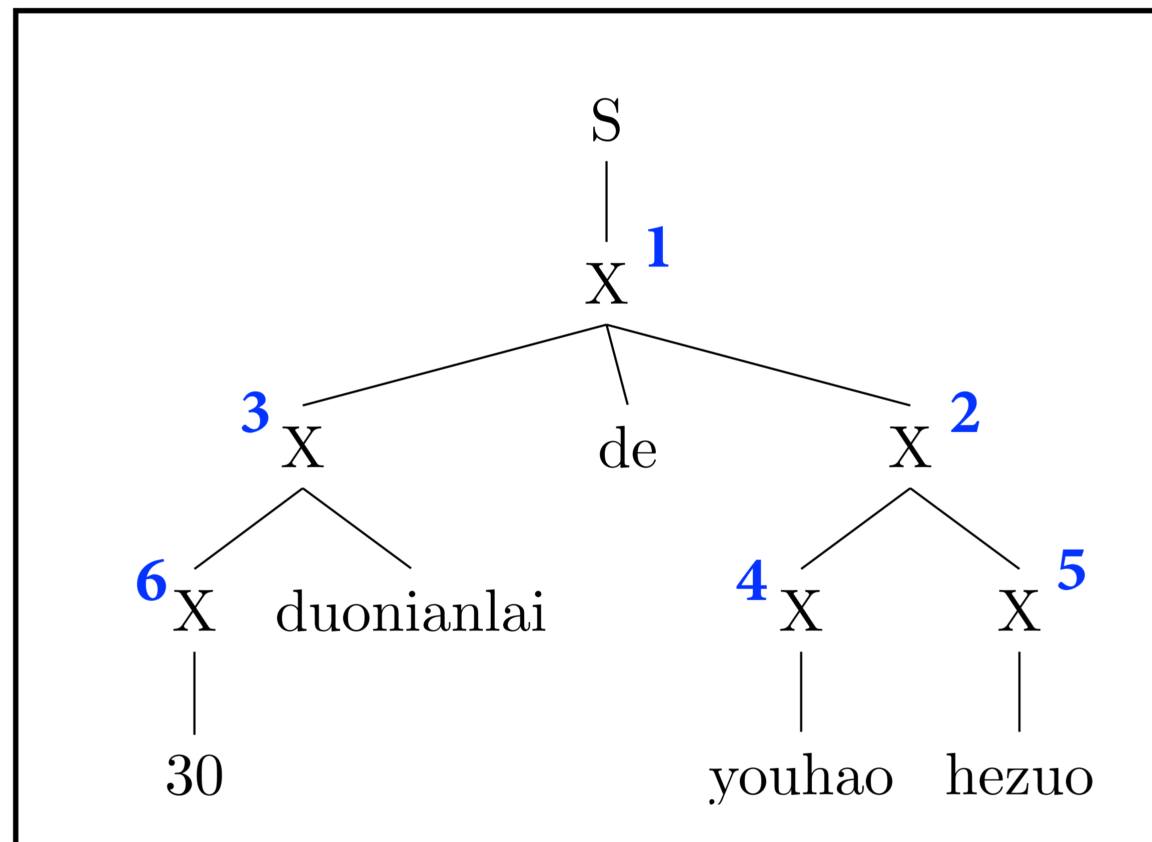
X → X① X② / X① X②

X → X① duonianlai / over the last X① years

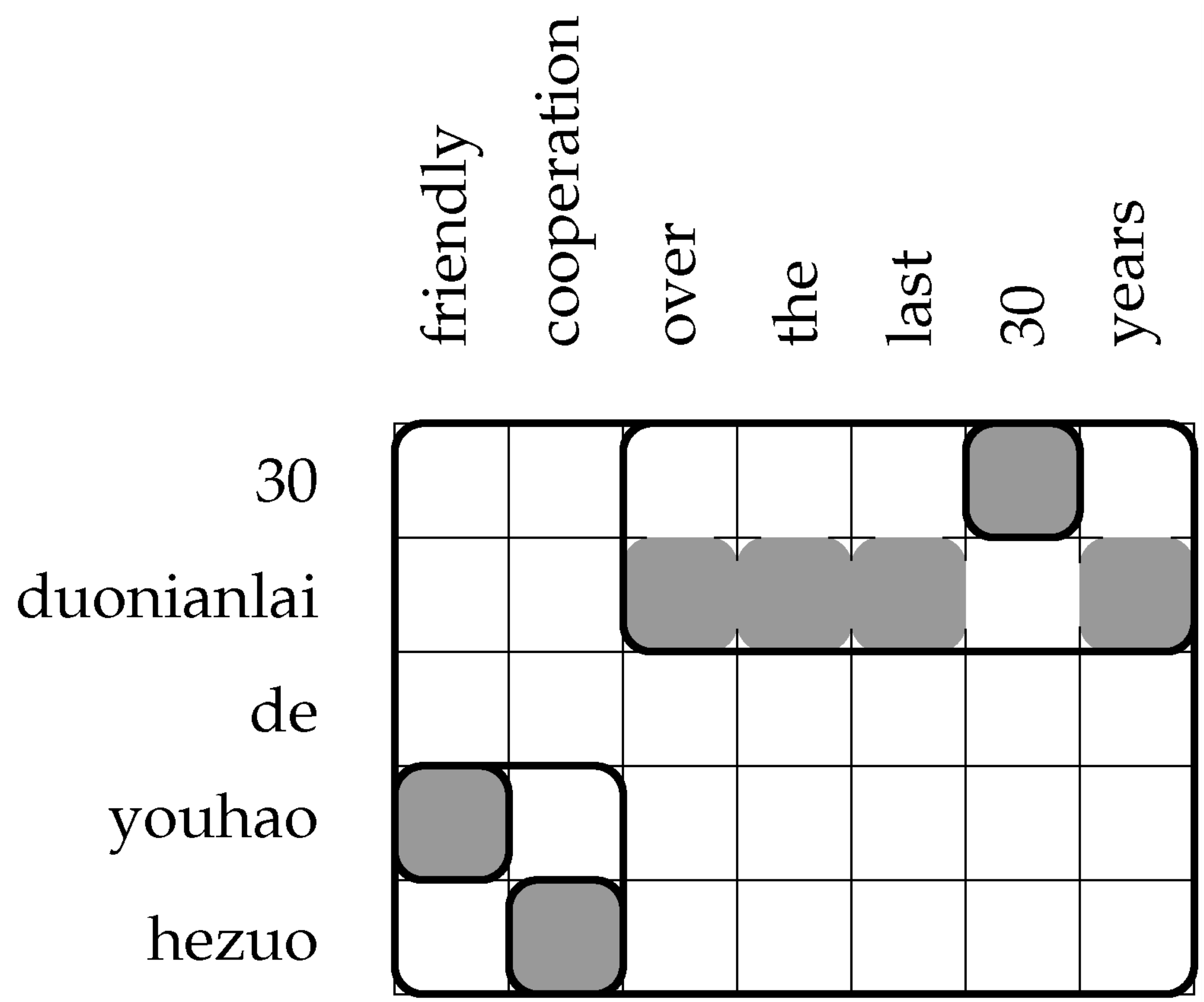
X → yuohao / friendly

X → 30 / 30





SCFG rule extraction



X → yuohao / friendly
X → 30 duonianlai / over the last 30 years
X → 30 / 30
X → X① duonianlai / over the last X① years
X → X① X② / X① X②
X → X① de X② / X② X①

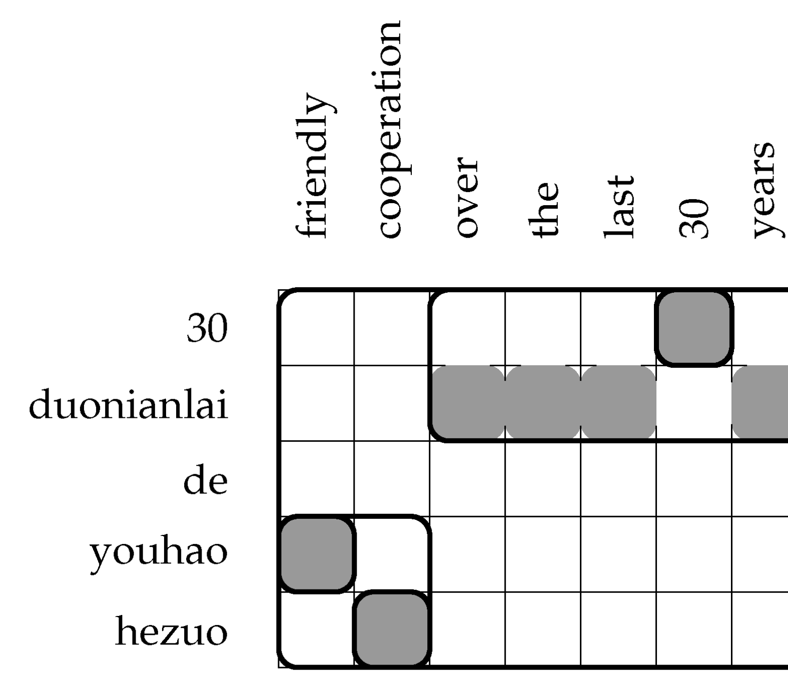
- Extract all phrase pairs as usual.
- Generate more rules by replacing sub-phrases by nonterminal X.
- Add "glue rules" S → S① X② / S① X② and S → X① / X①
  to start derivations.

Decoding schema

f = "30 duonianlai de youhao hezuo"

$$\frac{\begin{matrix}[X, 0, 2] \\ \textit{over … years}\end{matrix} \qquad \begin{matrix}[X, 3, 5] \\ \textit{friendly … coop.}\end{matrix} \qquad X \to X① \text{ de } X② \ / \ X② \ X①}{\begin{matrix}[X, 0, 5] \\ \textit{friendly … years}\end{matrix}}$$

$$\text{prob} = p_1 * p_2 * P(\text{rule}) * P_{LM}(\text{over} \mid \text{coop.})$$



Pruning

- Problem: number of items blown up by factor of $|V|^{2m}$ for an m-gram language model.

- Tackle with beam search: for each [X, i, j] for Foreign positions i, j, keep only the *k* best analyses.

- *Cube pruning*: improve runtime further by only computing a subset of the top *k* best analyses for each item by need.

BLEU Comparison

System	MT03	MT04	MT05
Hiero Monotone	28.27 ± 1.03	28.83 ± 0.74	26.35 ± 0.92
ATS	30.84 ± 0.99	31.74 ± 0.73	30.50 ± 0.95
Hiero	33.72 ± 1.12	34.57 ± 0.82	31.79 ± 0.91

phrase-based

Results: BLEU and Speed

Method	Settings	Time	BLEU
rescore	$k = 10^4$	16	33.31
rescore	$k = 10^5$	139	33.33
intersect*		1455	37.09
cube prune	$\varepsilon = 0$	23	36.14
cube prune	$\varepsilon = 0.1$	35	36.77
cube prune	$\varepsilon = 0.2$	111	36.91

time in seconds per sentence

Conclusion

- Noisy channel translation: combine translation model with language model.
- Phrase-based translation: Extract phrases (= arbitrary substrings) from word alignments.
  - different reordering models, e.g. with SCFGs
- Decoding algorithms must deal with huge search space. Need to do some clever form of beam search.
- Much current research uses neural networks instead.