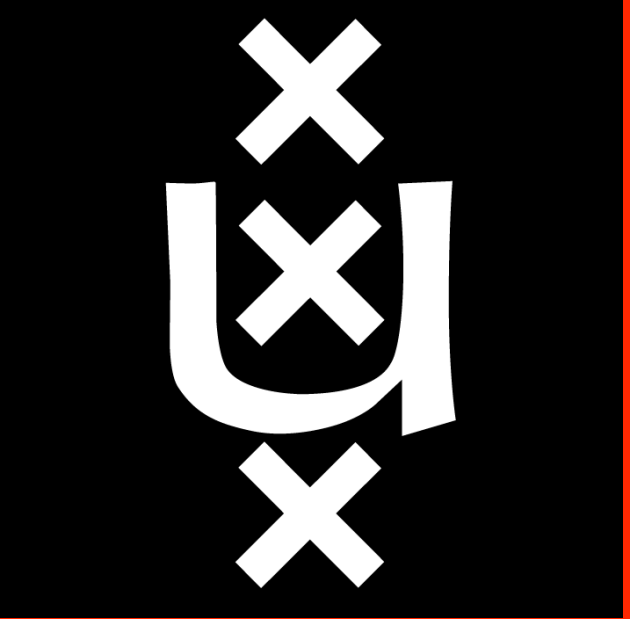


Christof Monz

Applied Language Technology
Reordering

Today's Class

- Distortion limit and distortion constraints
- Lexicalized reordering modeling
- Hierarchical Reordering modeling

Model Errors of Decoding

- Model errors occur if the highest model-scoring hypothesis is scored lower by some quality metric (e.g., BLEU) than another hypothesis with a lower model score
- The models used during decoding are inappropriate due to
  - noisy, unreliable estimates
  - lack of coverage
  - over-fitting to the training data
  - under-fitting of the models

Linear Reordering

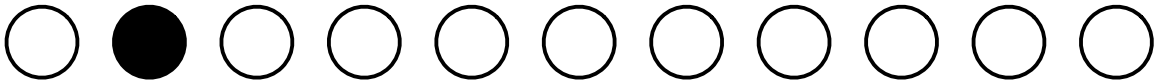
- So far we have only considered linear reordering where
  - phrases can be translated in any order
  - a distortion cost is computed based on the linear difference between the last translated word of the previously translated phrase (s_i) and the beginning of the next phrase (s_j): $h_{LD}(s_i, s_j) = -1 \cdot |\text{first\_pos}(s_j) - \text{last\_pos}(s_i) - 1|$
- In practice not all reorderings can be considered (for efficiency reasons) and distortion limits are introduced:
  - $\text{DL} = n$: Only consider reordering (s_i, s_j) if $h_{LD}(s_i, s_j) \leq n$
- How much do we lose by limiting reordering?
  - Consider long-distance reorderings (such as verb-final movements in German-English)

Linear Reordering Constraints

- A distortion limit by itself still allows for too many reorderings

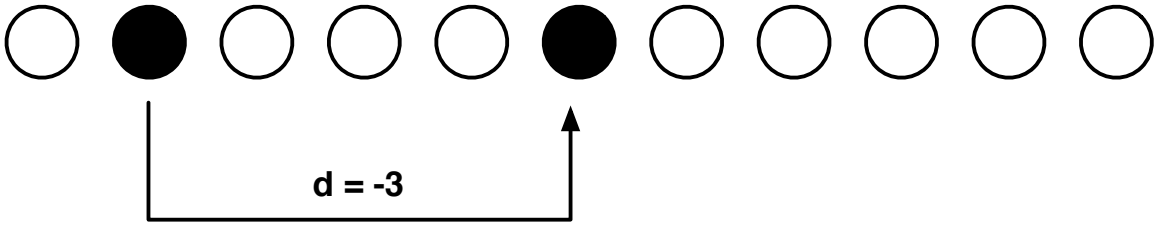
Linear Reordering Constraints

- A distortion limit by itself still allows for too many reorderings



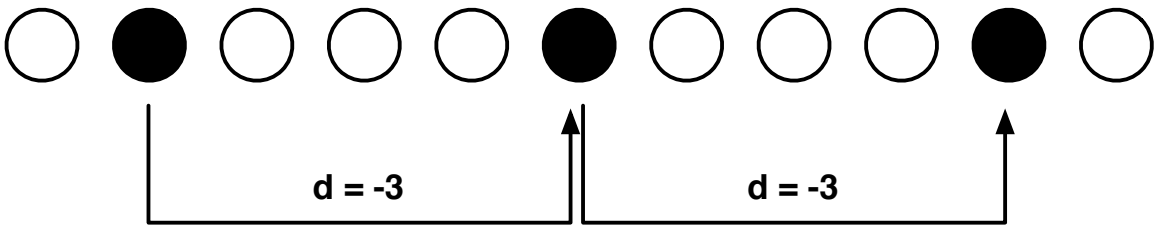
Linear Reordering Constraints

- A distortion limit by itself still allows for too many reorderings



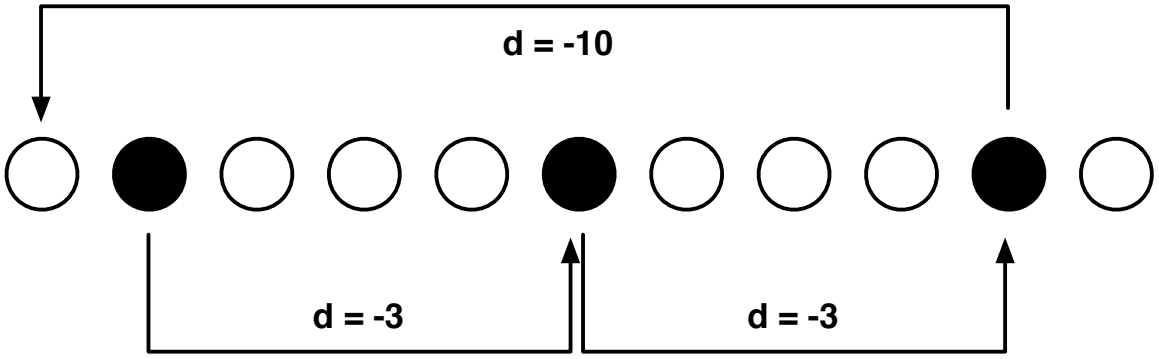
Linear Reordering Constraints

- A distortion limit by itself still allows for too many reorderings



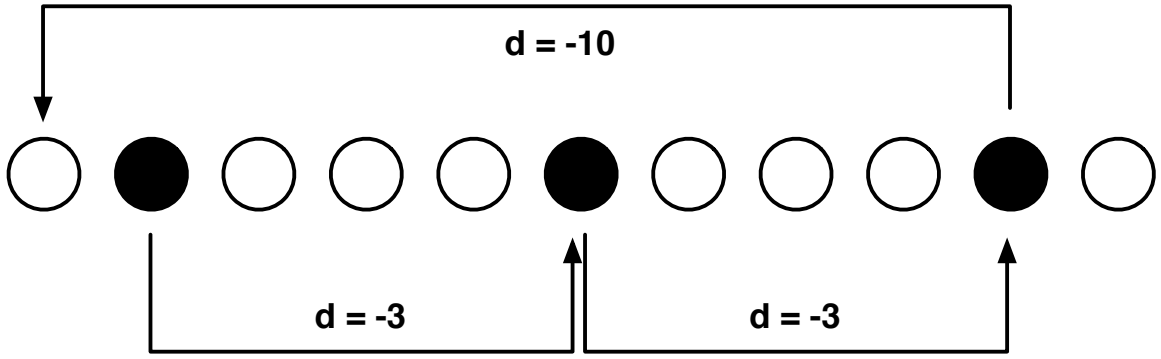
Linear Reordering Constraints

- A distortion limit by itself still allows for too many reorderings



Linear Reordering Constraints

- A distortion limit by itself still allows for too many reorderings



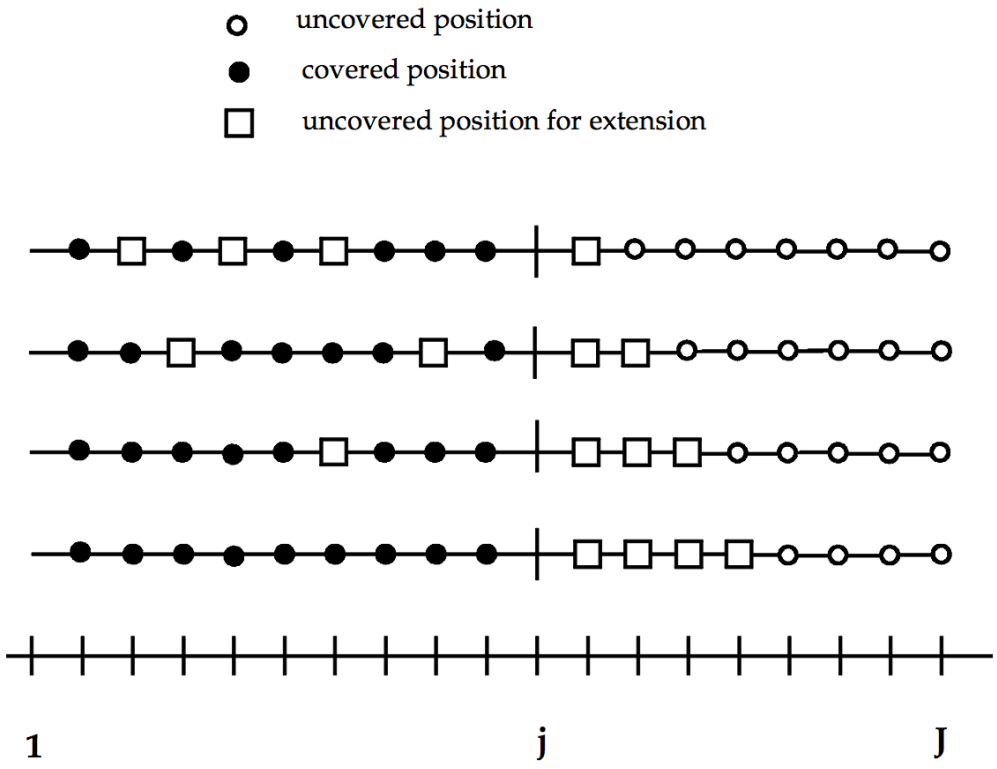
- Additional constraints are required that take the 'global' amount of reordering into account

IBM Constraints

- The IBM constraints allow for reorderings such that at most k positions are uncovered before the last covered position

IBM Constraints

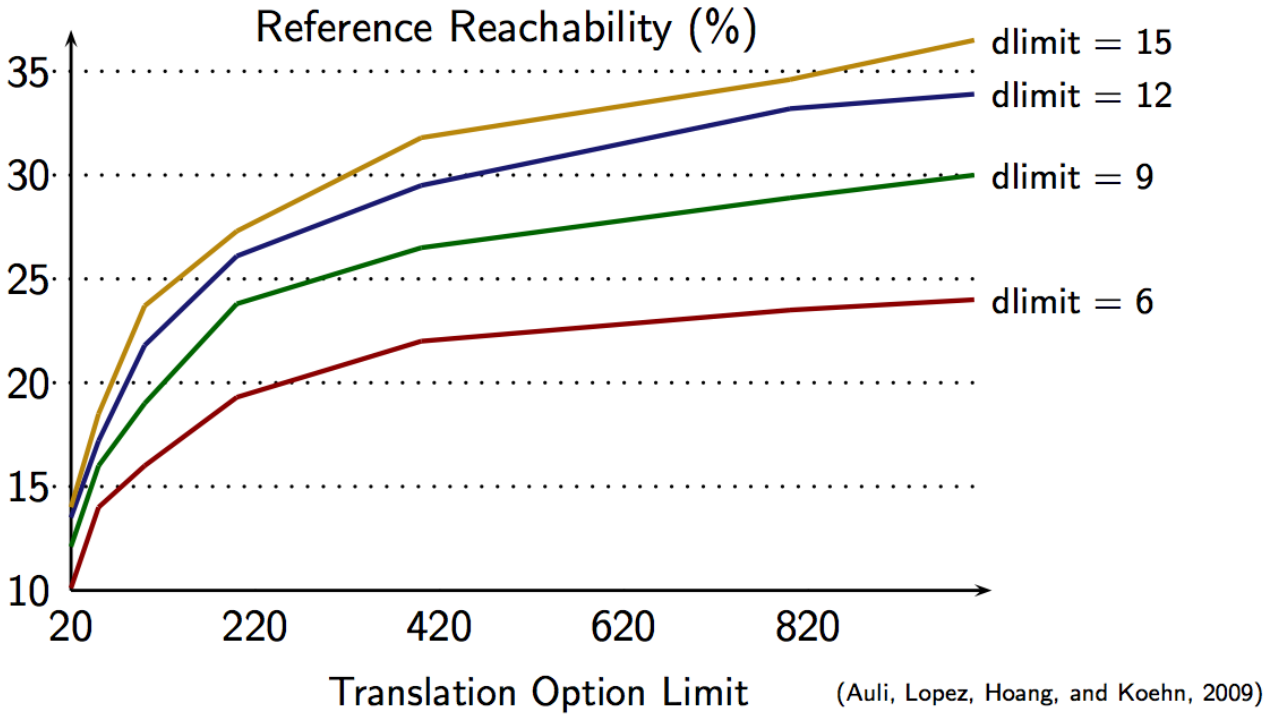
- The IBM constraints allow for reorderings such that at most k positions are uncovered before the last covered position



IBM Constraints

- The IBM constraints are often used in combination with a limit on the window size
- A window $w = n$ specifies the maximum distance (n) between the first uncovered position and the last translated word

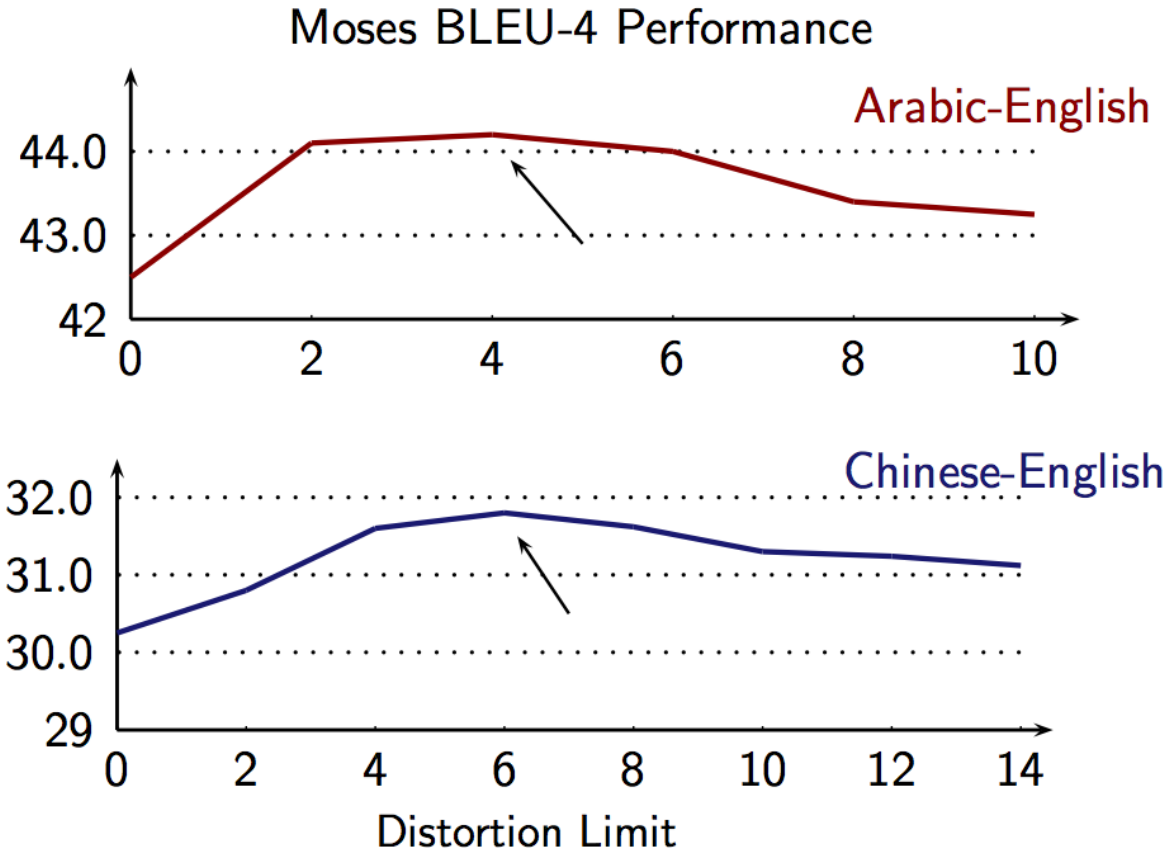
Search Constraints and Model Errors



(Auli, Lopez, Hoang, and Koehn, 2009)

- Increasing the distortion limit results in larger percentage of reachable references
- A reference (r) is reachable by the decoder if it can be produced by it, i.e., if $r \in \tilde{E}$

Distortion Limit and Quality



(Green et al., 2010)

- Increasing the distortion limit leads to increased noise and difficulties in discriminating good and poor hypotheses

Reordering and LMs

- Al-Onaizan & Papineni (2006): Reordering English wrt original Arabic word order

Arabic	Ezp_1 $AbrAhym_2$ $ystqbl_3$ $ms\&wlA_4$ $AqtSAdyA_5$ $sEwdyA_6$ fy_7 $bgdAd_8$
English	$Izzet_1$ $Ibrahim_2$ $Meets_3$ $Saudi_4$ $Trade_5$ $official_6$ in_7 $Baghdad_8$
Word Alignment	(Ezp_1,$Izzet_1$) ($AbrAhym_2$,$Ibrahim_2$) ($ystqbl_3$,$Meets_3$) ($ms\&wlA_4$,$official_6$) ($AqtSAdyA_5$,$Trade_5$) ($sEwdyA_6$,$Saudi_4$) (fy_7,in_7) ($bgdAd_8$,$Baghdad_8$)
Reordered English	$Izzet_1$ $Ibrahim_2$ $Meets_3$ $official_6$ $Trade_5$ $Saudi_4$ in_7 $Baghdad_8$

Reordering and LMs

- Al-Onaizan & Papineni (2006): Reordering English wrt original Arabic word order

Arabic	Ezp_1 AbrAhym_2 ystqbl_3 $\text{ms\&wlA}_4$ AqtSAdyA_5 sEwdyA_6 fy_7 bgdAd_8
English	Izzet_1 Ibrahim_2 Meets_3 Saudi_4 Trade_5 official_6 in_7 Baghdad_8
Word Alignment	(Ezp_1,Izzet_1) (AbrAhym_2,Ibrahim_2) (ystqbl_3,Meets_3) ($\text{ms\&wlA}_4$,official_6) (AqtSAdyA_5,Trade_5) (sEwdyA_6,Saudi_4) (fy_7,in_7) (bgdAd_8,Baghdad_8)
Reordered English	Izzet_1 Ibrahim_2 Meets_3 official_6 Trade_5 Saudi_4 in_7 Baghdad_8

- How well can the LM recover the correct order?

s	0	1	1	1	1	1	2	2	2	2
w	0	4	6	8	10	12	4	6	8	10
BLEUr1n4c	0.5617	0.6507	0.6443	0.6430	0.6461	0.6456	0.6831	0.6706	0.6609	0.6596

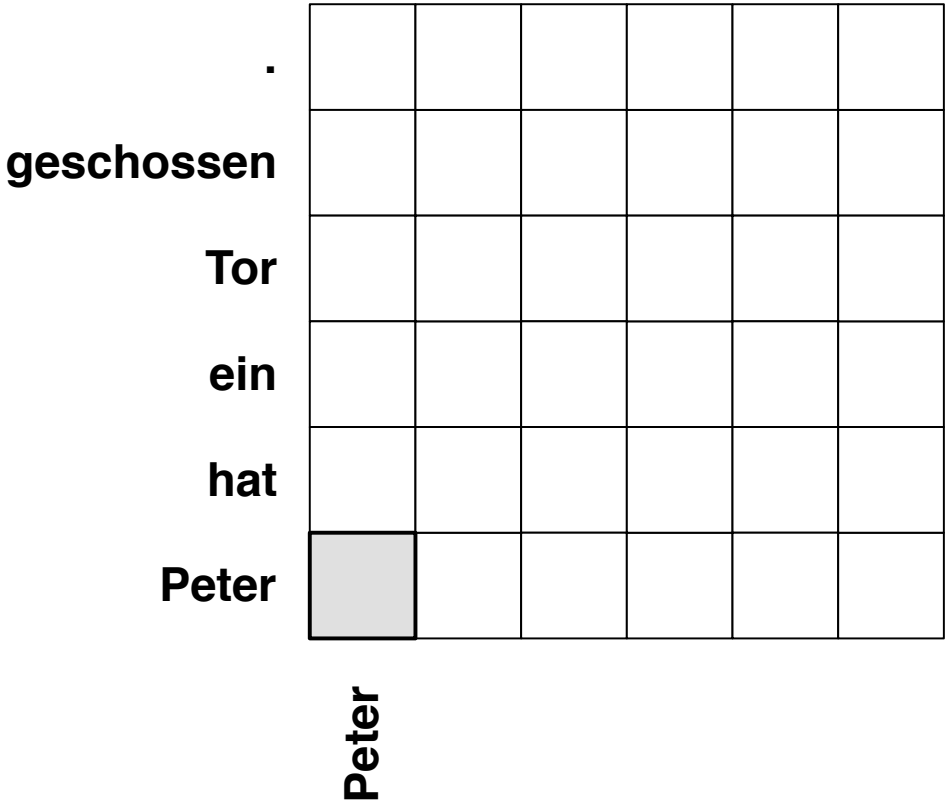
2	3	3	3	3	3	4	4	4	4	4
12	4	6	8	10	12	4	6	8	10	12
0.6626	**0.6919**	0.6751	0.6580	0.6505	0.6490	0.6851	0.6592	0.6317	0.6237	**0.6081**

Lexicalized Reordering

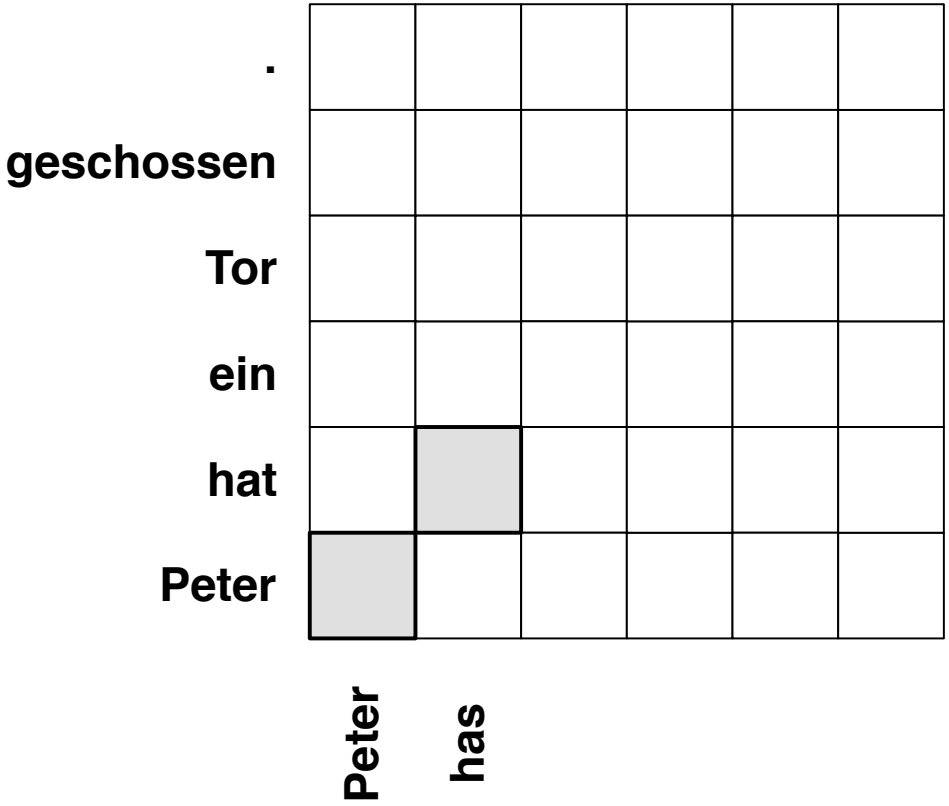
- Linear reordering is very lightly parameterized
  - only the length (+ some penalty for changing orientation)
- No information is encoded relating to
  - Syntactic structure
    - Source syntax
    - Target syntax
  - Lexicalization of the words/phrases involved in a reordering
- Lexicalized reordering (Tillmann 2004 and Koehn et al. 2005) introduces additional parameters
  - Which phrase (pair) are we jumping from
  - Which phrase (pair) are we jumping to

Lexicalized Reordering

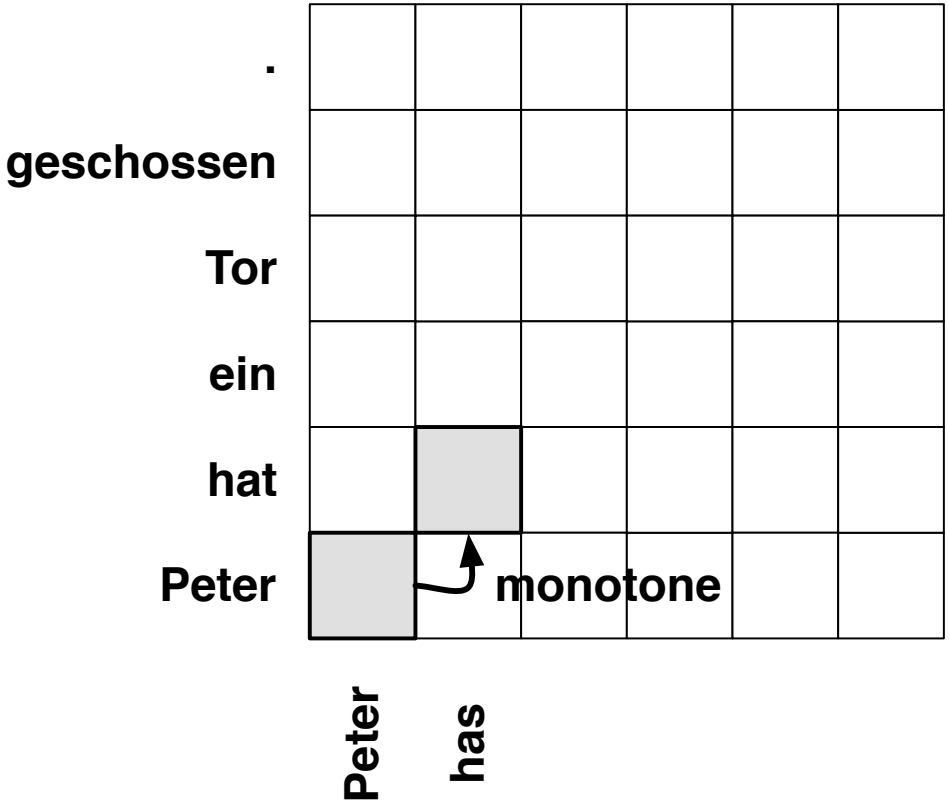
Lexicalized Reordering



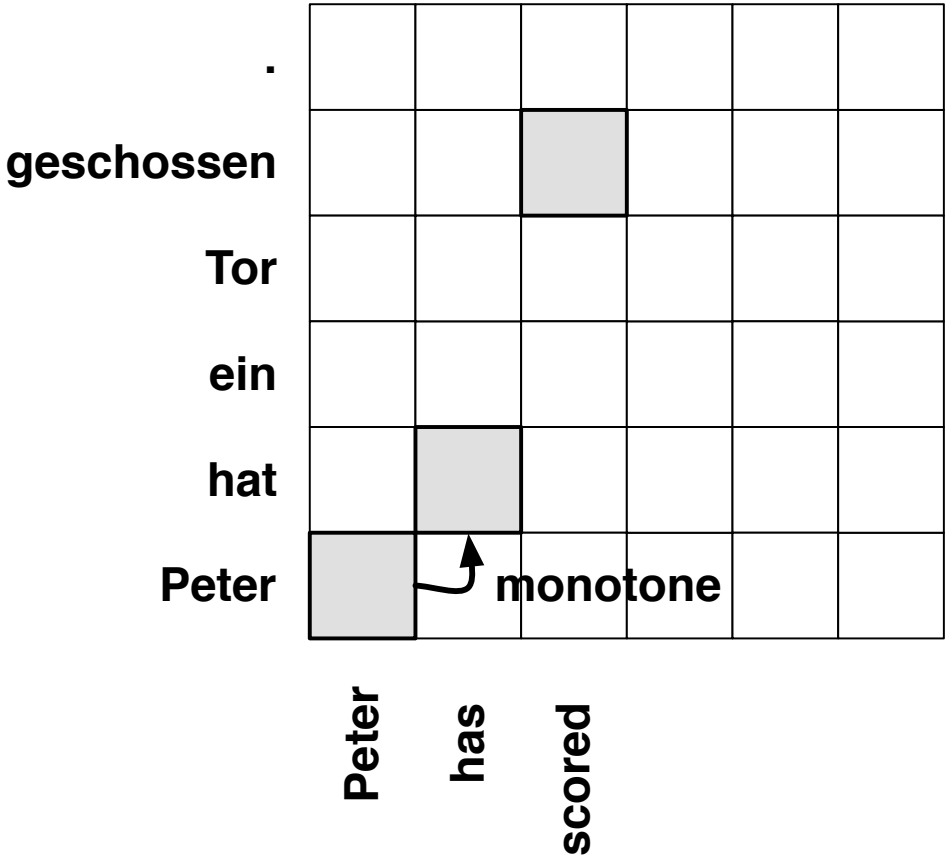
Lexicalized Reordering



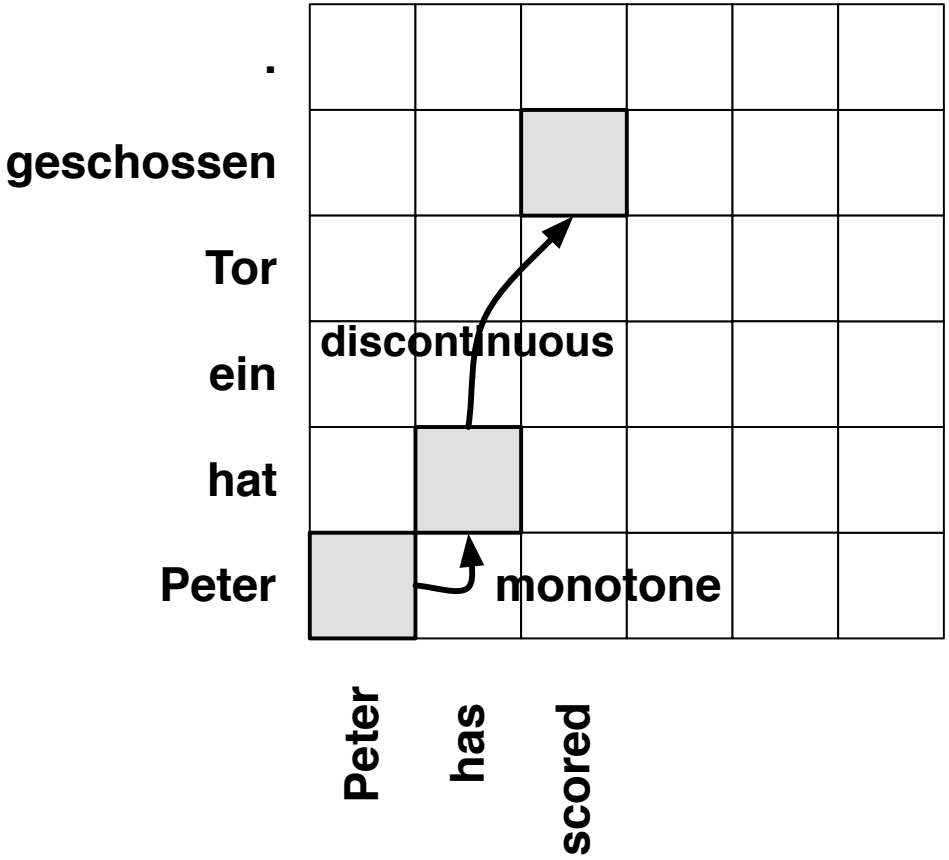
Lexicalized Reordering



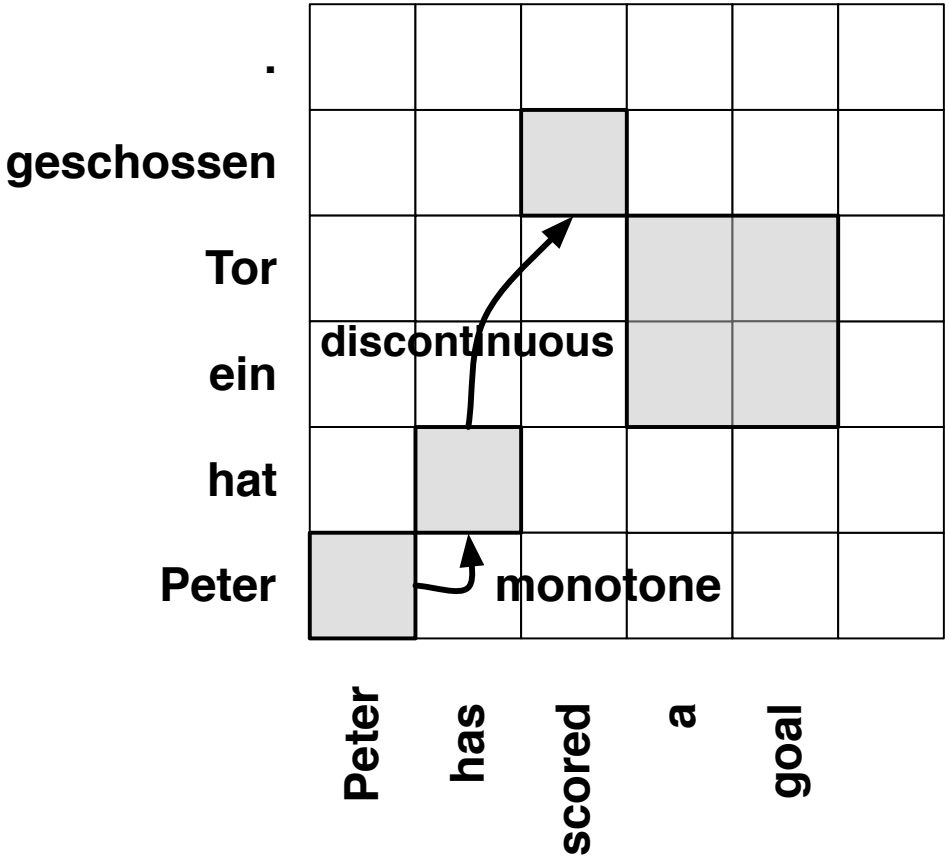
Lexicalized Reordering



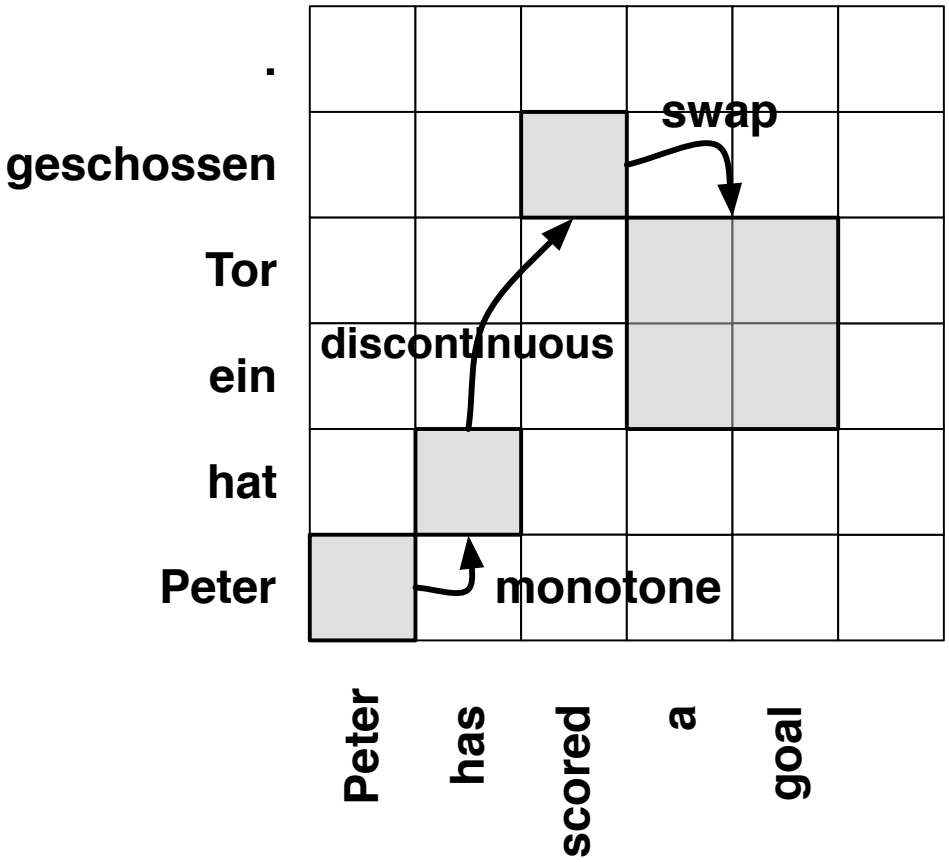
Lexicalized Reordering



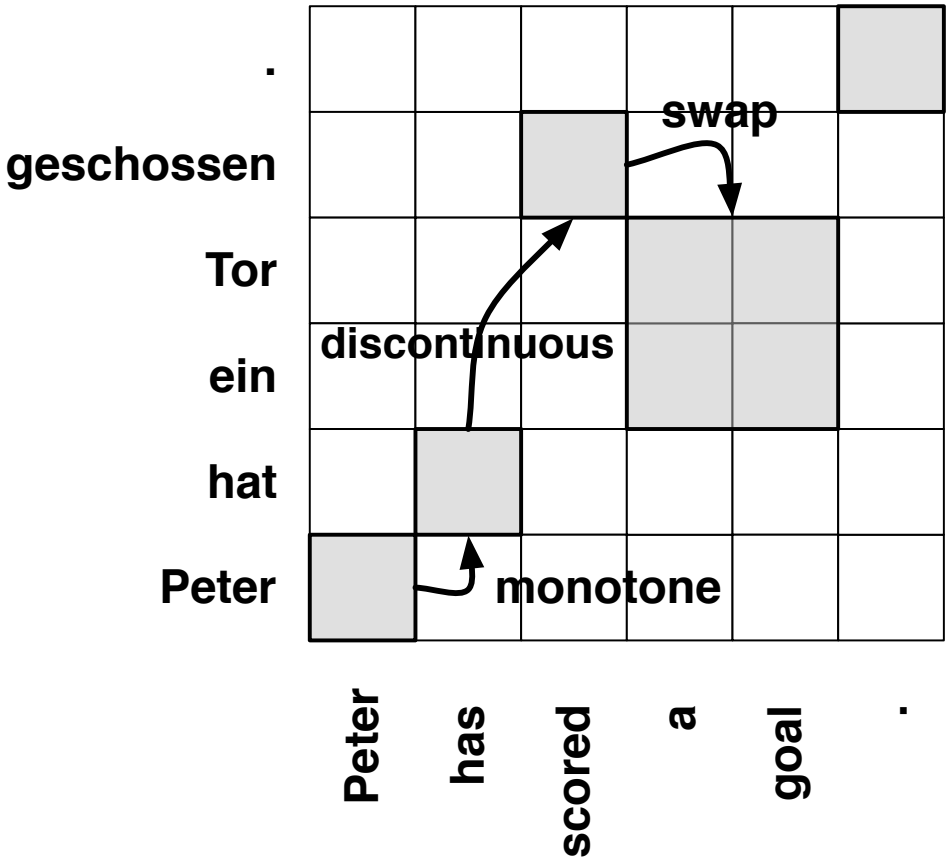
Lexicalized Reordering



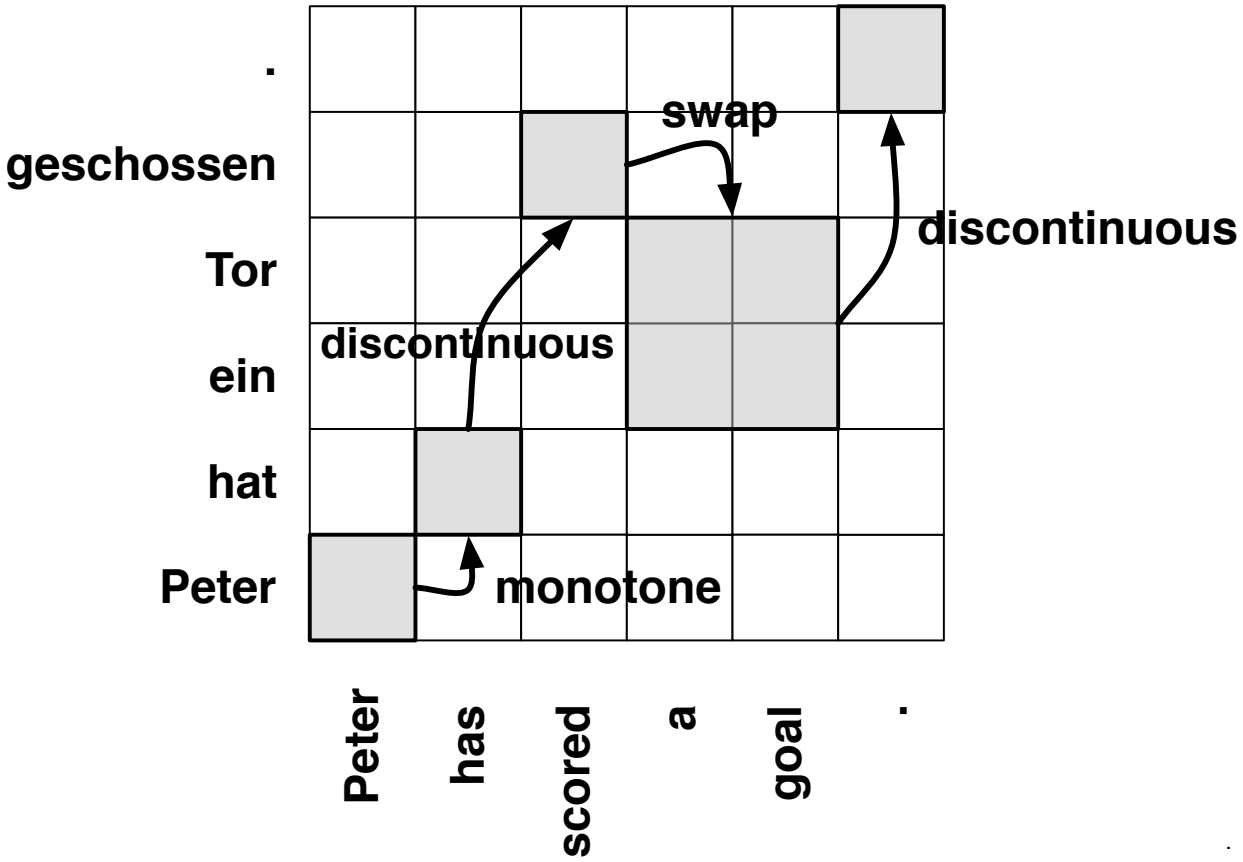
Lexicalized Reordering



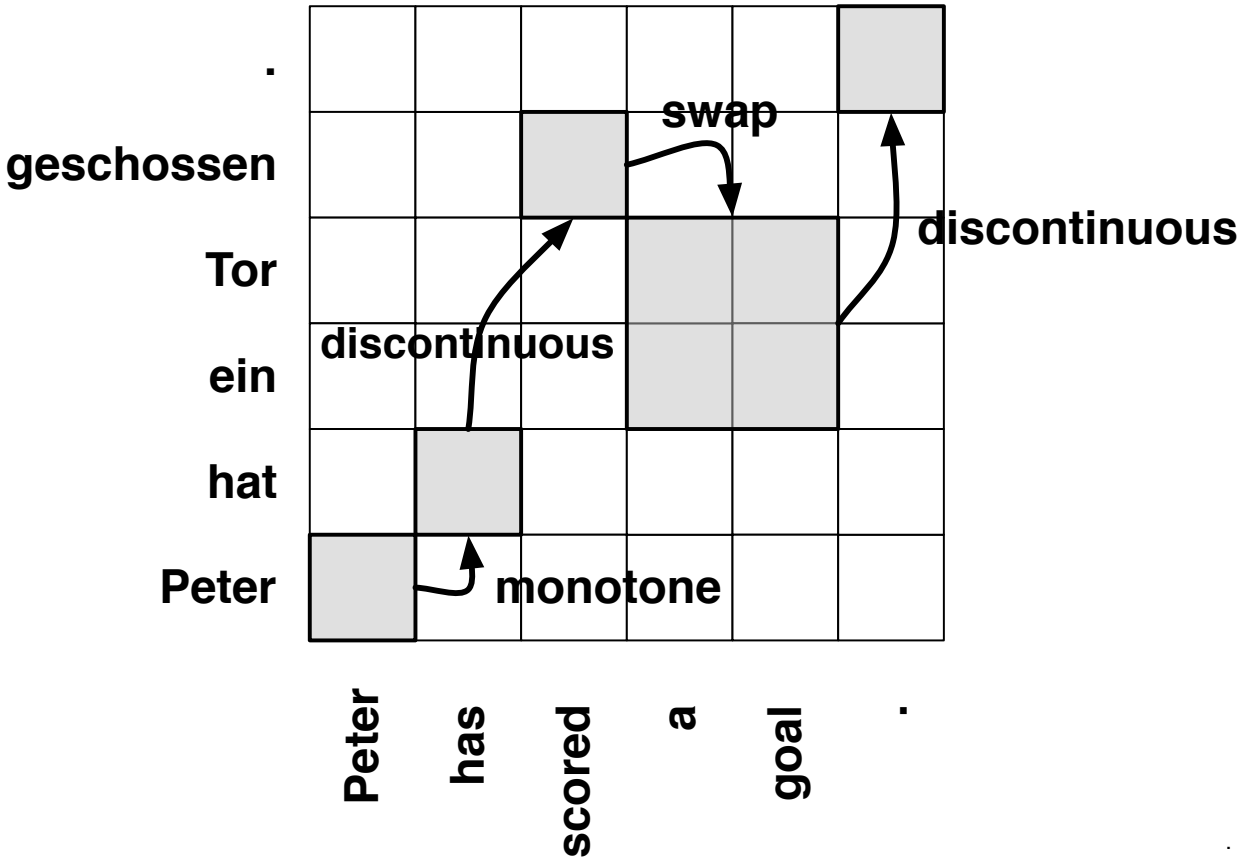
Lexicalized Reordering



Lexicalized Reordering

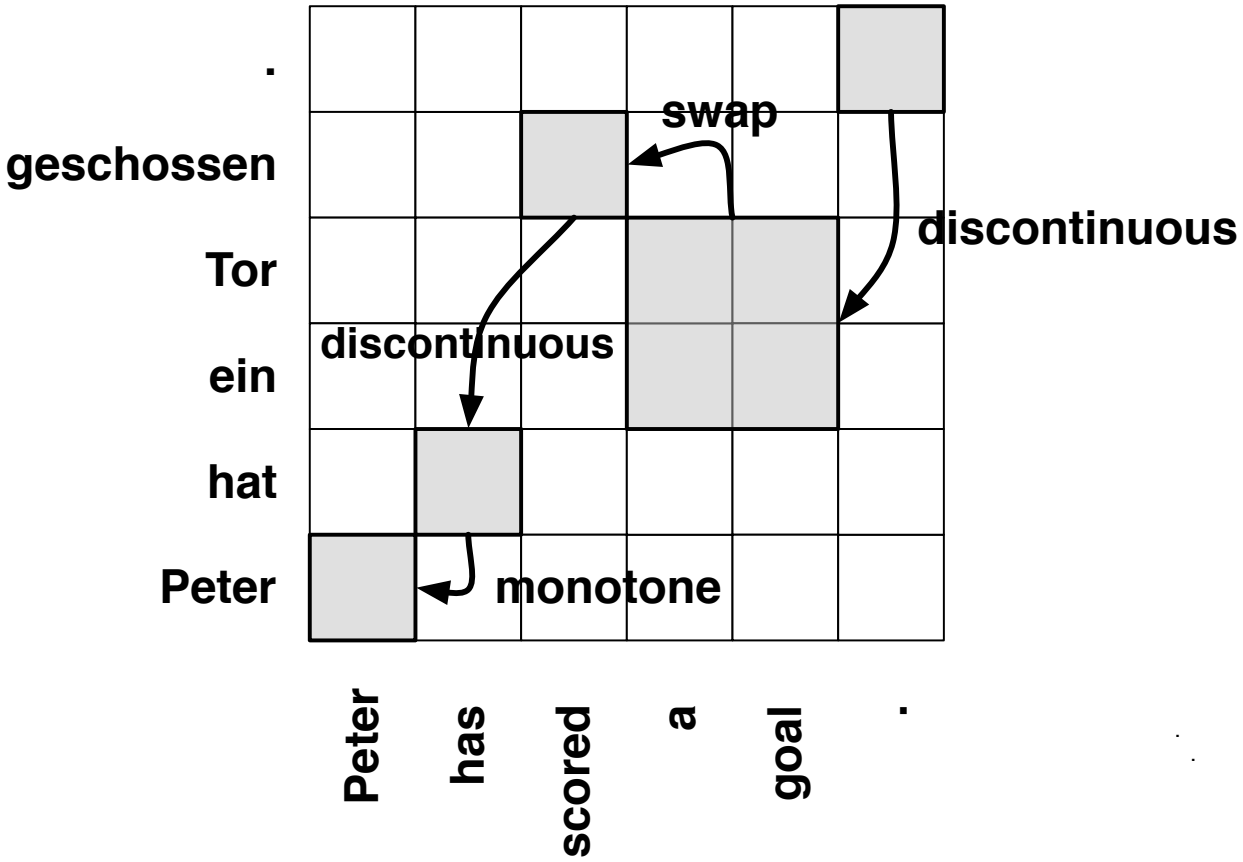


Lexicalized Reordering

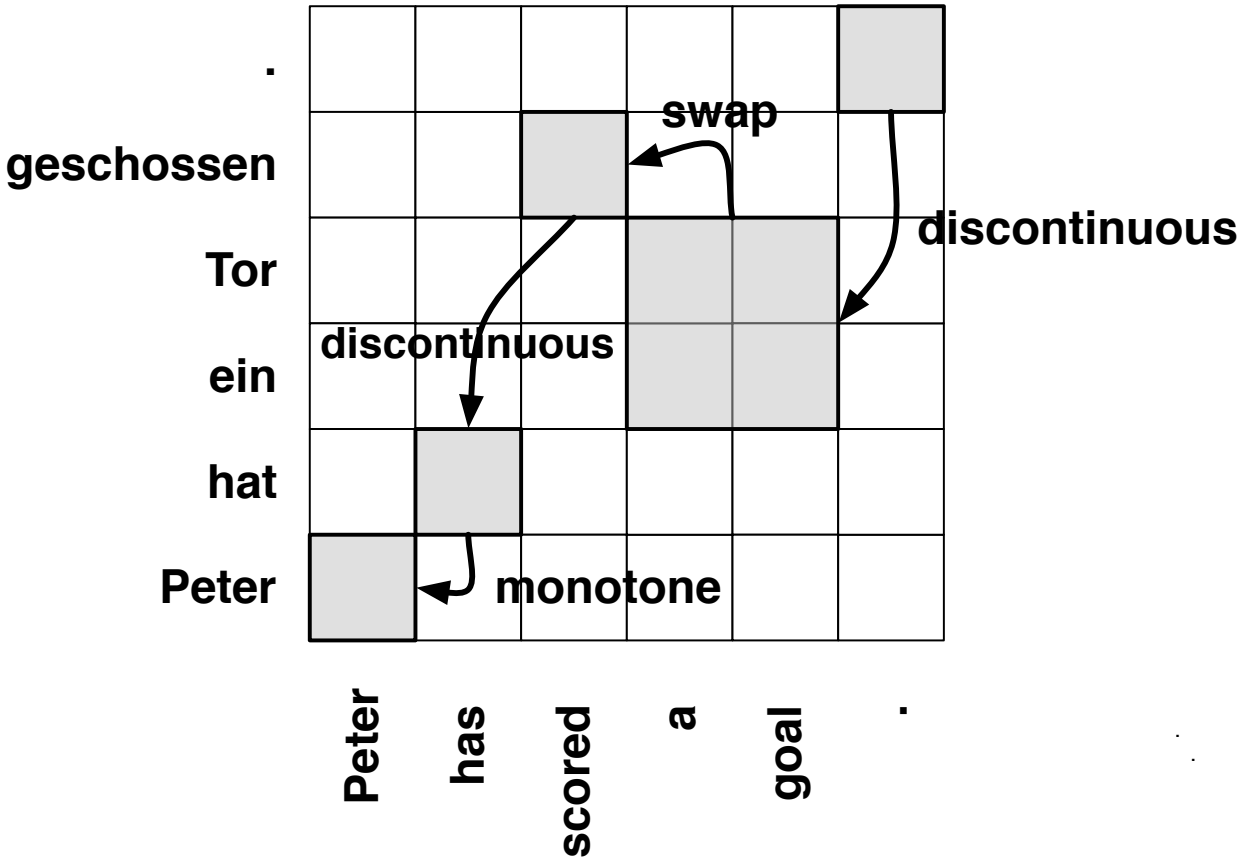


- Reordering probabilities are of the form: $p(o|(\bar{f},\bar{e}))$, where $o \in \{\text{monotone}, \text{discontinous}, \text{swap}\}$

Lexicalized Reordering



Lexicalized Reordering



- We can also condition on the next phrase resulting in two different distributions: $p_{l \to r}(o|(\bar{f}, \bar{e}_i))$ and $p_{r \to l}(o|(\bar{f}', \bar{e}_{i+1}))$

Lexicalized Reordering

- Given two phrase applications: $(\bar{f}_i^j, \bar{e}_k^l)$ and $(\bar{f}_s^t, \bar{e}_{l+1}^m)$
- The orientation (o) of a reordering is
  - **monotone (m):** if $s = j+1$
  - **swap (s):** if $i = t+1$
  - **discontinuous (d):** otherwise
- Left-to-right reordering conditions on the previous phrase pair wrt the next phrase pair: $p_{l\to r}(o|(\bar{f}_i^j, \bar{e}_k^l))$
- Right-to-left reordering conditions on the next phrase pair wrt to the previous phrase pair: $p_{r\to l}(o|(\bar{f}_s^t, \bar{e}_{l+1}^m))$
- The total lexicalized reordering probability is the product of both orientations: $p_{LRM}(o|(\bar{f}_i^j, \bar{e}_k^l), (\bar{f}_s^t, \bar{e}_{l+1}^m))$
  $= p_{l\to r}(o|(\bar{f}_i^j, \bar{e}_k^l)) \cdot p_{r\to l}(o|(\bar{f}_s^t, \bar{e}_{l+1}^m))$

Lexicalized Reordering

- There are a number of ways to weight the contributions of reordering models
  - One weight for reordering: $p_{LRM}(o|(\bar{f}_i^j, \bar{e}_k^l), (\bar{f}_s^t, \bar{e}_{l+1}^m))^{\lambda_{LRM}}$
  - One weight for each direction:

    $p_{LRM}(o|(\bar{f}_i^j, \bar{e}_k^l), (\bar{f}_s^t, \bar{e}_{l+1}^m)) =$
    $p_{l\rightarrow r}(o|(\bar{f}_i^j, \bar{e}_k^l))^{\lambda_{l\rightarrow r}} \cdot p_{r\rightarrow l}(o|(\bar{f}_s^t, \bar{e}_{l+1}^m))^{\lambda_{r\rightarrow l}}$
  - One weight for each direction and orientation:

    $p_{LRM}(o|(\bar{f}_i^j, \bar{e}_k^l), (\bar{f}_s^t, \bar{e}_{l+1}^m)) =$
    $p_{l\rightarrow r}(o|(\bar{f}_i^j, \bar{e}_k^l))^{\lambda_{l\rightarrow r(o)}} \cdot p_{r\rightarrow l}(o|(\bar{f}_s^t, \bar{e}_{l+1}^m))^{\lambda_{r\rightarrow l(o)}}$
- Experimental results show that individual weights for each direction and orientation lead to the best performance
  - Downside: 6 weight parameters instead of 2 (or 1) need to be estimated

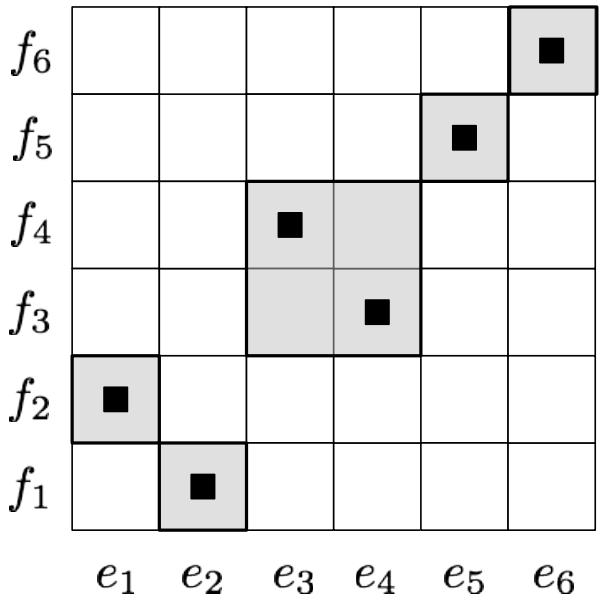
Estimating LRMs

- Lexicalized reordering models are estimated similarly to the way phrase translation models are estimated
- Using maximum likelihood, we can define

$$p_{r\to l}(o|(\bar{f},\bar{e})) = \frac{c_{r\to l}(o,(\bar{f},\bar{e}))}{\sum_{o\in O} c_{r\to l}(o,(\bar{f},\bar{e}))}$$

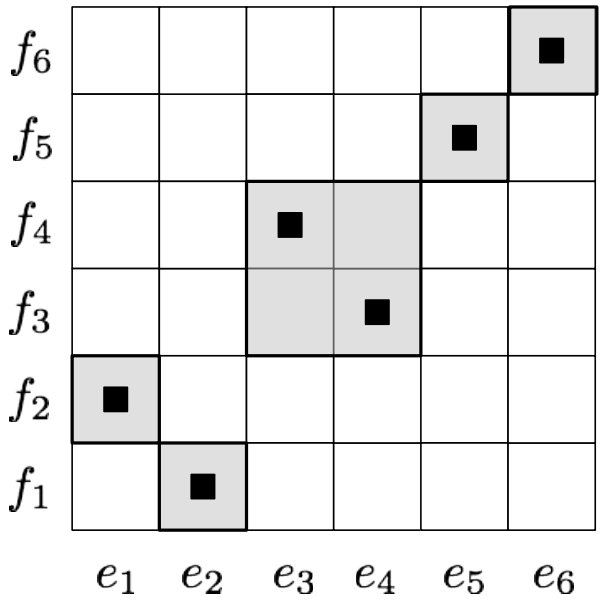
- How do we collect the event counts $c_{r\to l}(o,(\bar{f},\bar{e}))$?

Estimating LRMs



- Here we focus on the phrase pair $(\bar{f}, \bar{e}) = ((f_3 f_4), (e_3 e_4))$

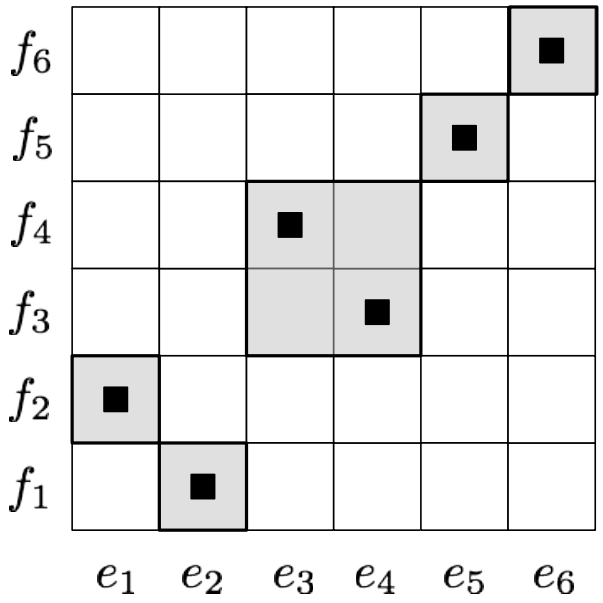
Estimating LRMs



- Here we focus on the phrase pair $(\bar{f}, \bar{e}) = ((f_3 f_4), (e_3 e_4))$

  $c_{r \to l}(d, (\bar{f}, \bar{e}))++$

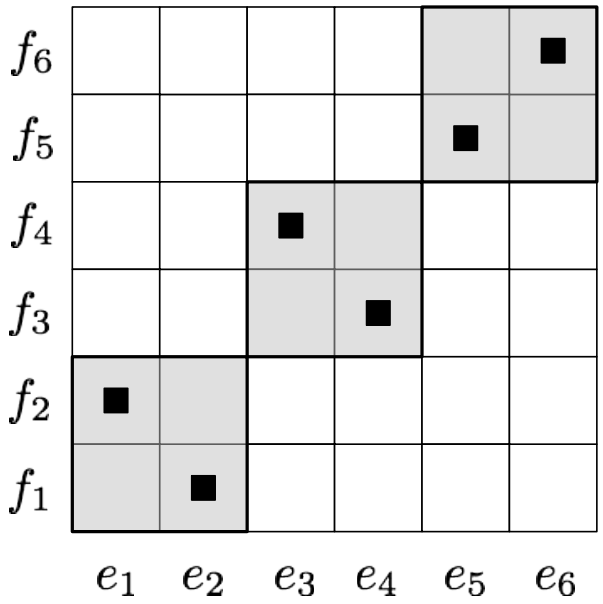
Estimating LRMs



- Here we focus on the phrase pair $(\bar{f}, \bar{e}) = ((f_3 f_4), (e_3 e_4))$

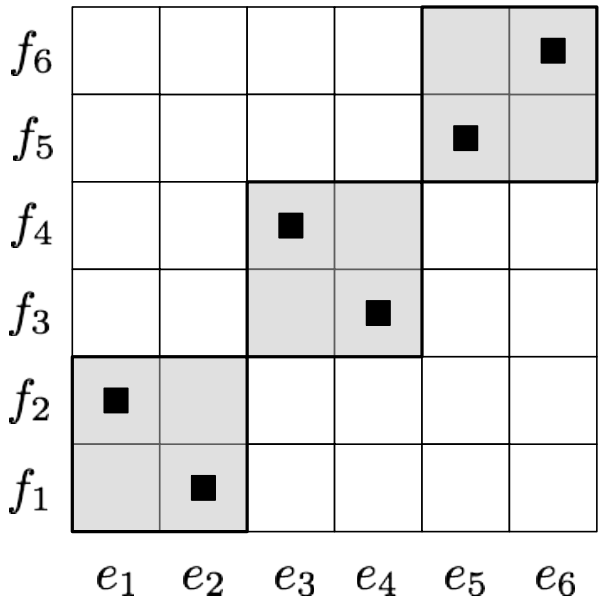
$$c_{r \to l}(d, (\bar{f}, \bar{e})) + +$$
$$c_{l \to r}(m, (\bar{f}, \bar{e})) + +$$

Estimating LRMs



- How about different phrase extractions?

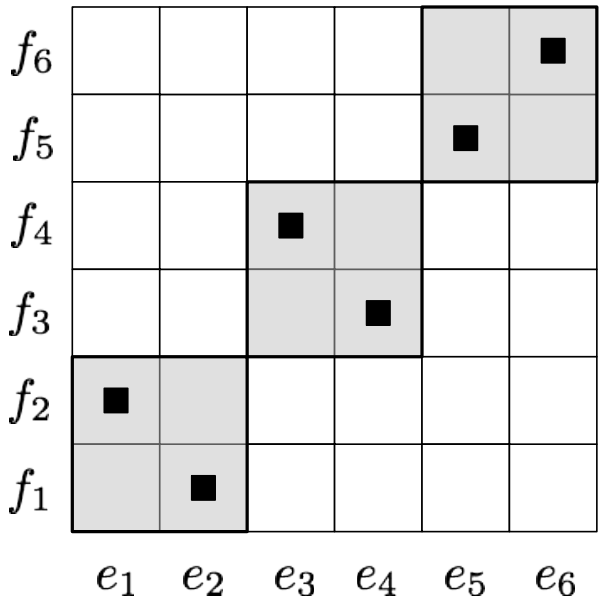
Estimating LRMs



- How about different phrase extractions?

$$c_{r \to l}(m, (\bar{f}, \bar{e})) + +$$

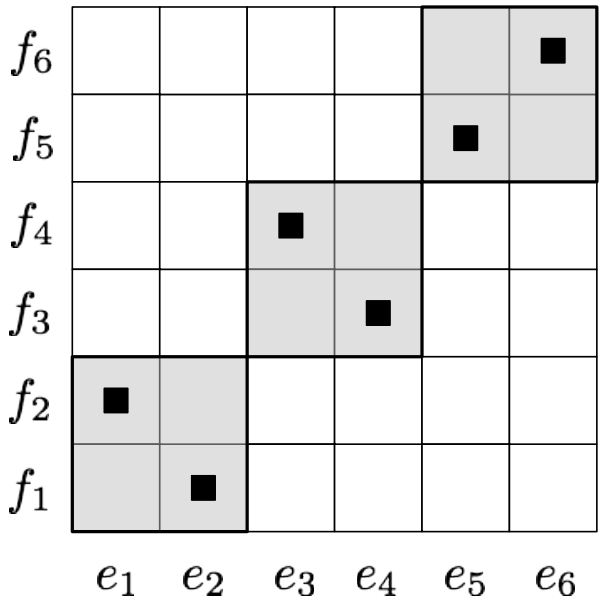
Estimating LRMs



- How about different phrase extractions?

$$c_{r\to l}(m,(\bar{f},\bar{e}))++$$
$$c_{l\to r}(m,(\bar{f},\bar{e}))++$$

Estimating LRMs



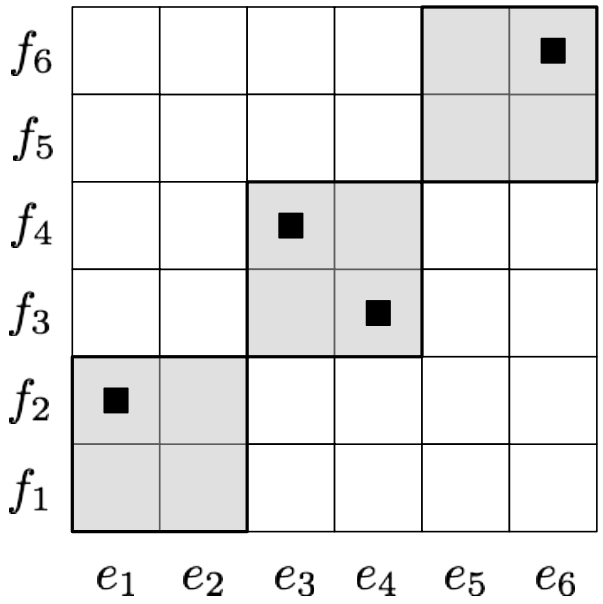
- How about different phrase extractions?

  $c_{r\to l}(m,(\bar{f},\bar{e}))++$
  $c_{l\to r}(m,(\bar{f},\bar{e}))++$

  In addition to

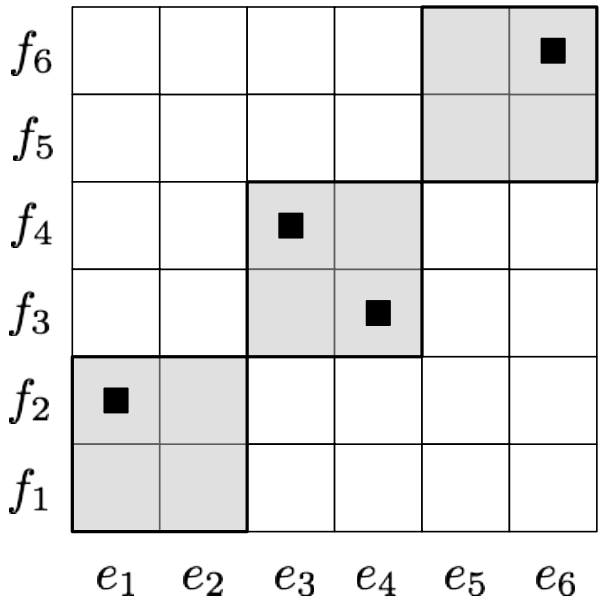
  $c_{r\to l}(d,(\bar{f},\bar{e}))++$
  $c_{l\to r}(m,(\bar{f},\bar{e}))++$?

Estimating LRMs



- How about phrases involving unaligned words?

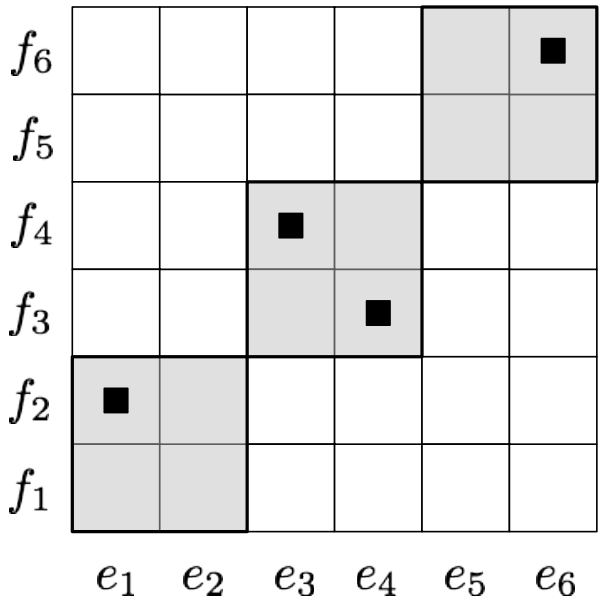
Estimating LRMs



- How about phrases involving unaligned words?

$$c_{r \to l}(m, (\bar{f}, \bar{e})) + +$$

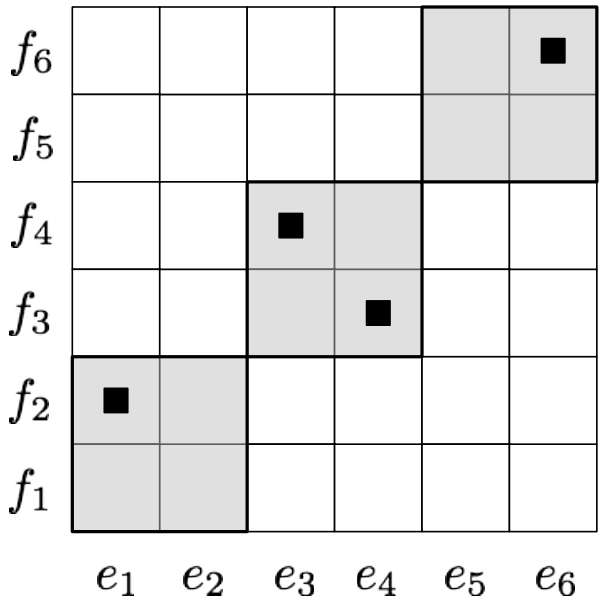
Estimating LRMs



- How about phrases involving unaligned words?

$$c_{r \to l}(m, (\bar{f}, \bar{e})) + +$$
$$c_{l \to r}(m, (\bar{f}, \bar{e})) + +$$

Estimating LRMs



- How about phrases involving unaligned words?

$$c_{r\to l}(m,(\bar{f},\bar{e}))++$$
$$c_{l\to r}(m,(\bar{f},\bar{e}))++$$

Or

$$c_{r\to l}(m,(\bar{f},\bar{e}))+=0$$
$$c_{l\to r}(m,(\bar{f},\bar{e}))+=0\ ?$$

Estimating LRMs

- There are two types of counting reordering events
  - **Word based:** consider only the word alignments surrounding the current phrase
  - **Phrase based:** consider the phrases surrounding the current phrase
- Given a sentence word alignment A, in the word-based orientation estimation approach $c_{r\rightarrow l}(o,(\bar{f},\bar{e}))++$ if
  - **o=m** and $(\text{left}(\bar{f})-1,\text{left}(\bar{e})-1)\in A$
  - **o=s** and $(\text{right}(\bar{f})+1,\text{left}(\bar{e})-1)\in A$
  - **o=d** and $\exists j:(j,\text{left}(\bar{e})-1)\in A$
    where $\text{left}(\cdot)/\text{right}(\cdot)$ is the left-/rightmost position of a phrase
- $c_{l\rightarrow r}(o,(\bar{f},\bar{e}))++$ if
  - **o=m** and $(\text{right}(\bar{f})+1,\text{right}(\bar{e})+1)\in A$
  - **o=s** and $(\text{left}(\bar{f})-1,\text{right}(\bar{e})+1)\in A$
  - **o=d** and $\exists j:(j,\text{right}(\bar{e})+1)\in A$

Estimating LRMs

- Phrase-based orientation estimation only considers the phrases next to the current phrase
- $c_{r \to l}(o, (\bar{f}, \bar{e})) + +$ if
  - **o=m** and $\exists (\bar{f}', \bar{e}') : \text{right}(\bar{e}') = \text{left}(\bar{e}) - 1 \wedge \text{right}(\bar{f}') = \text{left}(\bar{f}) - 1$
  - **o=s** and $\exists (\bar{f}', \bar{e}') : \text{right}(\bar{e}') = \text{left}(\bar{e}) - 1 \wedge \text{left}(\bar{f}') = \text{right}(\bar{f}) + 1$
  - **o=d** and $\exists (\bar{f}', \bar{e}') : \text{right}(\bar{e}') = \text{left}(\bar{e}) - 1$
- $c_{l \to r}(o, (\bar{f}, \bar{e})) + +$ if
  - **o=m** and $\exists (\bar{f}', \bar{e}') : \text{left}(\bar{e}') = \text{right}(\bar{e}) + 1 \wedge \text{left}(\bar{f}') = \text{right}(\bar{f}) + 1$
  - **o=s** and $\exists (\bar{f}', \bar{e}') : \text{left}(\bar{e}') = \text{right}(\bar{e}) - 1 \wedge \text{right}(\bar{f}') = \text{left}(\bar{f}) - 1$
  - **o=d** and $\exists (\bar{f}', \bar{e}') : \text{left}(\bar{e}') = \text{right}(\bar{e}) + 1$

Smoothing LRMs

- Using maximum likelihood estimation

$$p_{r\to l}(o|(\bar{f},\bar{e})) = \frac{c_{r\to l}(o,(\bar{f},\bar{e}))}{\sum_{o\in O} c_{r\to l}(o,(\bar{f},\bar{e}))}$$

  leads to overfitting on the training data

- Therefore simple additive smoothing is commonly applied:

$$p_{r\to l}(o|(\bar{f},\bar{e})) = \frac{\sigma p_{r\to l}(o)+c_{r\to l}(o,(\bar{f},\bar{e}))}{\sigma+\sum_{o\in O} c_{r\to l}(o,(\bar{f},\bar{e}))}$$

  where σ is a small constant (e.g., 0.5)

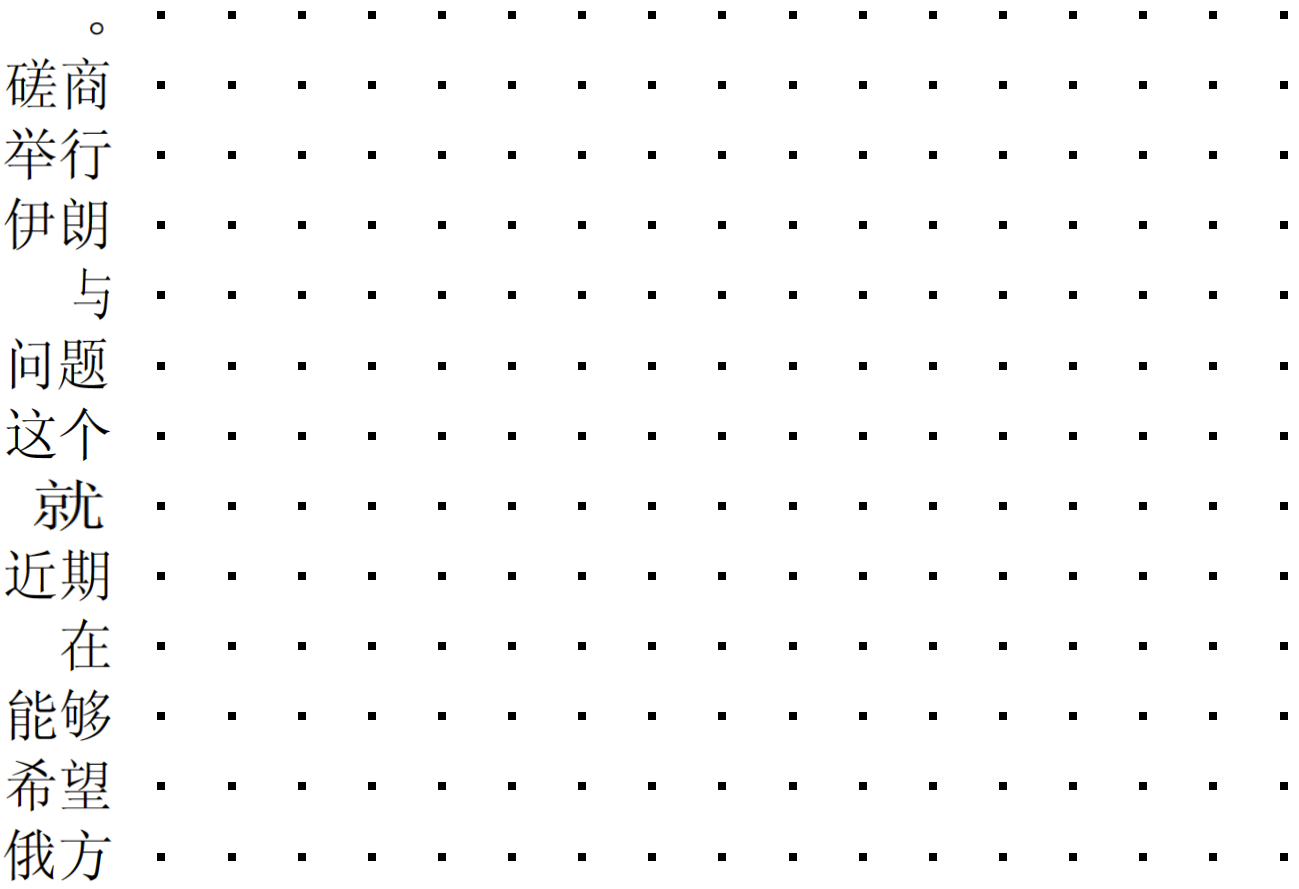
$$p_{r\to l}(o) = \frac{\sum_{(\bar{f},\bar{e})} c_{r\to l}(o,(\bar{f},\bar{e}))}{\sum_{o\in O}\sum_{(\bar{f},\bar{e})} c_{r\to l}(o,(\bar{f},\bar{e}))}$$

- The smoothed variant of $p_{l\to r}(o|(\bar{f},\bar{e}))$ is defined analoguously

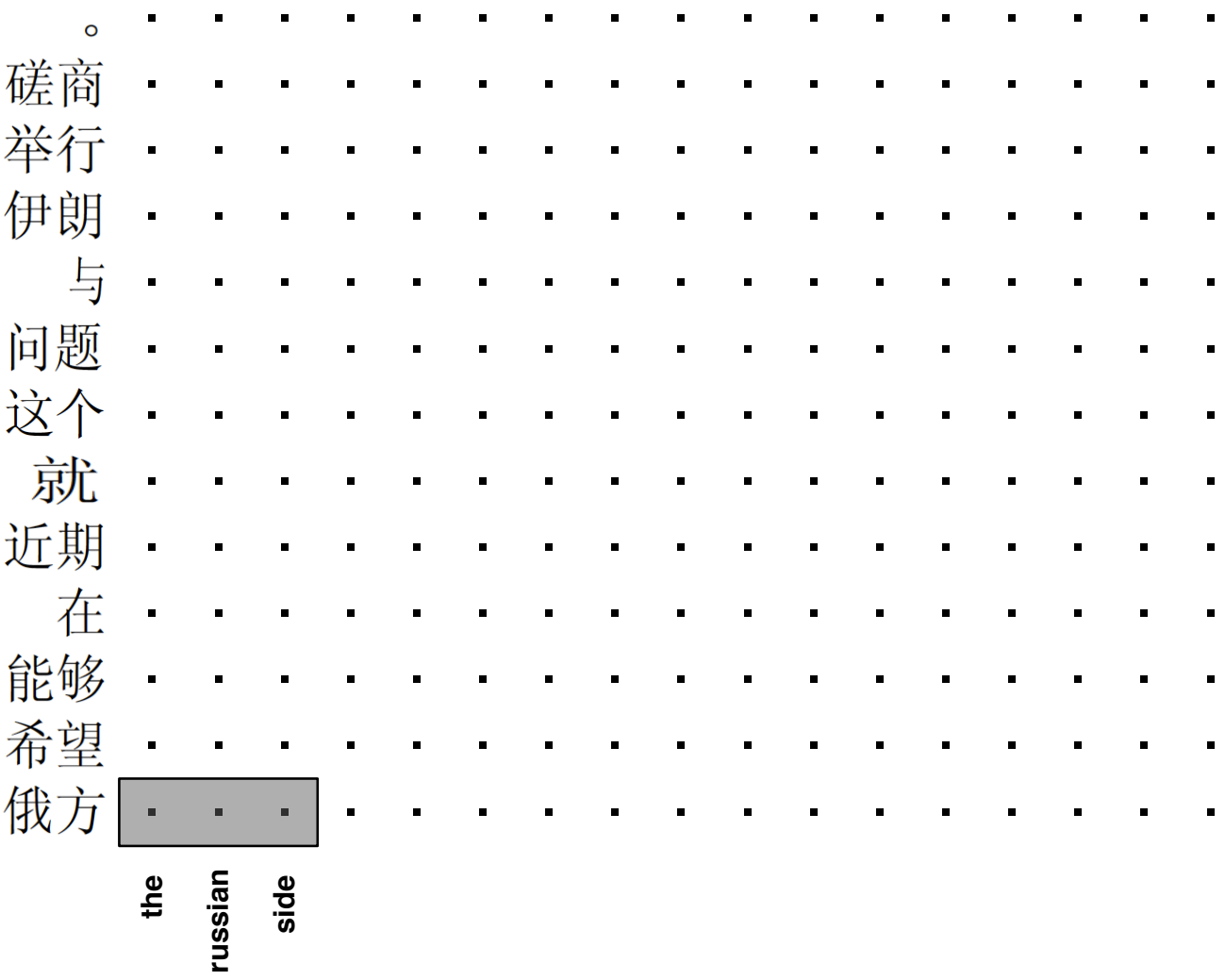
Using LRMs During Decoding

- The phrase-based estimation allows for consider multiple extractions, and therefore multiple orientations, wrt the previous $(r \rightarrow l)$ and next phrases $((l \rightarrow r)$
  - This is not the case during decoding!
- During decoding the $r \rightarrow l$ orientations are computed with the respect to the one phrase pair applied in the previous state
  - Even though an alternative derivation might have lead to different phrase applications
- Note that an additional constraint has to be respected for two states s_i and s_j to be recombinable
  - the current phrase pair of s_i and s_j has to be identical
  - Can be weakened to: $\forall_{o \in O} : p_{l \rightarrow r}(o|(\bar{f}_j, \bar{e}_j)) = p_{l \rightarrow r}(o|(\bar{f}_i, \bar{e}_i))$

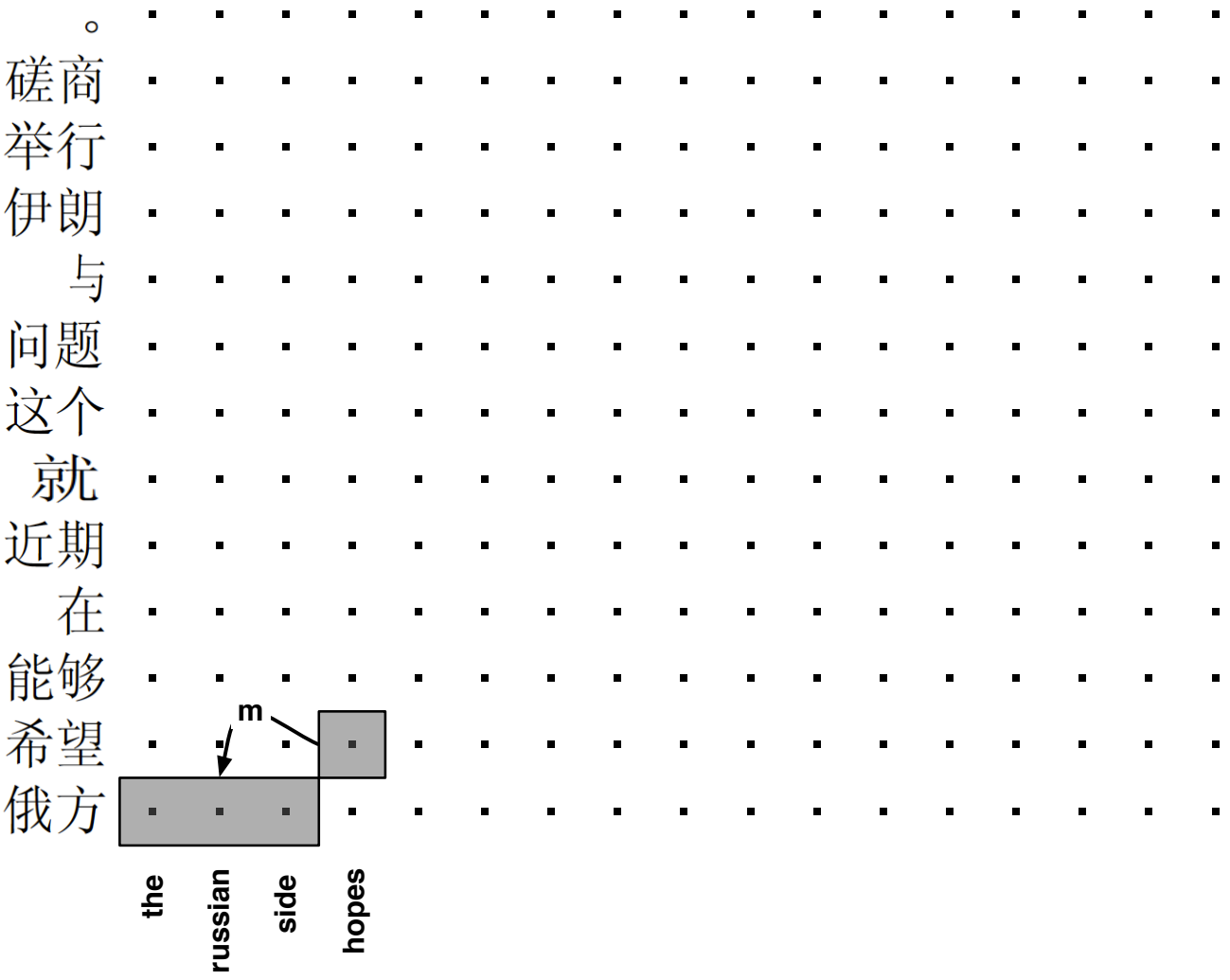
Hierarchical Reordering



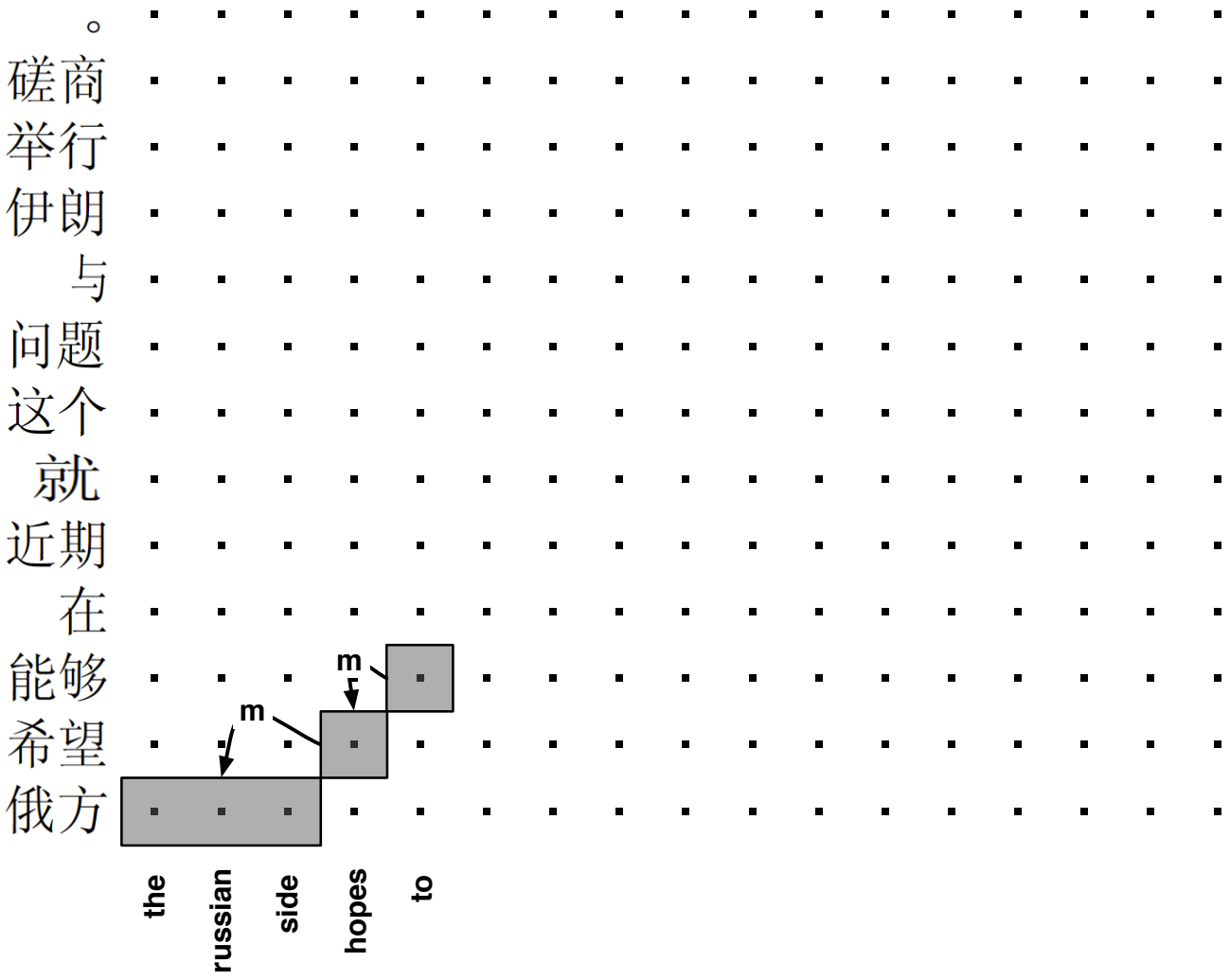
Hierarchical Reordering



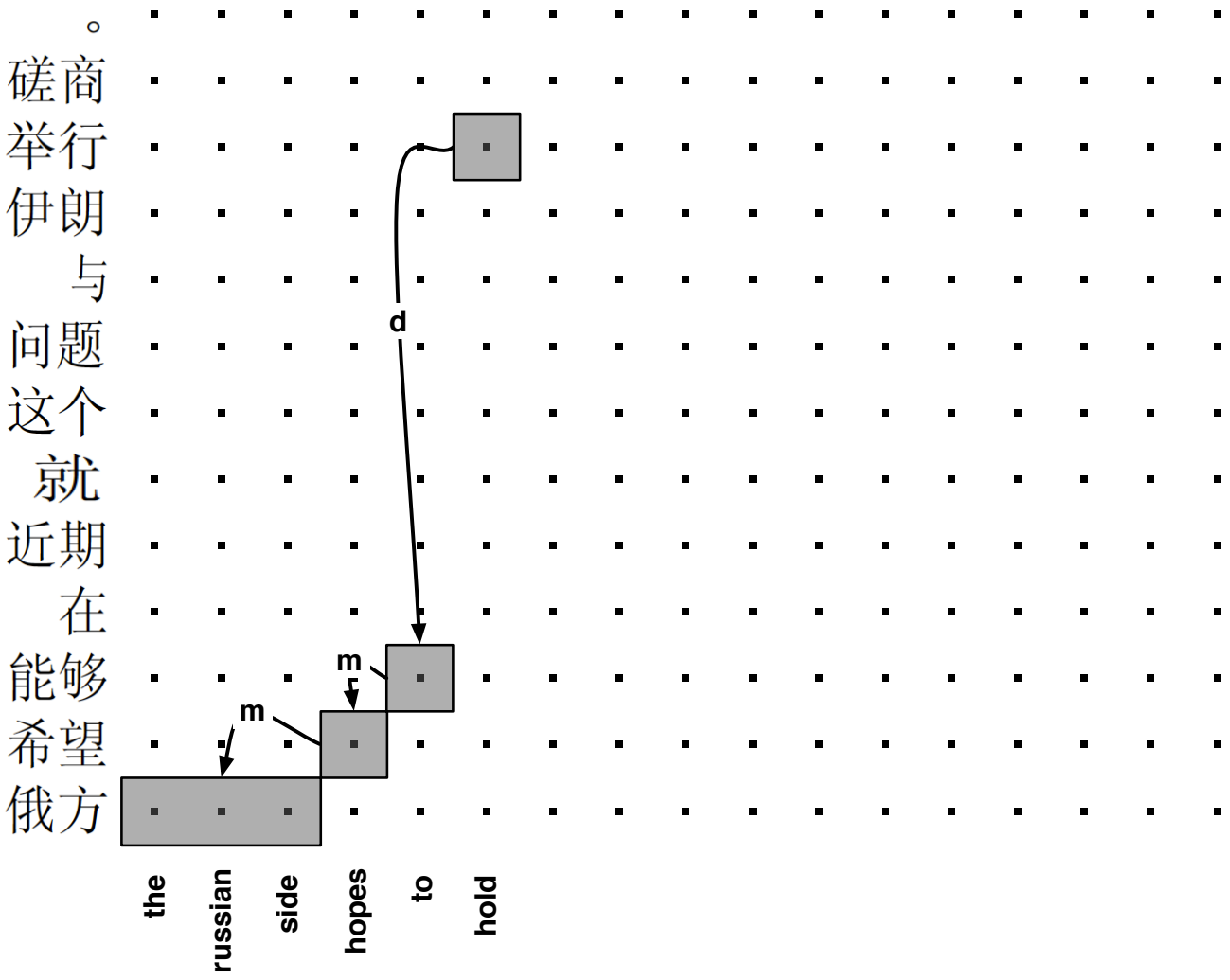
Hierarchical Reordering



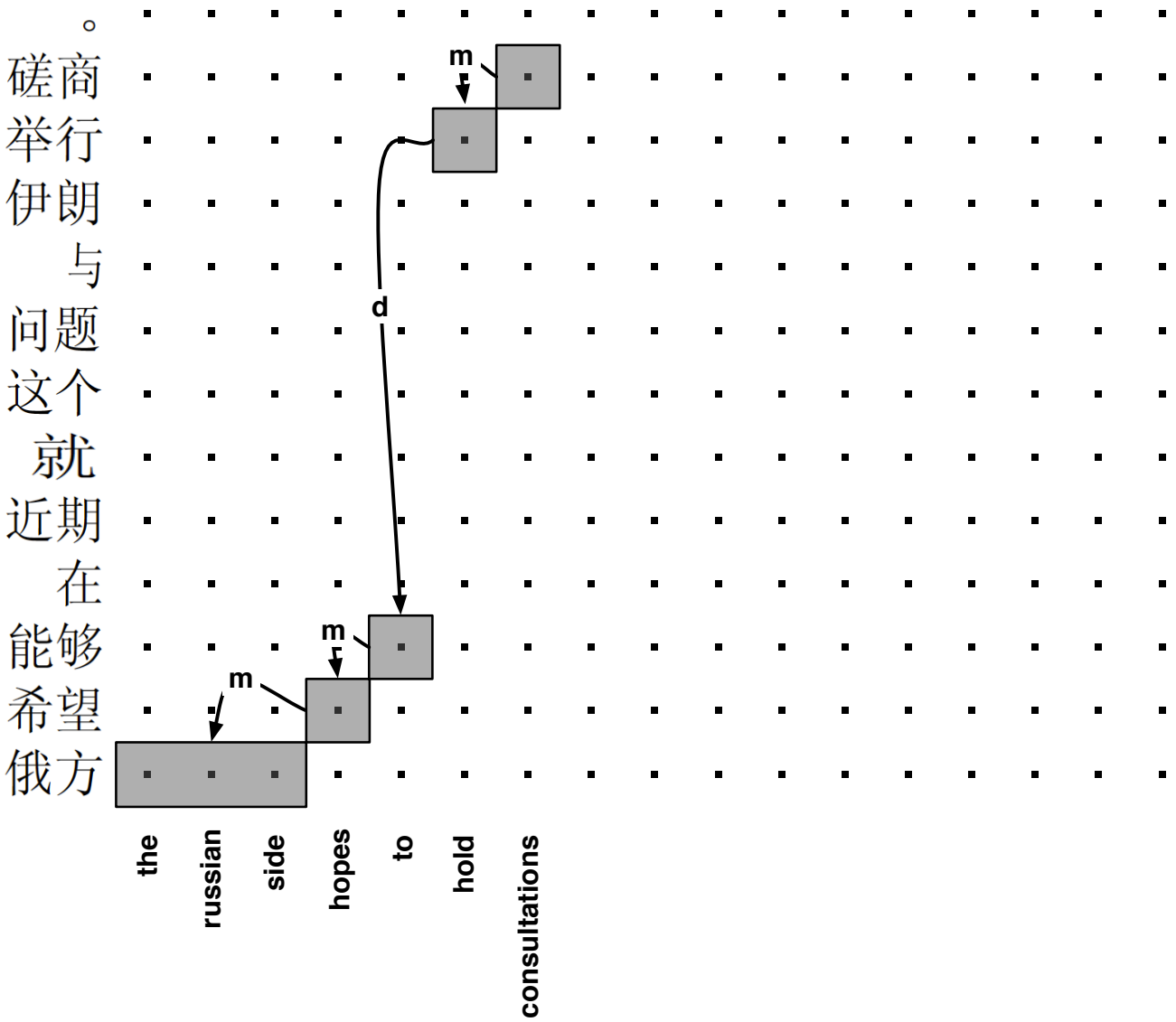
Hierarchical Reordering



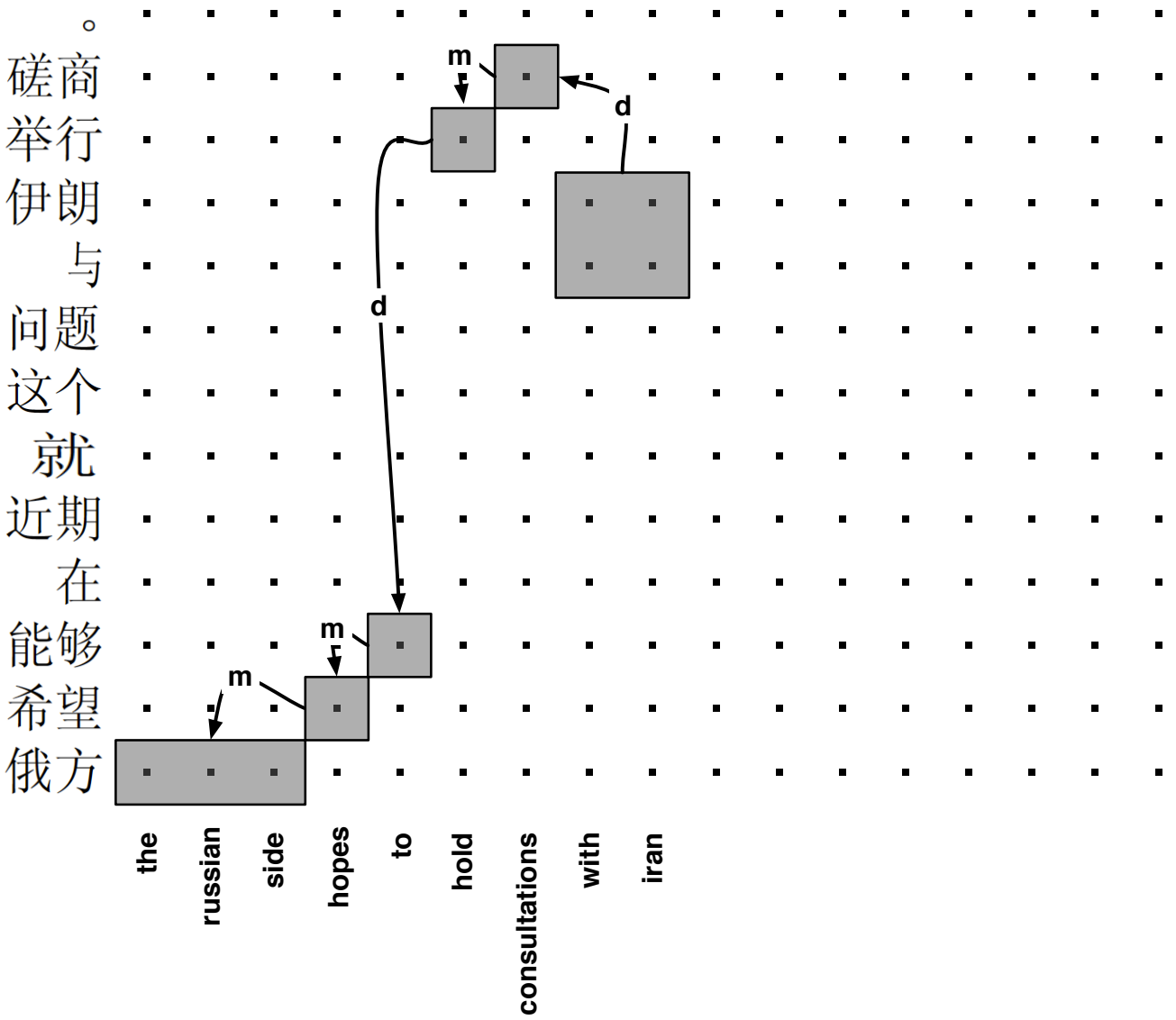
Hierarchical Reordering



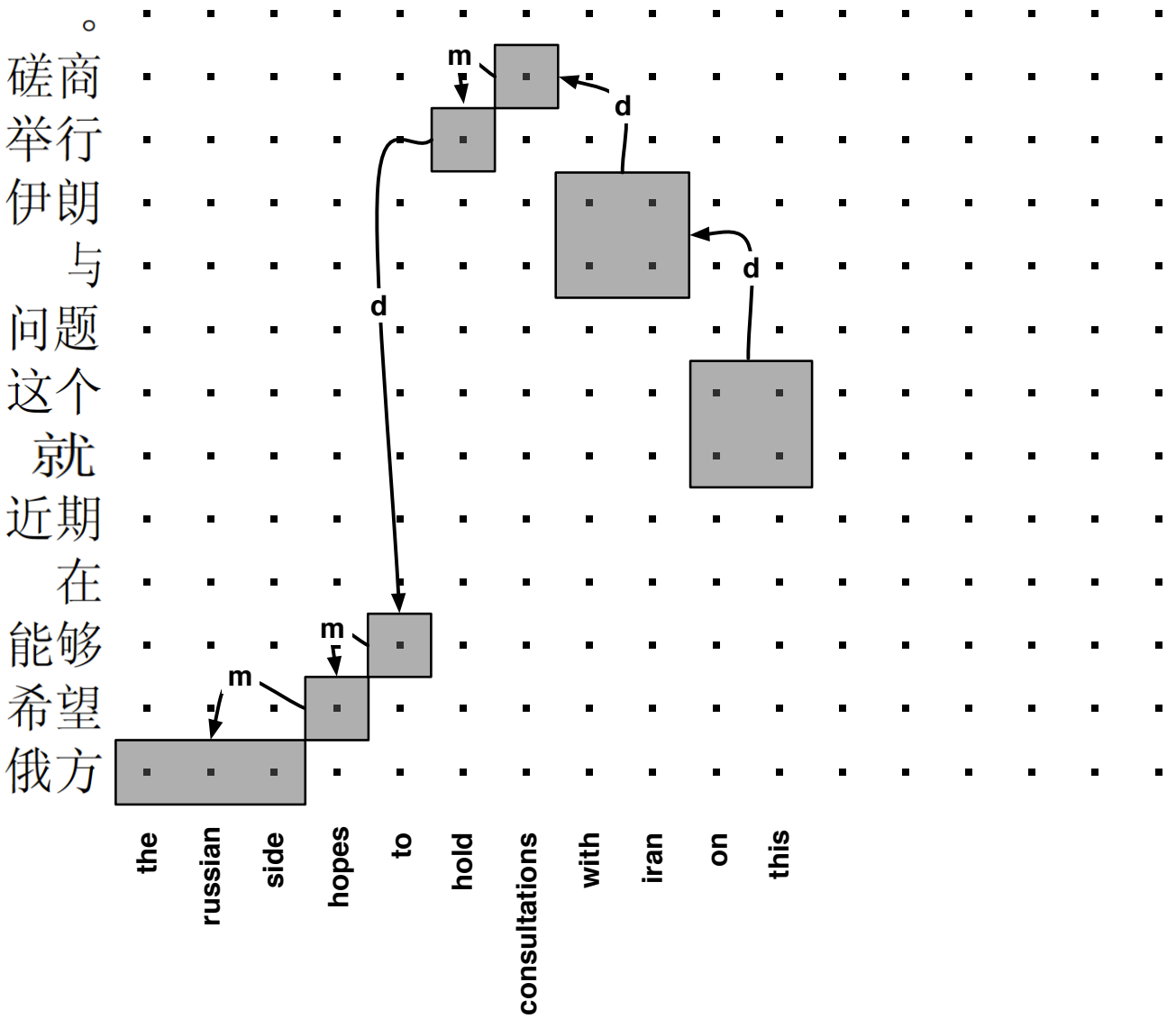
Hierarchical Reordering



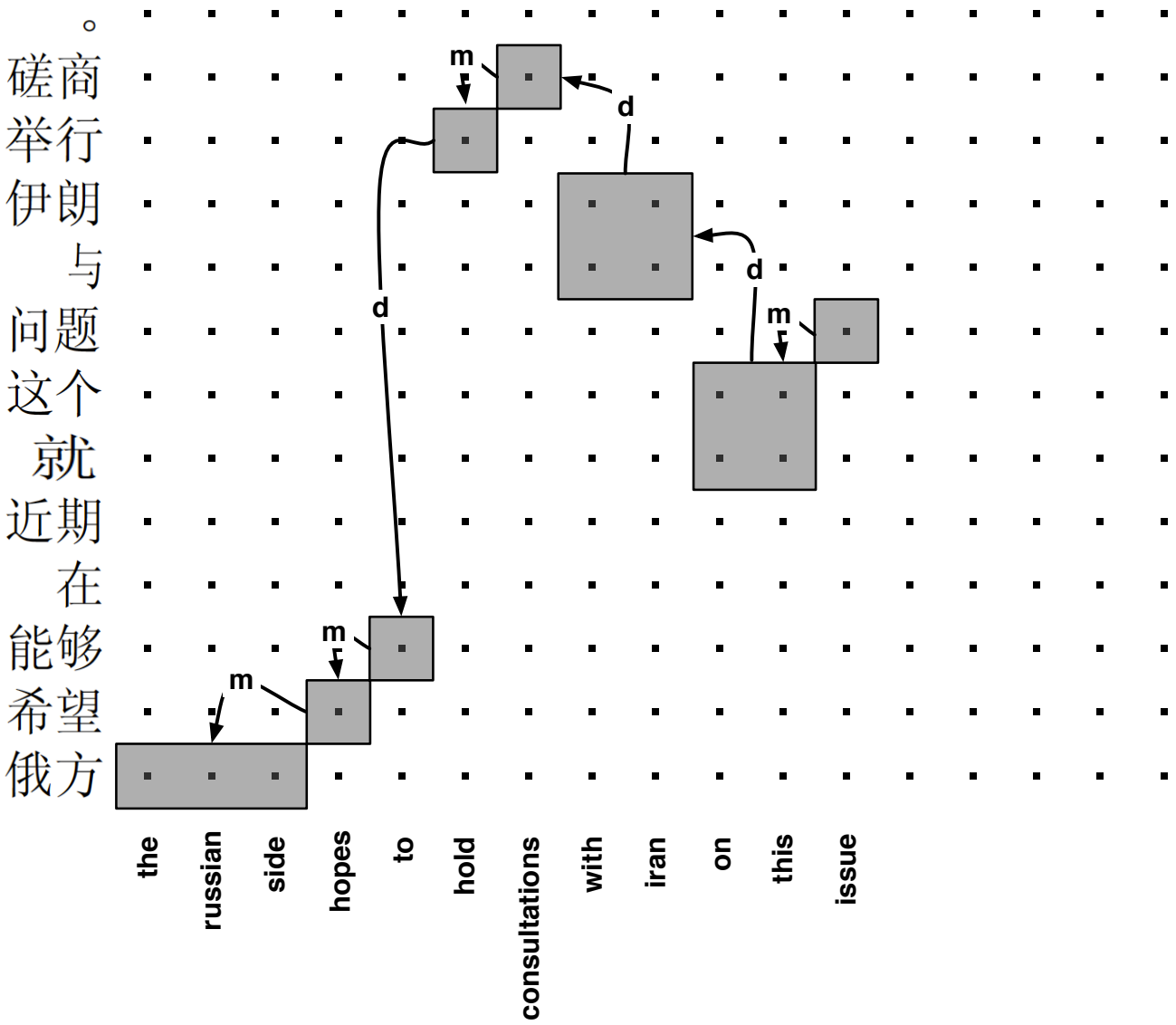
Hierarchical Reordering



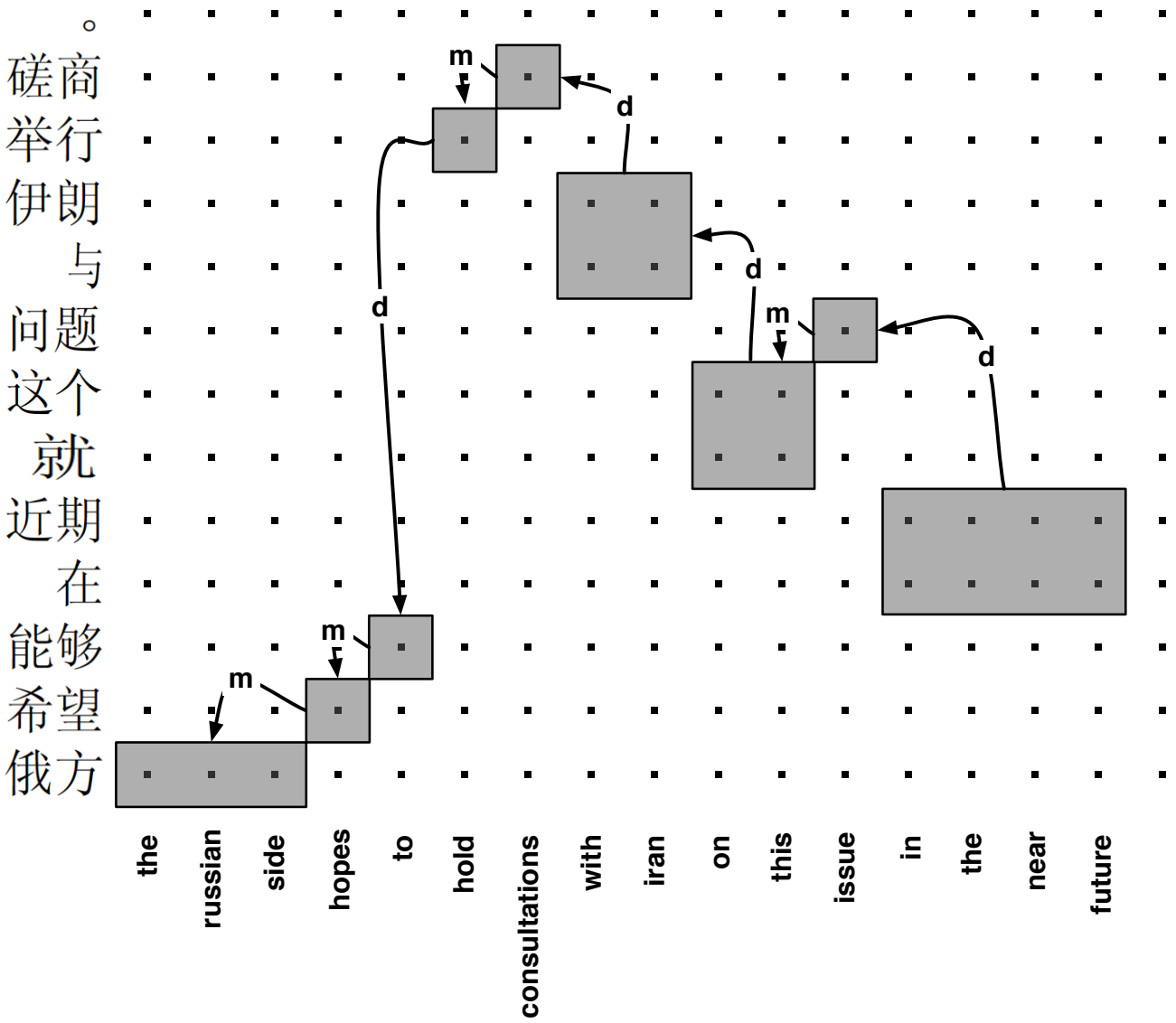
Hierarchical Reordering



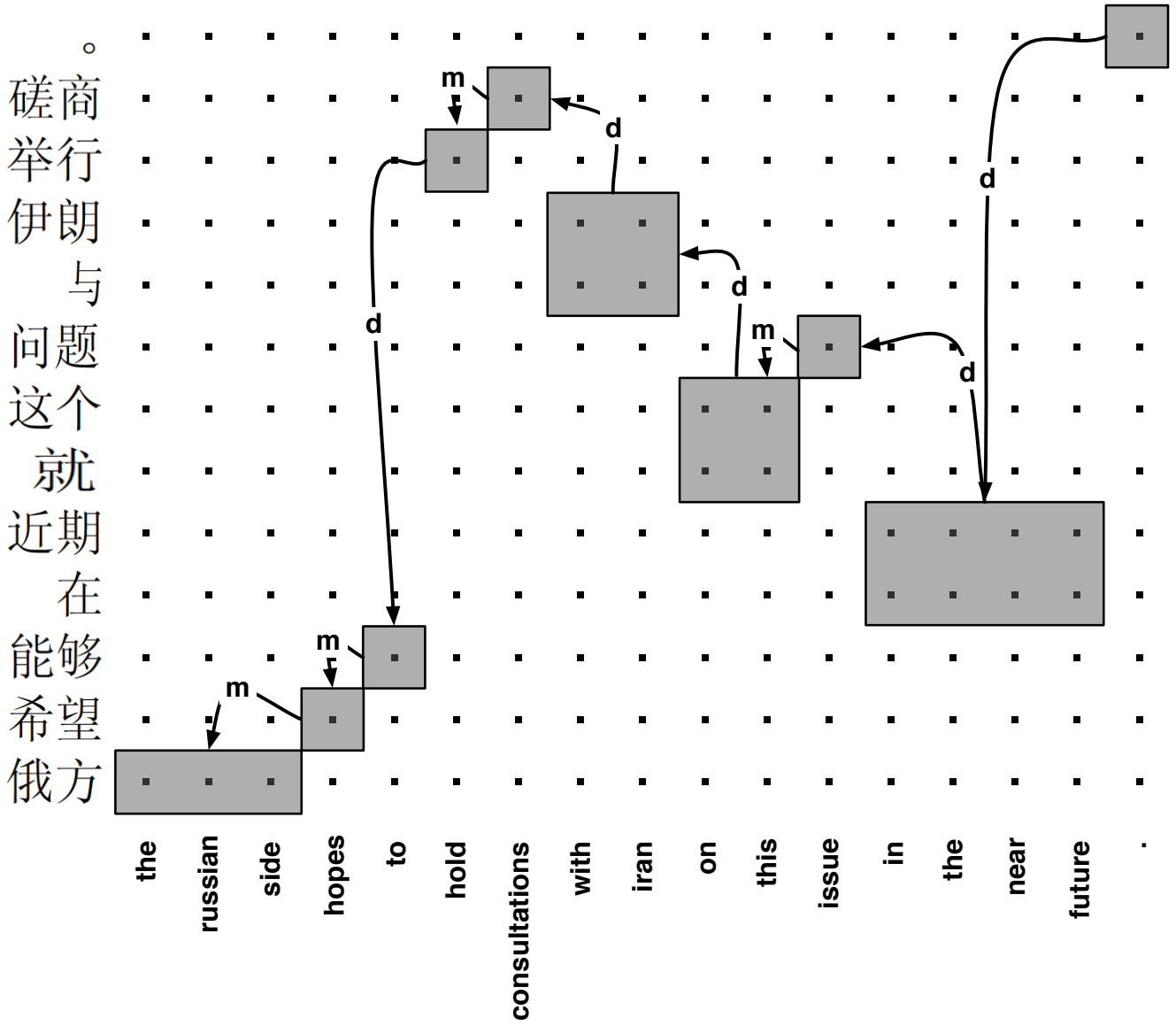
Hierarchical Reordering



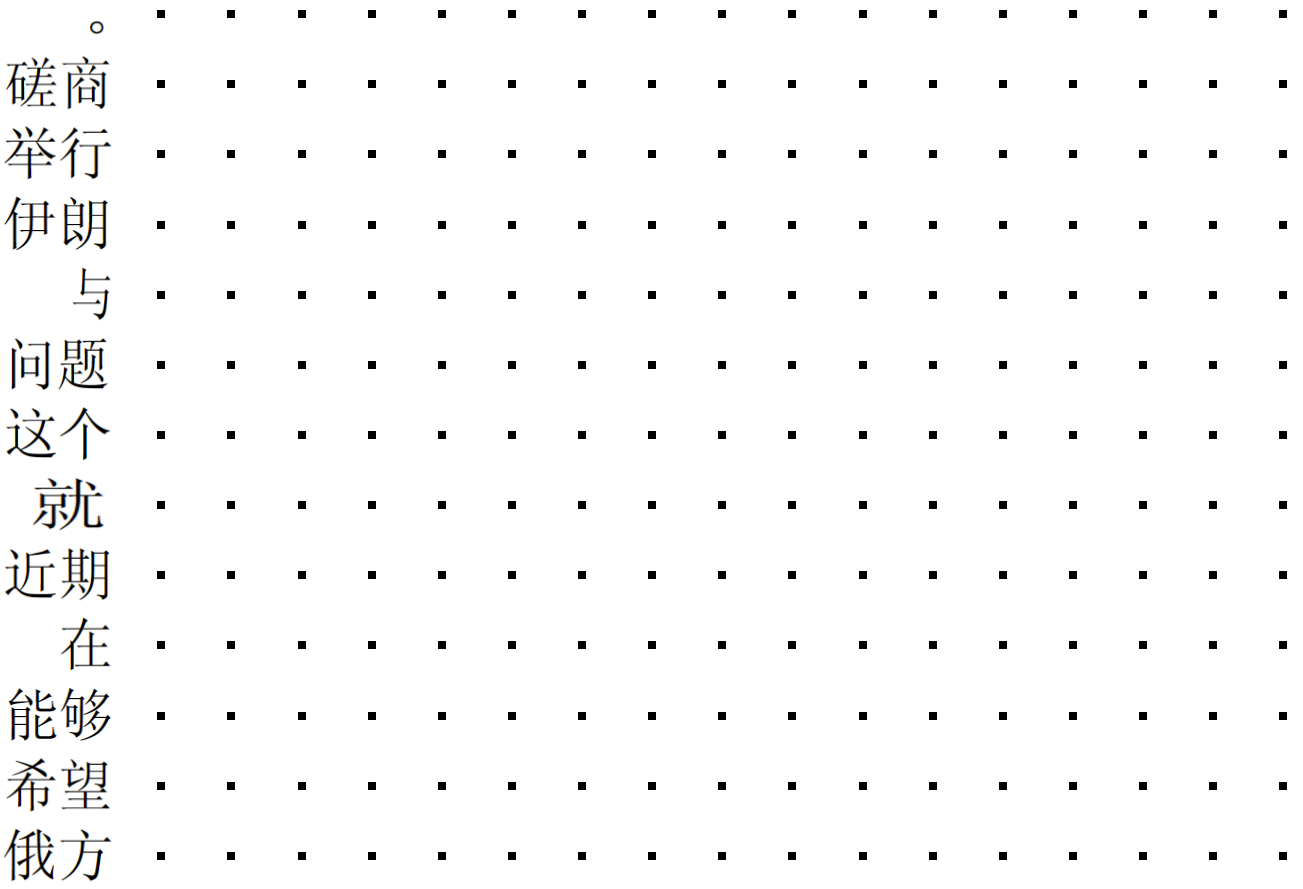
Hierarchical Reordering



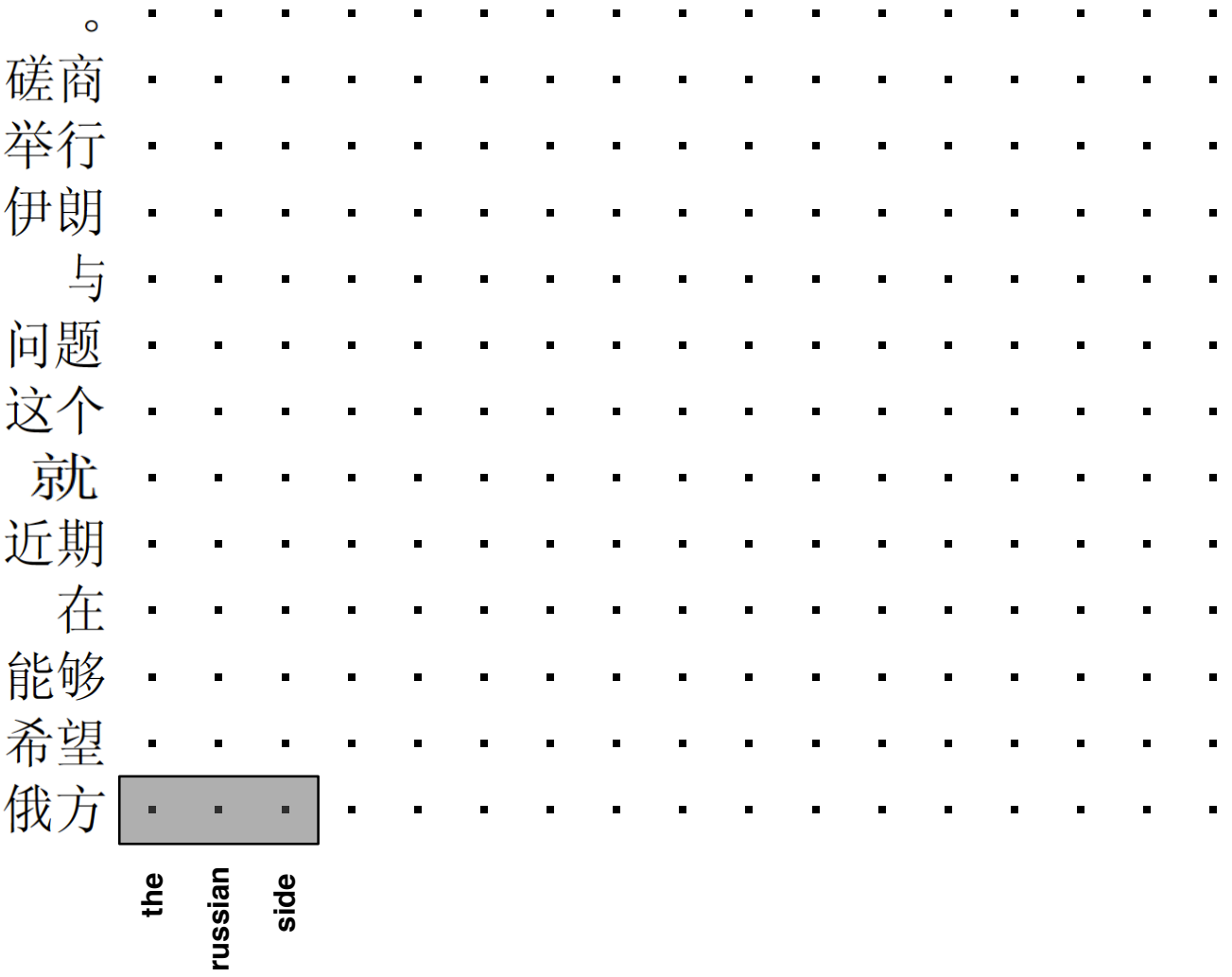
Hierarchical Reordering



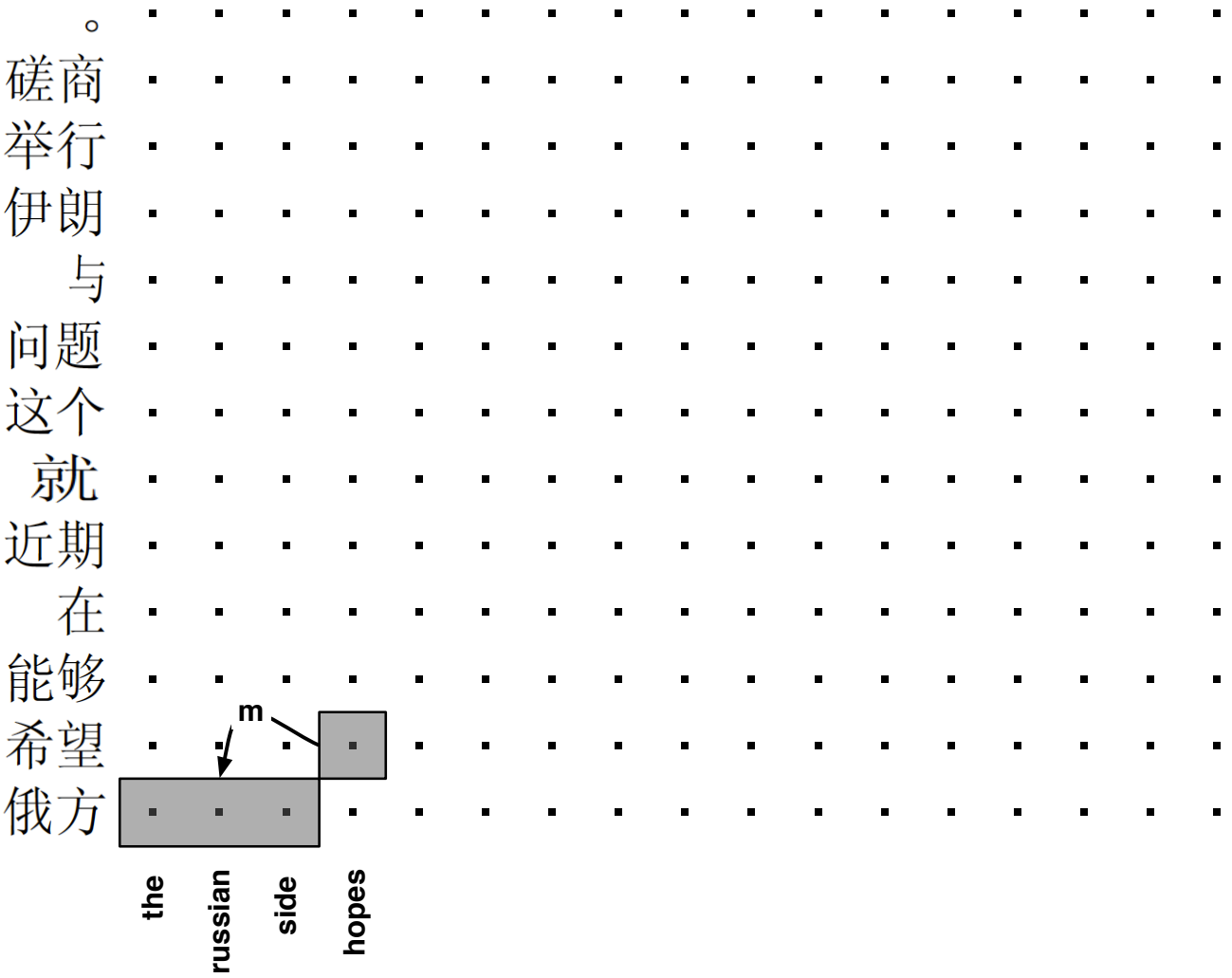
Hierarchical Reordering



Hierarchical Reordering

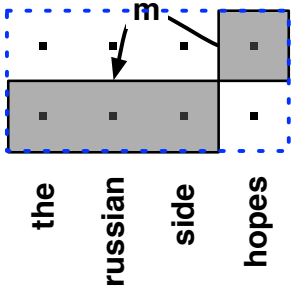


Hierarchical Reordering

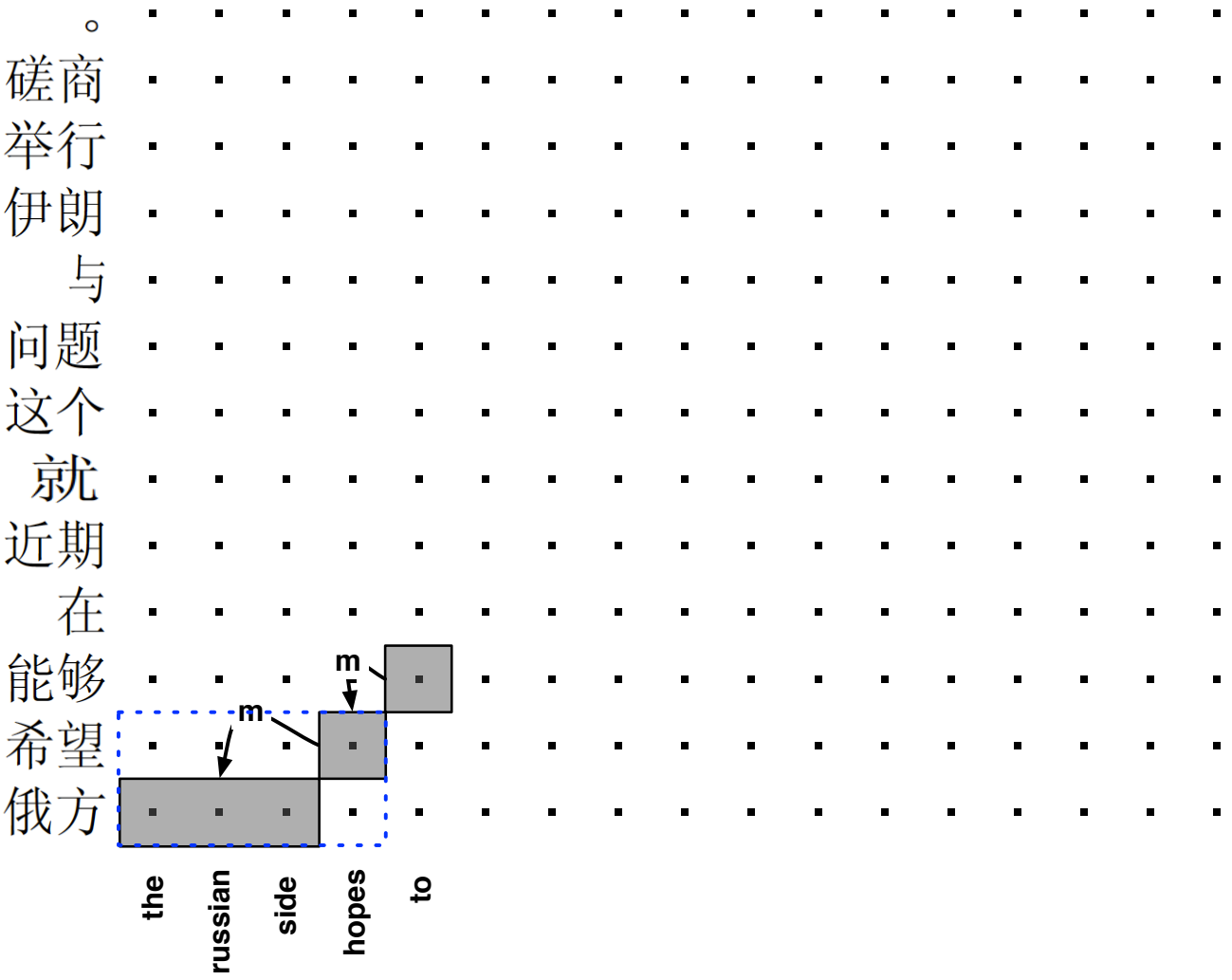


Hierarchical Reordering

。
磋商
举行
伊朗
与
问题
这个
就
近期
在
能够
希望
俄方

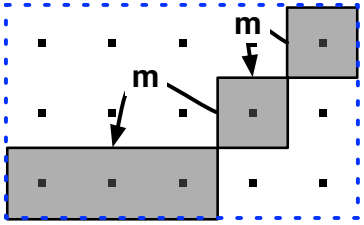


Hierarchical Reordering



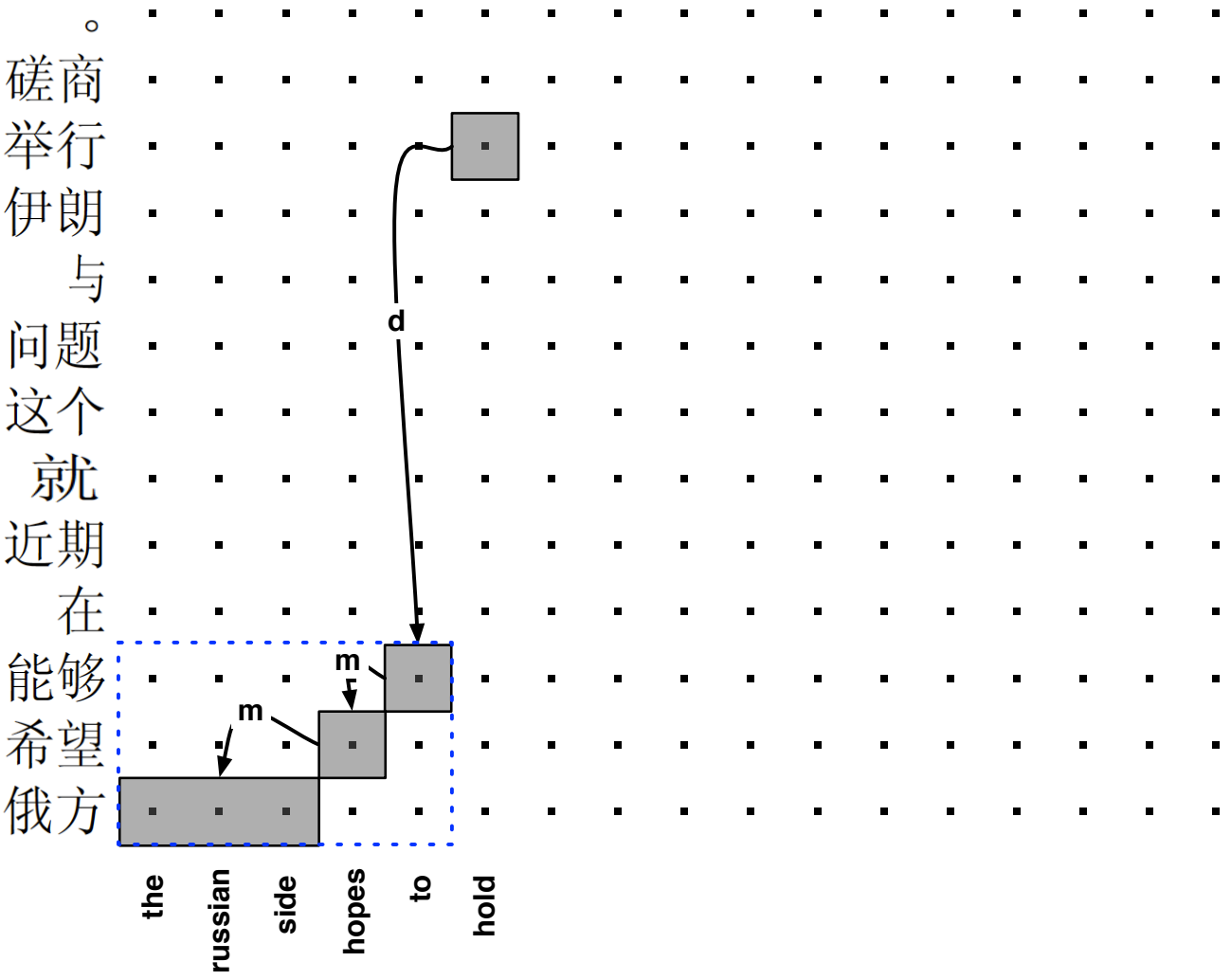
Hierarchical Reordering

。
磋商
举行
伊朗
与
问题
这个
就
近期
在
能够
希望
俄方

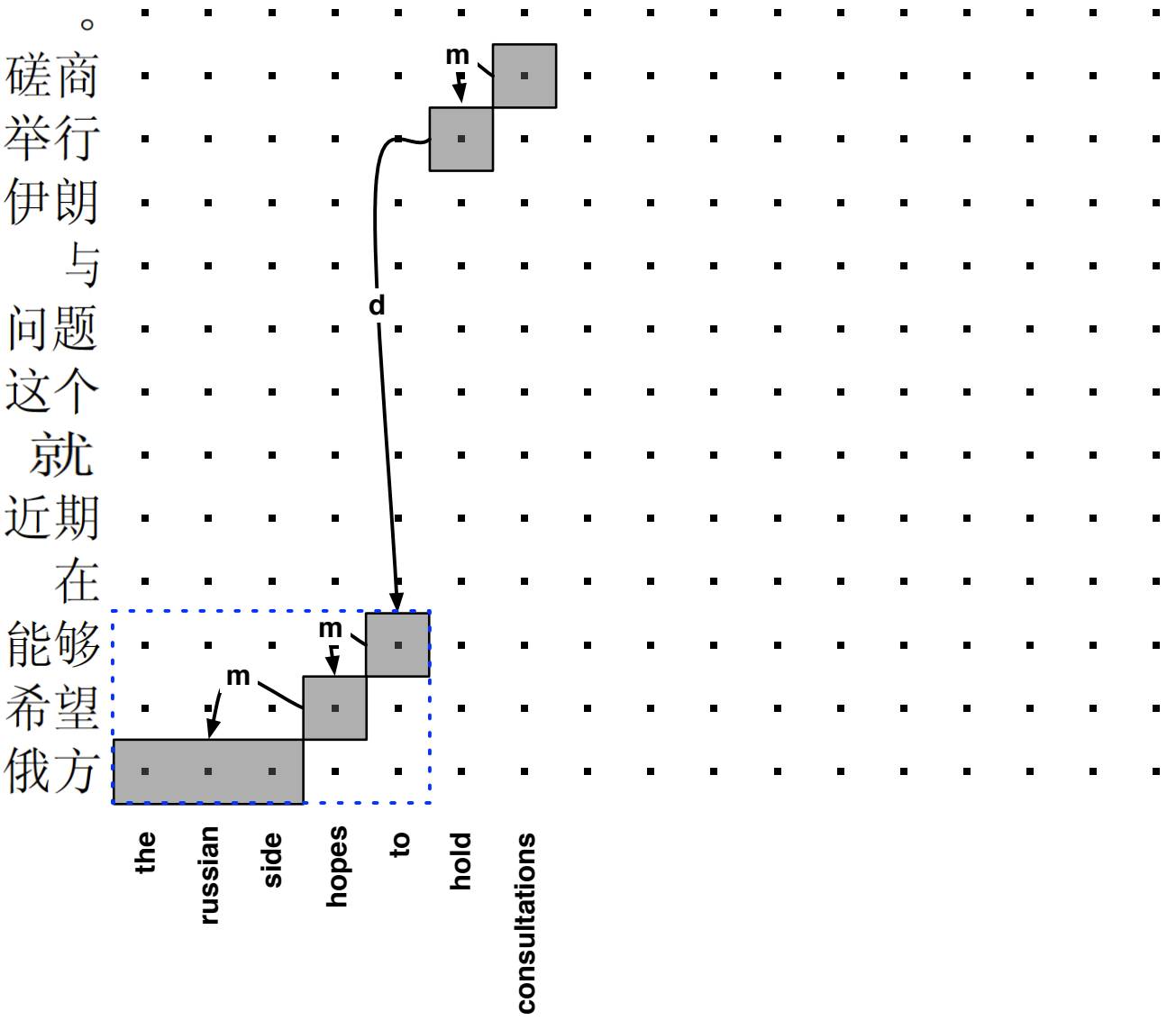


the russian side hopes to

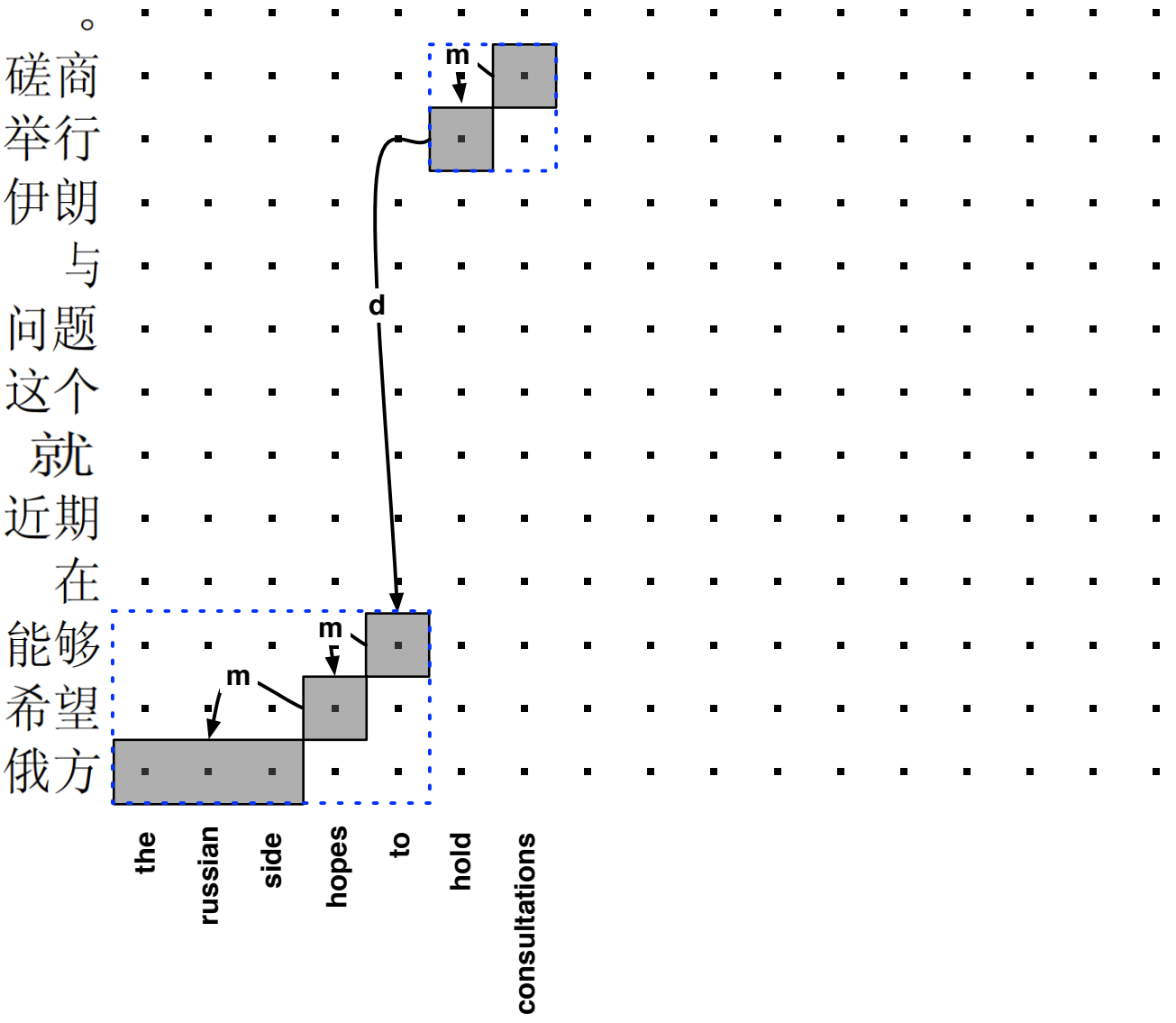
Hierarchical Reordering



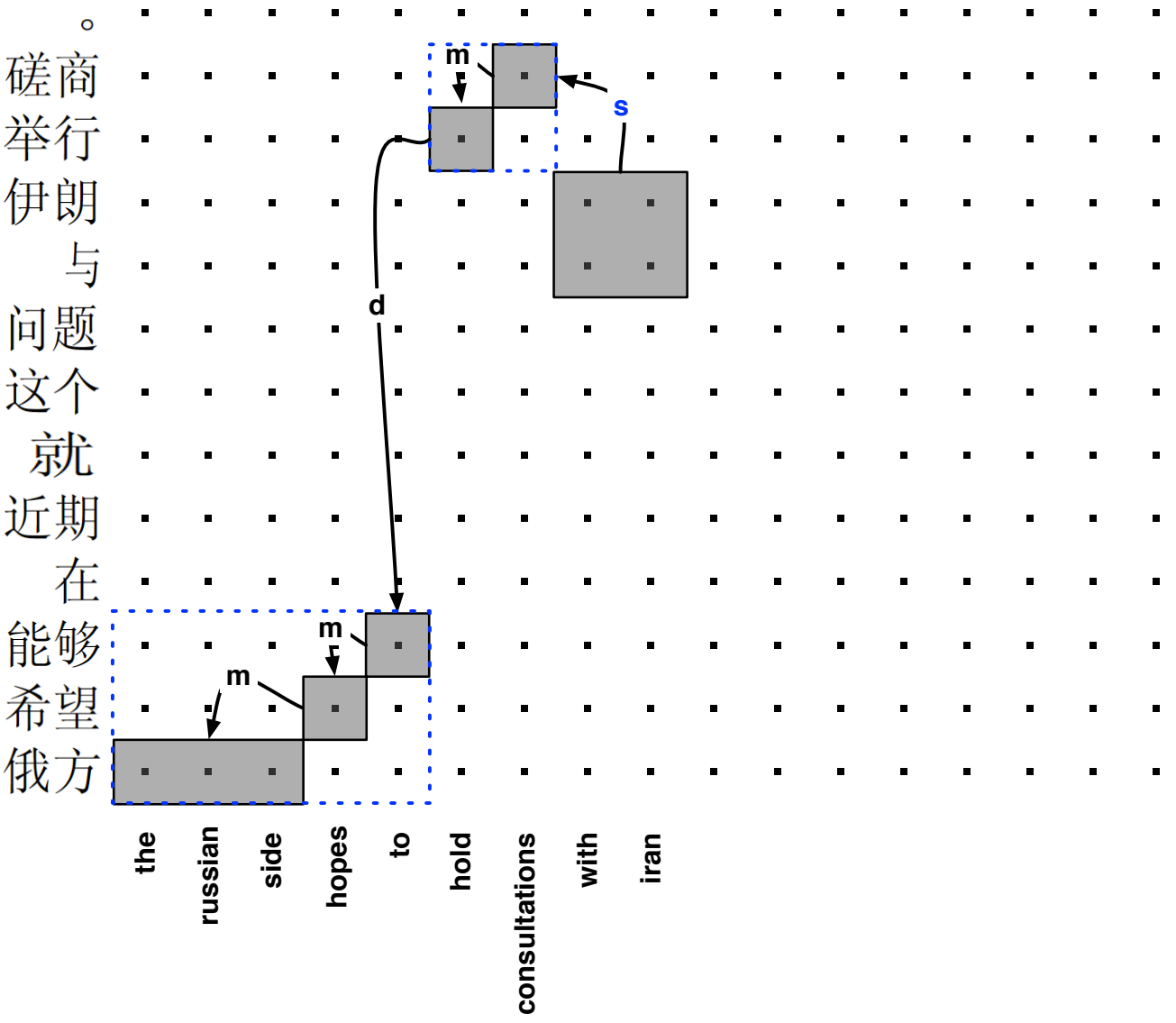
Hierarchical Reordering



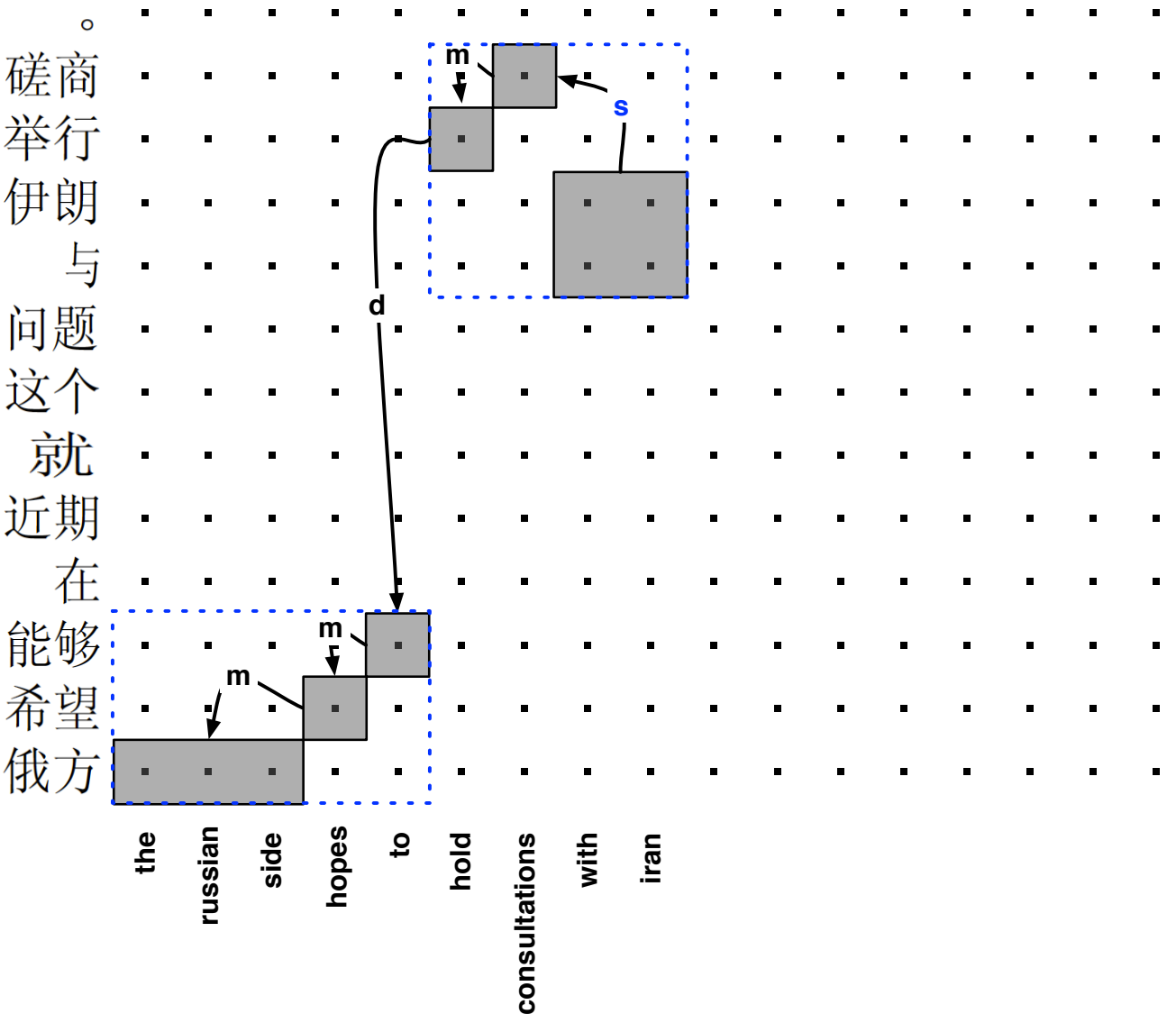
Hierarchical Reordering



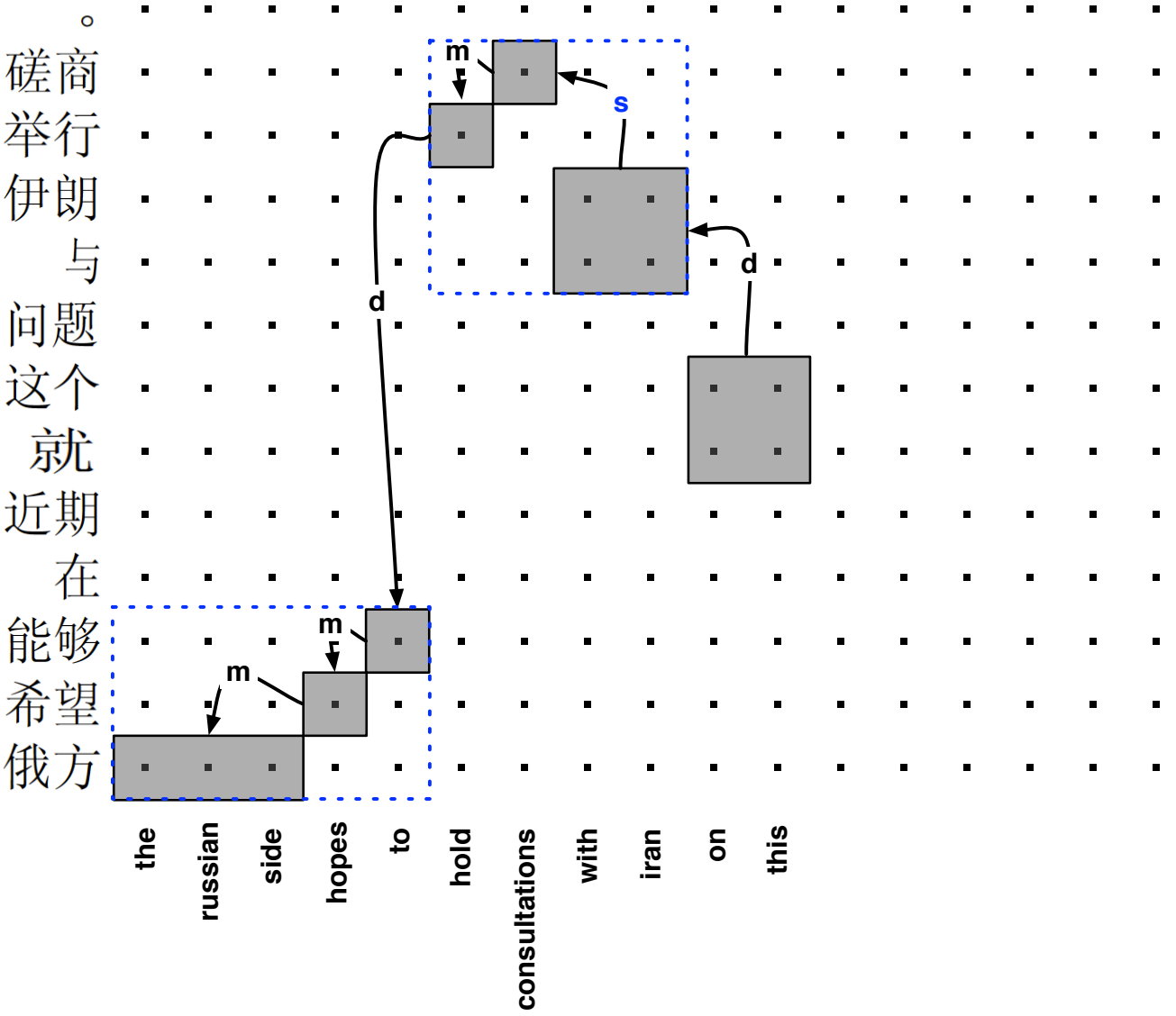
Hierarchical Reordering



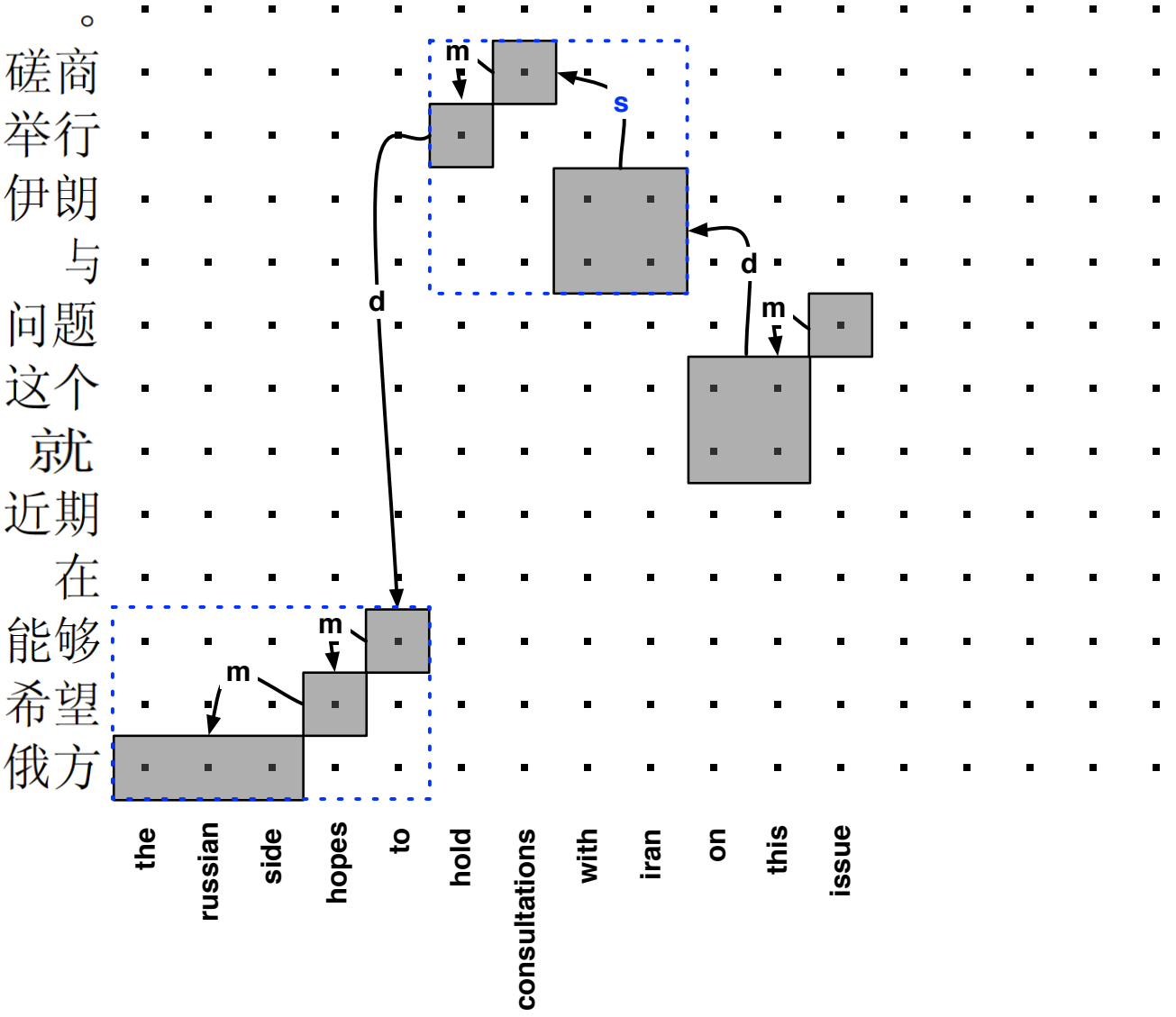
Hierarchical Reordering



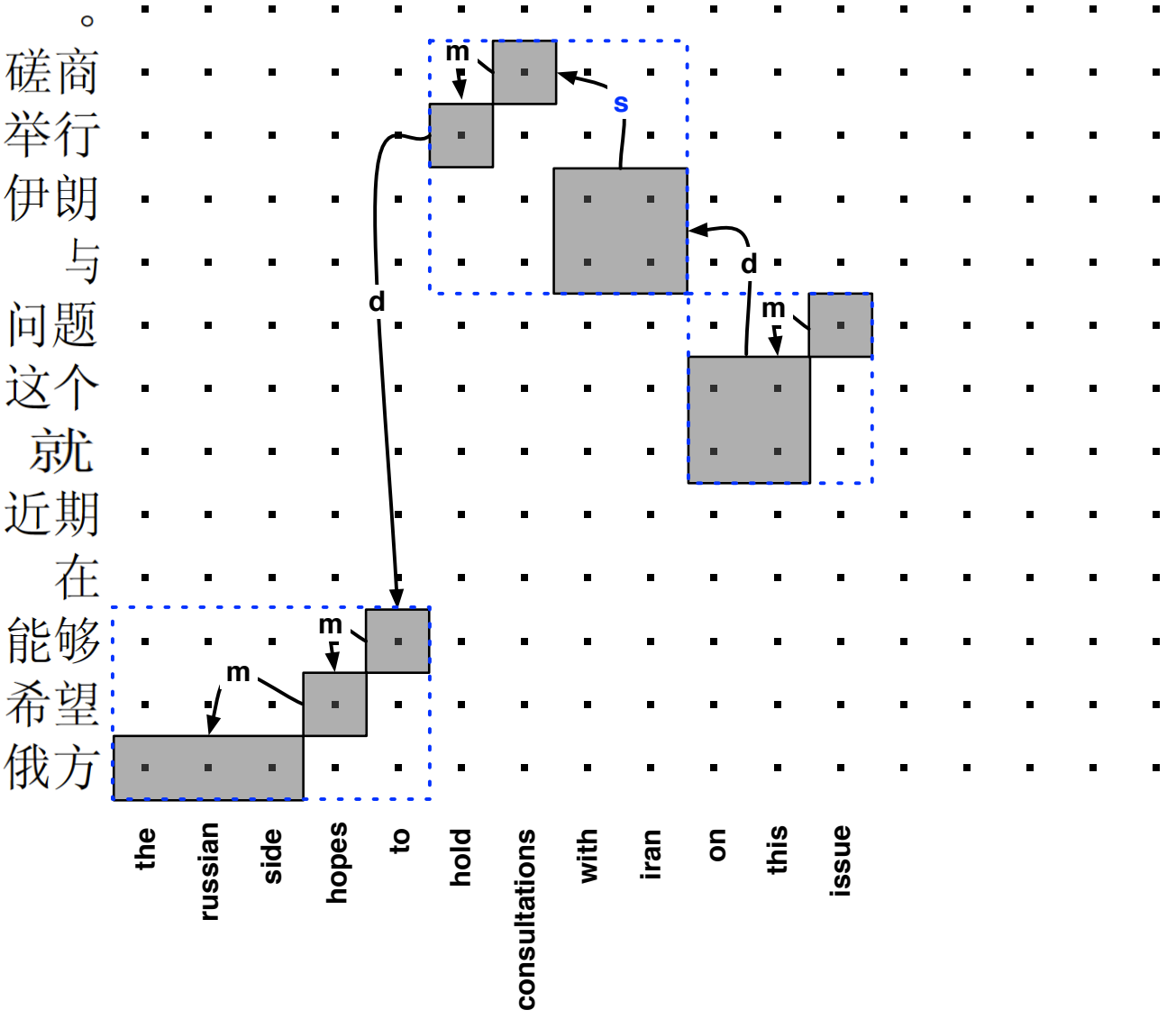
Hierarchical Reordering



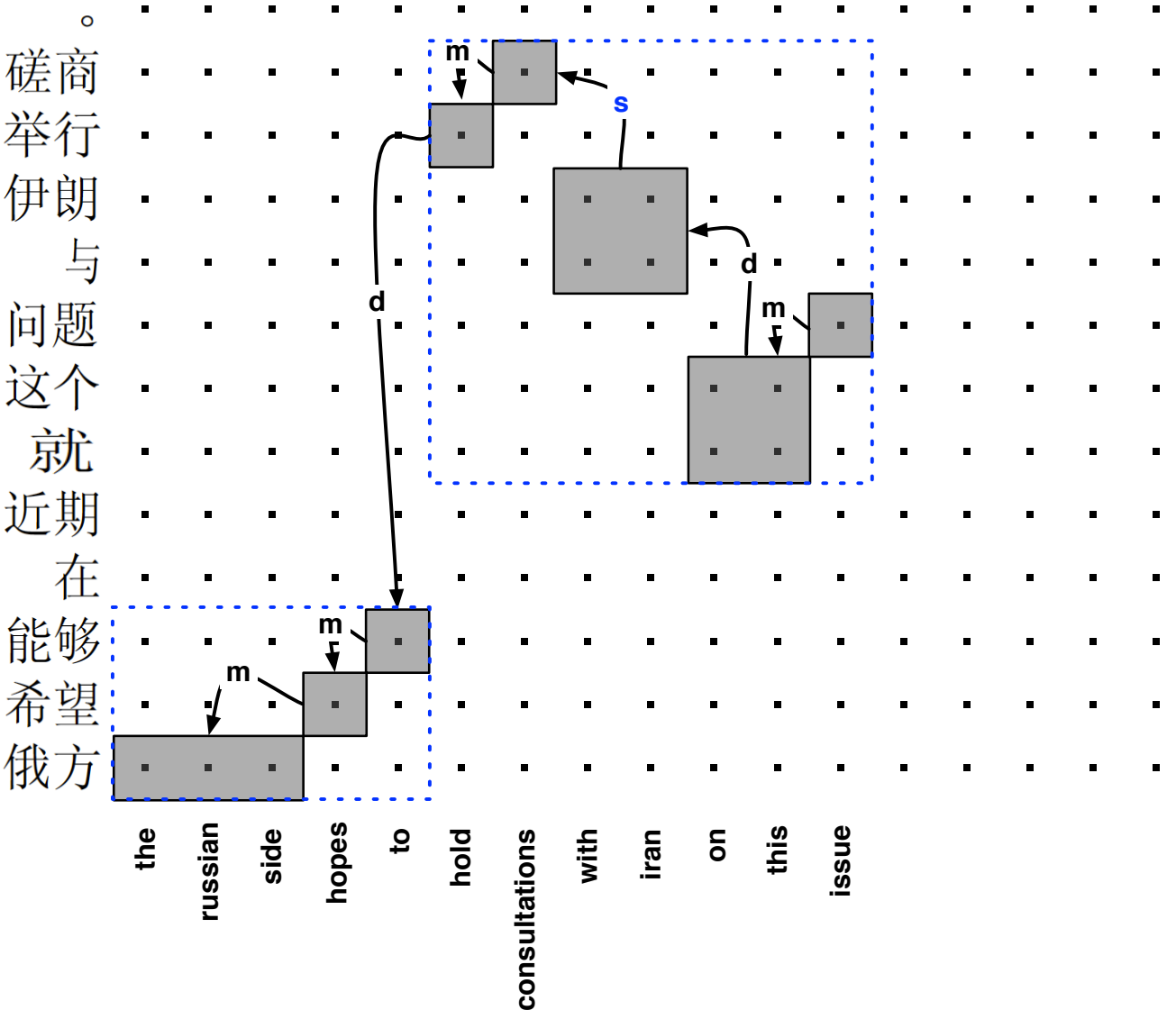
Hierarchical Reordering



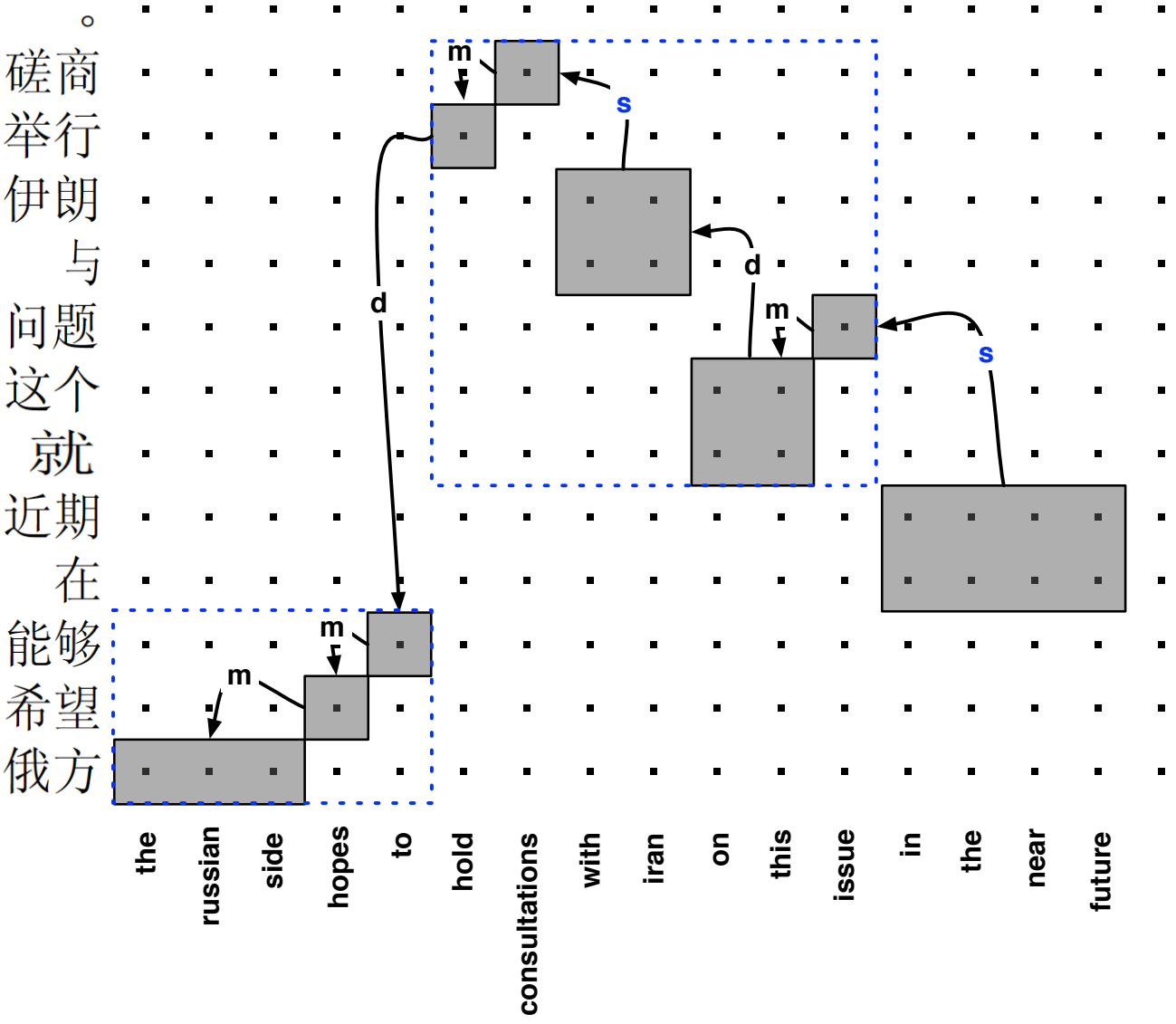
Hierarchical Reordering



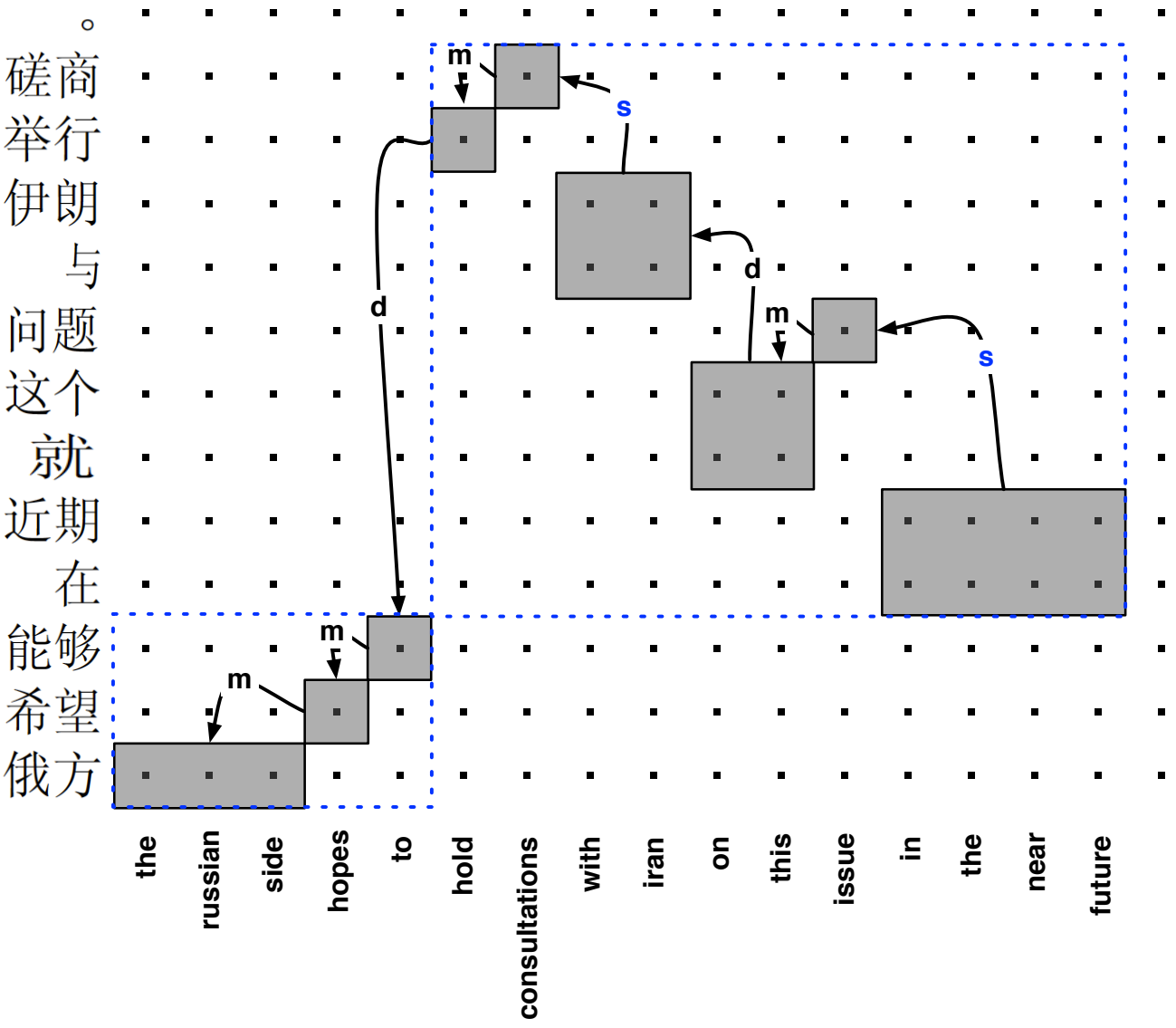
Hierarchical Reordering



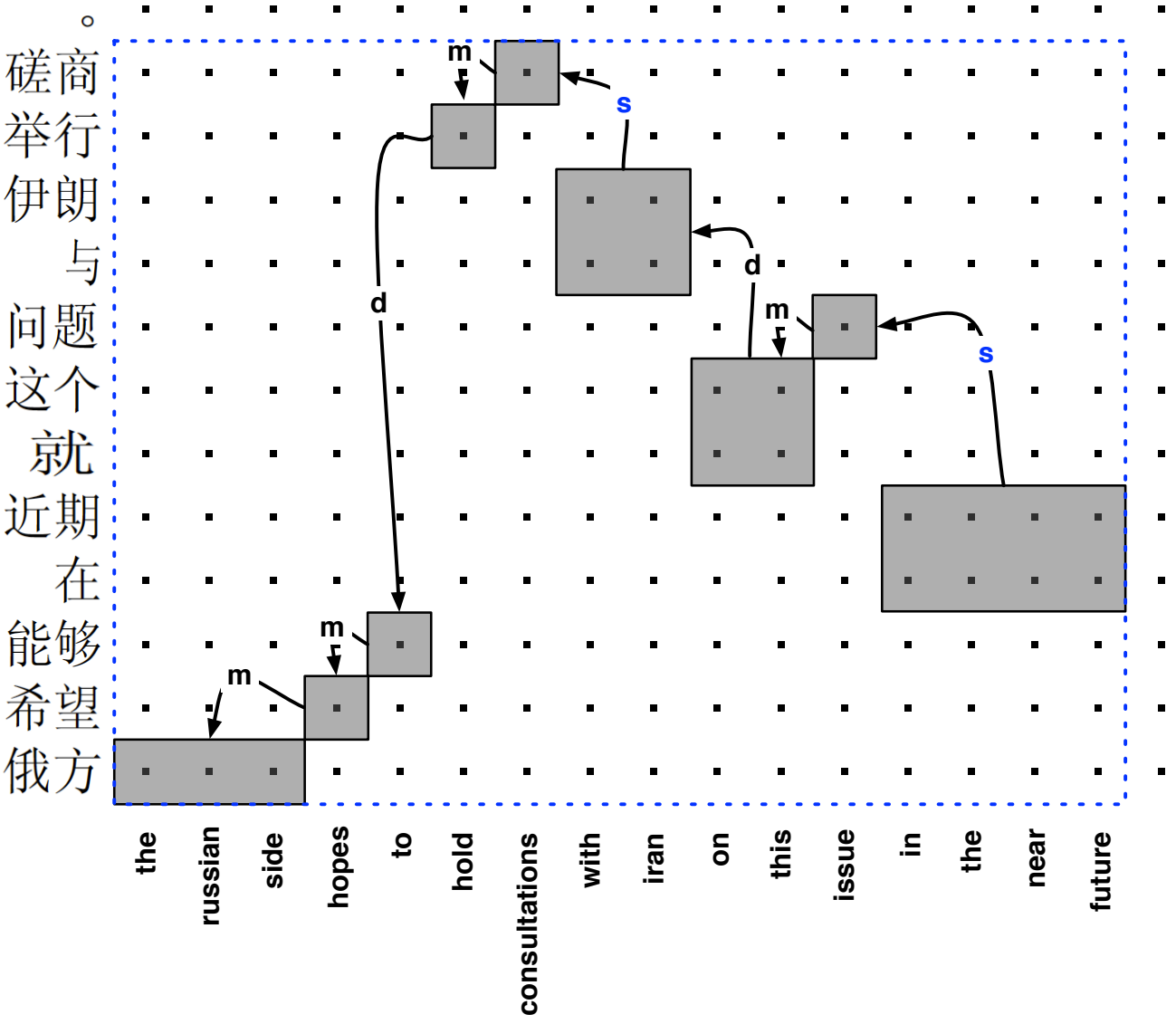
Hierarchical Reordering



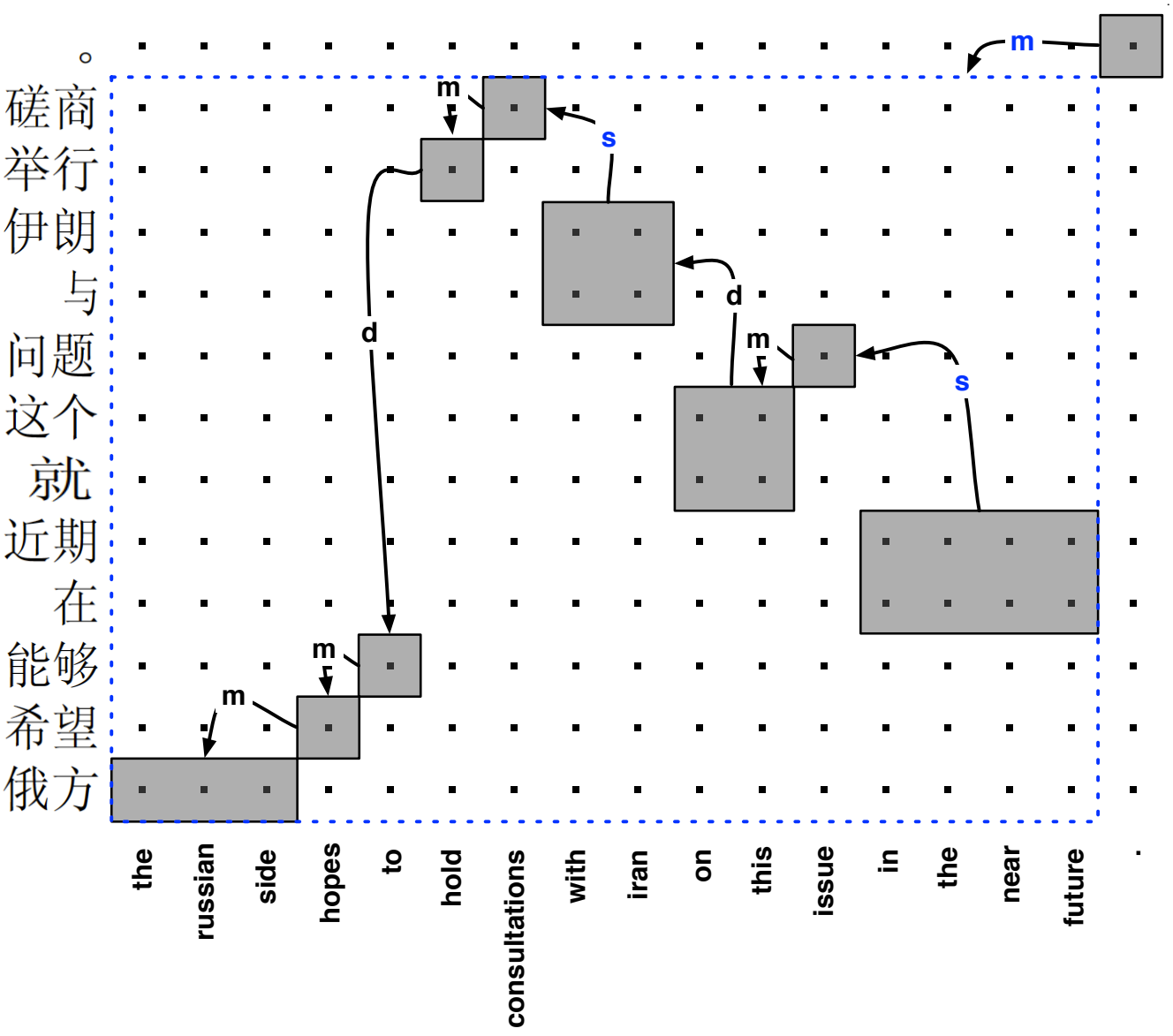
Hierarchical Reordering



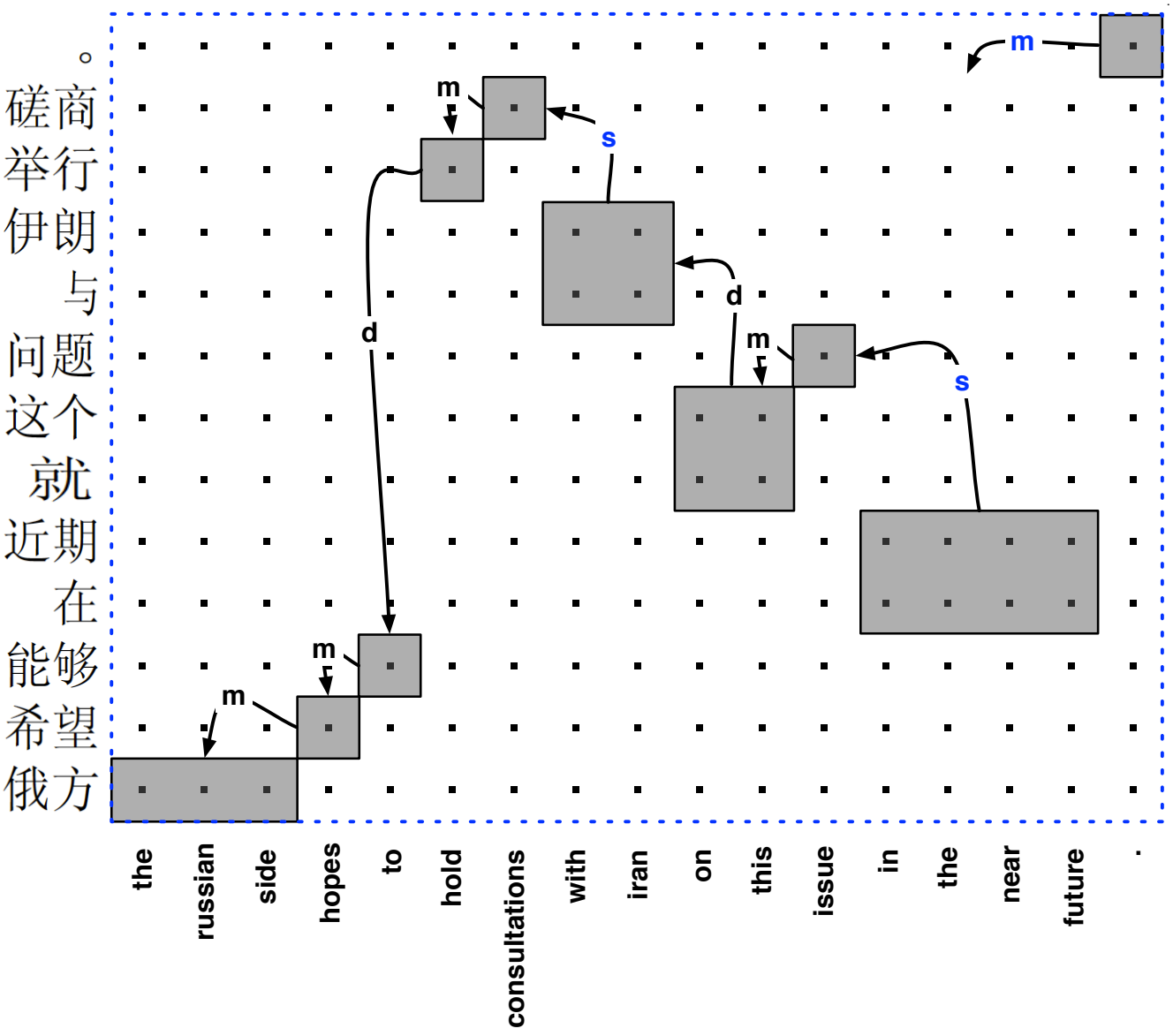
Hierarchical Reordering



Hierarchical Reordering



Hierarchical Reordering



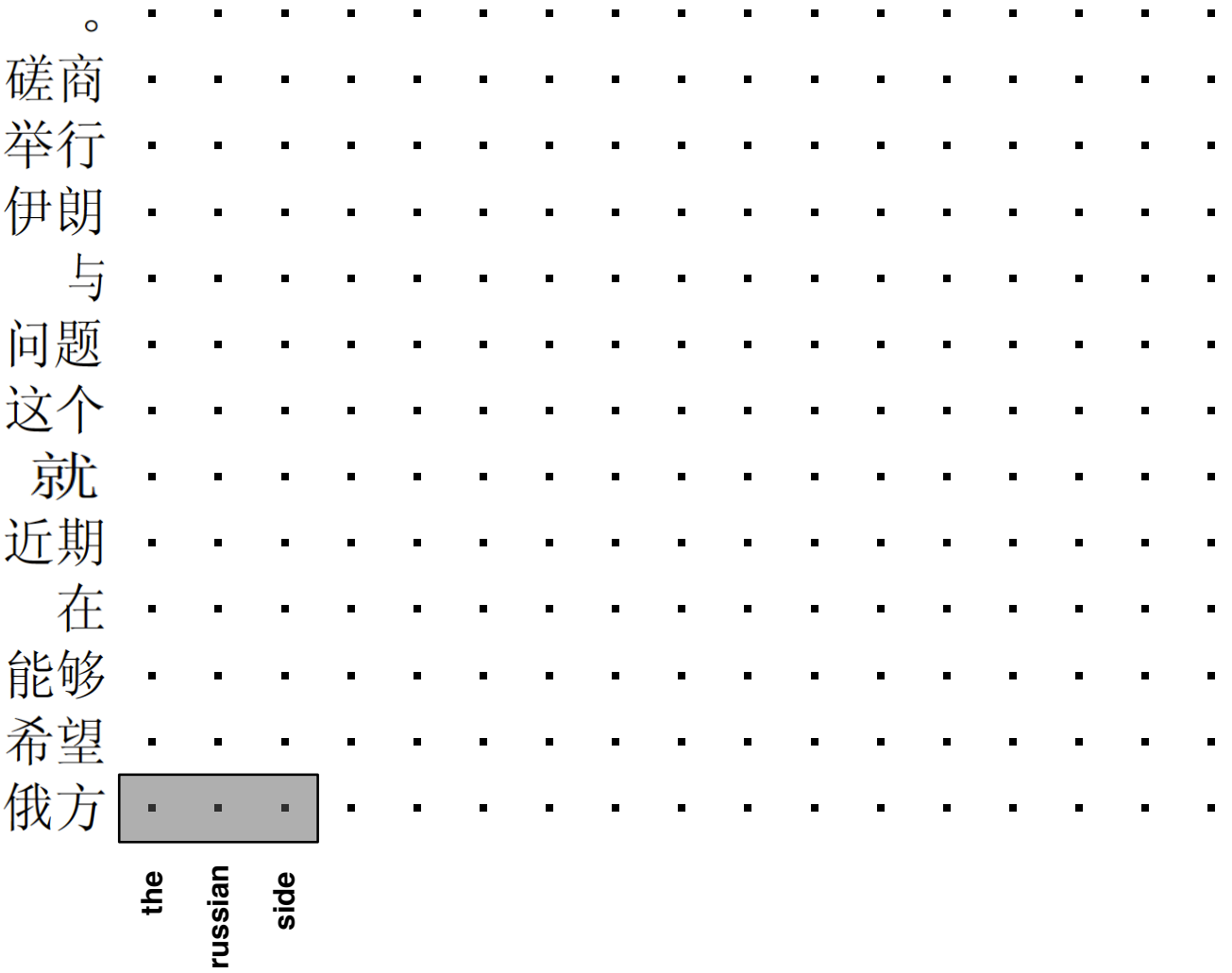
Hierarchical Reordering

- Hierarchical Reordering Models introduced by Galley and Manning (2008)
- Phrase-based alternative to hierarchical reordering found in syntax-based and hierarchical SMT systems
- Translations are still generated from left to right (no chart decoding)
- Hierarchical grouping of phrase applications realized as a shift-reduce (SR) parser
  - Reductions are carried out *before* shifts
- Hypothesis (states) contain a stack of foreign phrase spans
  - Additional criteria for recombination
- Hierarchical reordering often used in combination with lexicalized reordering (and linear distortion)

Hierarchical Reordering: SR-Parsing

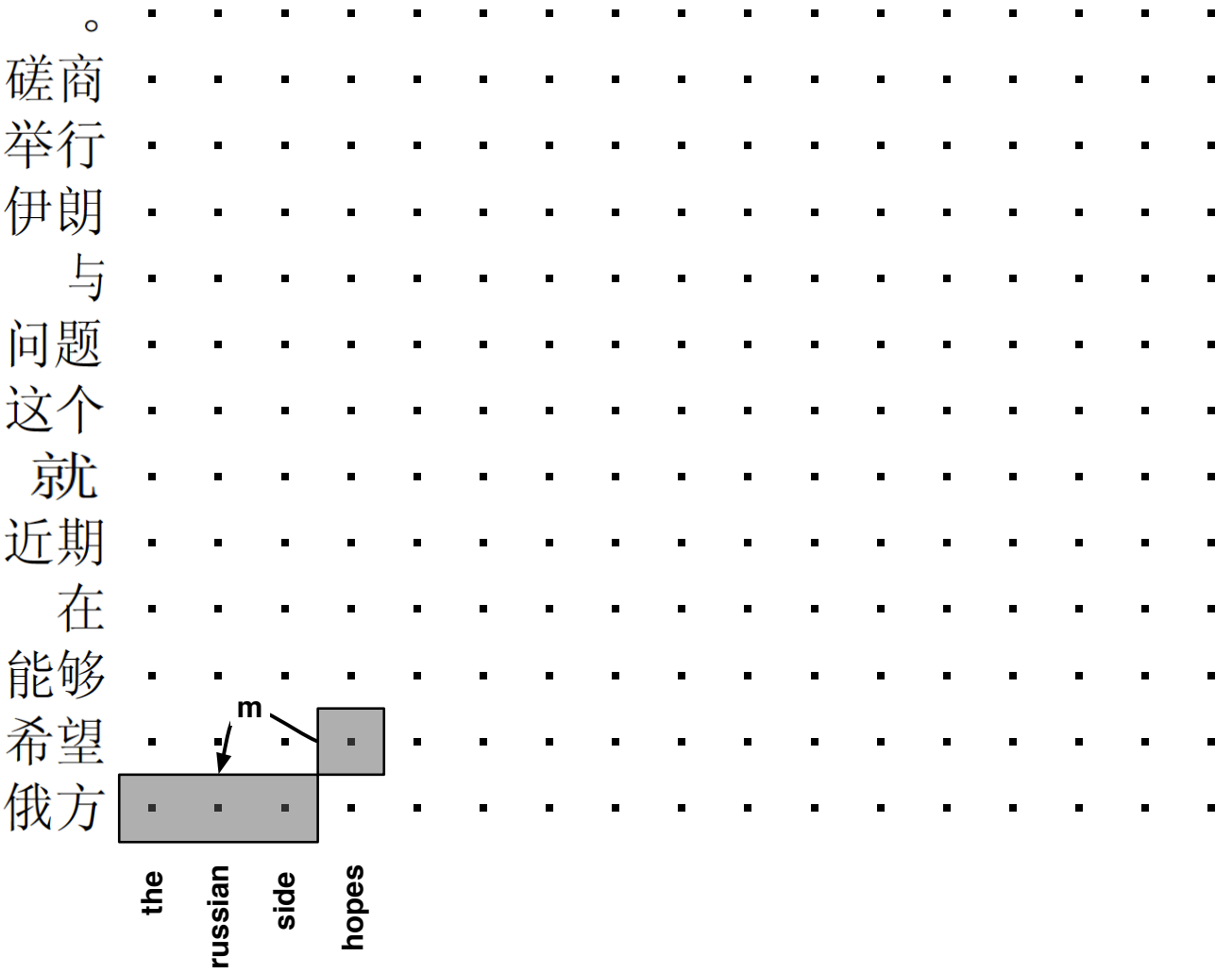
。
磋商
举行
伊朗
与
问题
这个
就
近期
在
能够
希望
俄方

Hierarchical Reordering: SR-Parsing



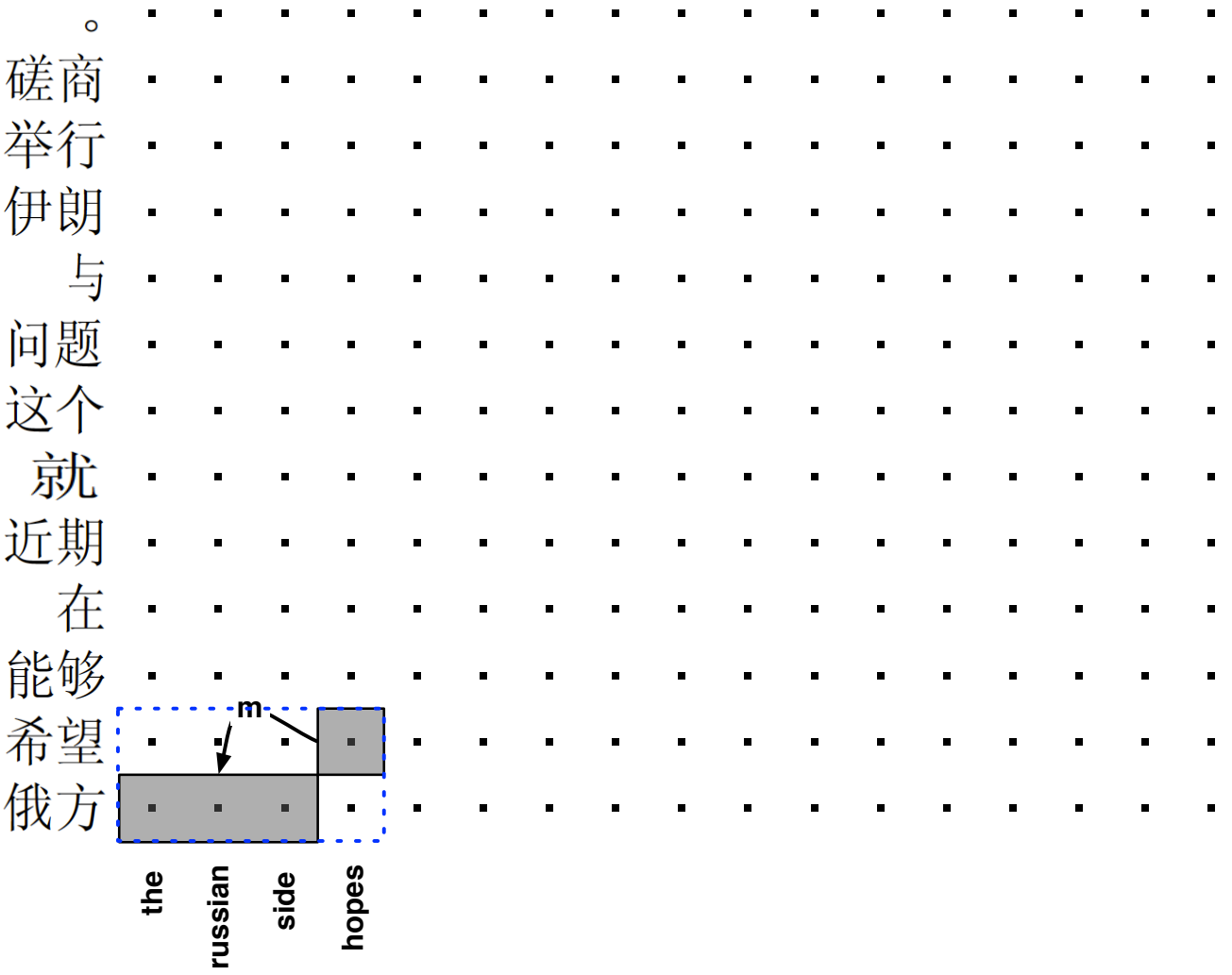
s [1]

Hierarchical Reordering: SR-Parsing



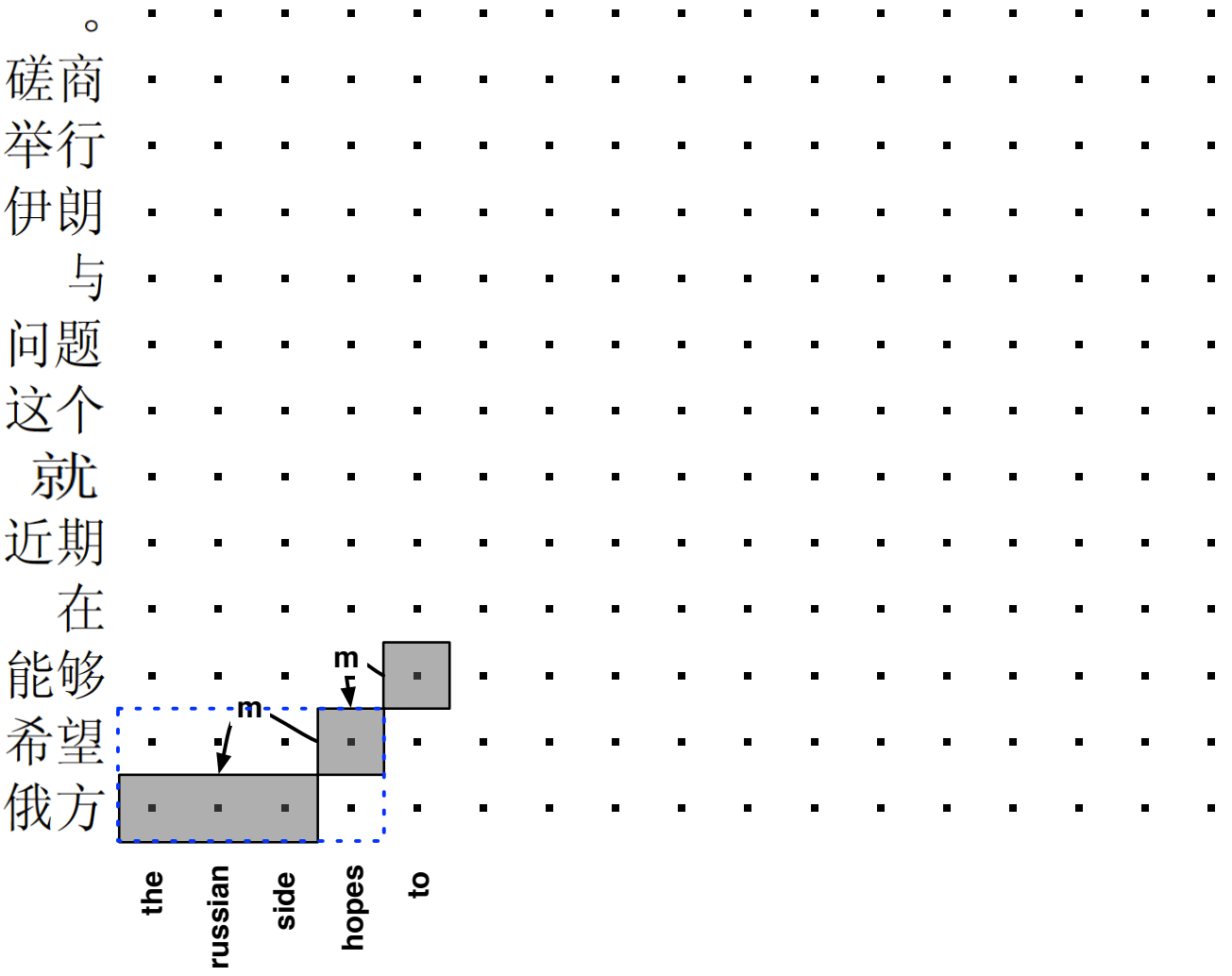
s [1]
s [2], [1]

Hierarchical Reordering: SR-Parsing



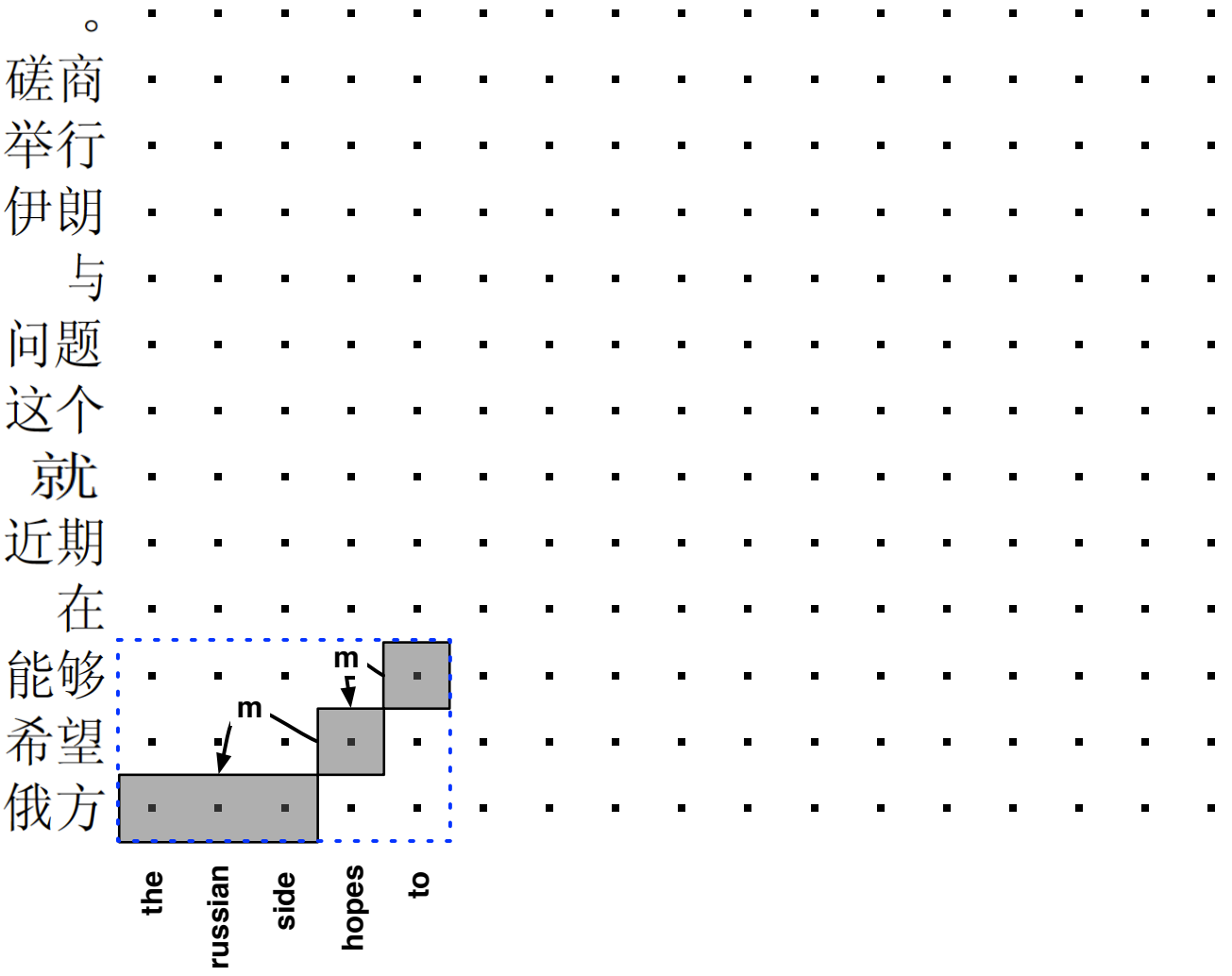
s [1]
s [2], [1]
r [1–2]

Hierarchical Reordering: SR-Parsing



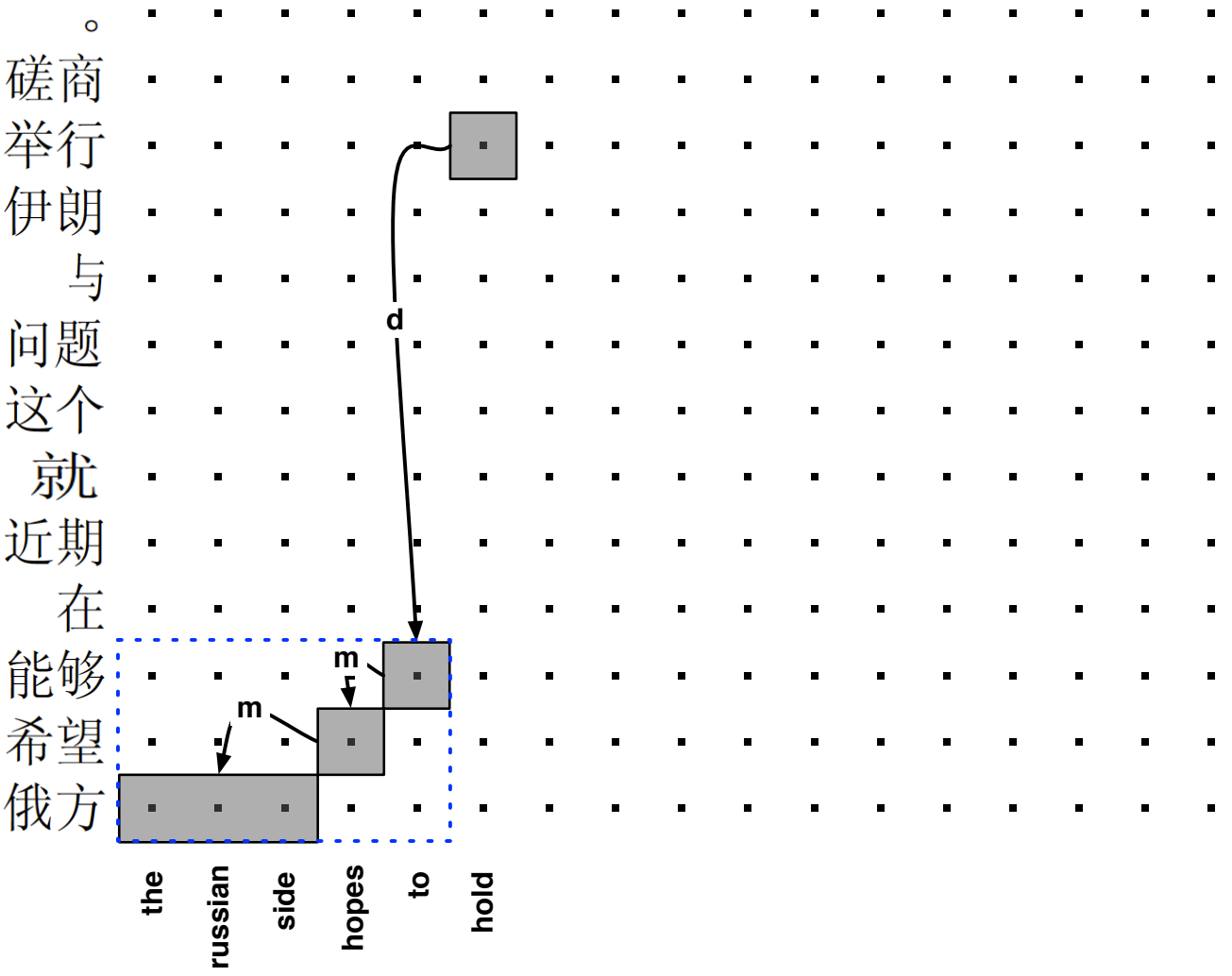
s [1]
s [2], [1]
r [1–2]
s [3], [1–2]

Hierarchical Reordering: SR-Parsing



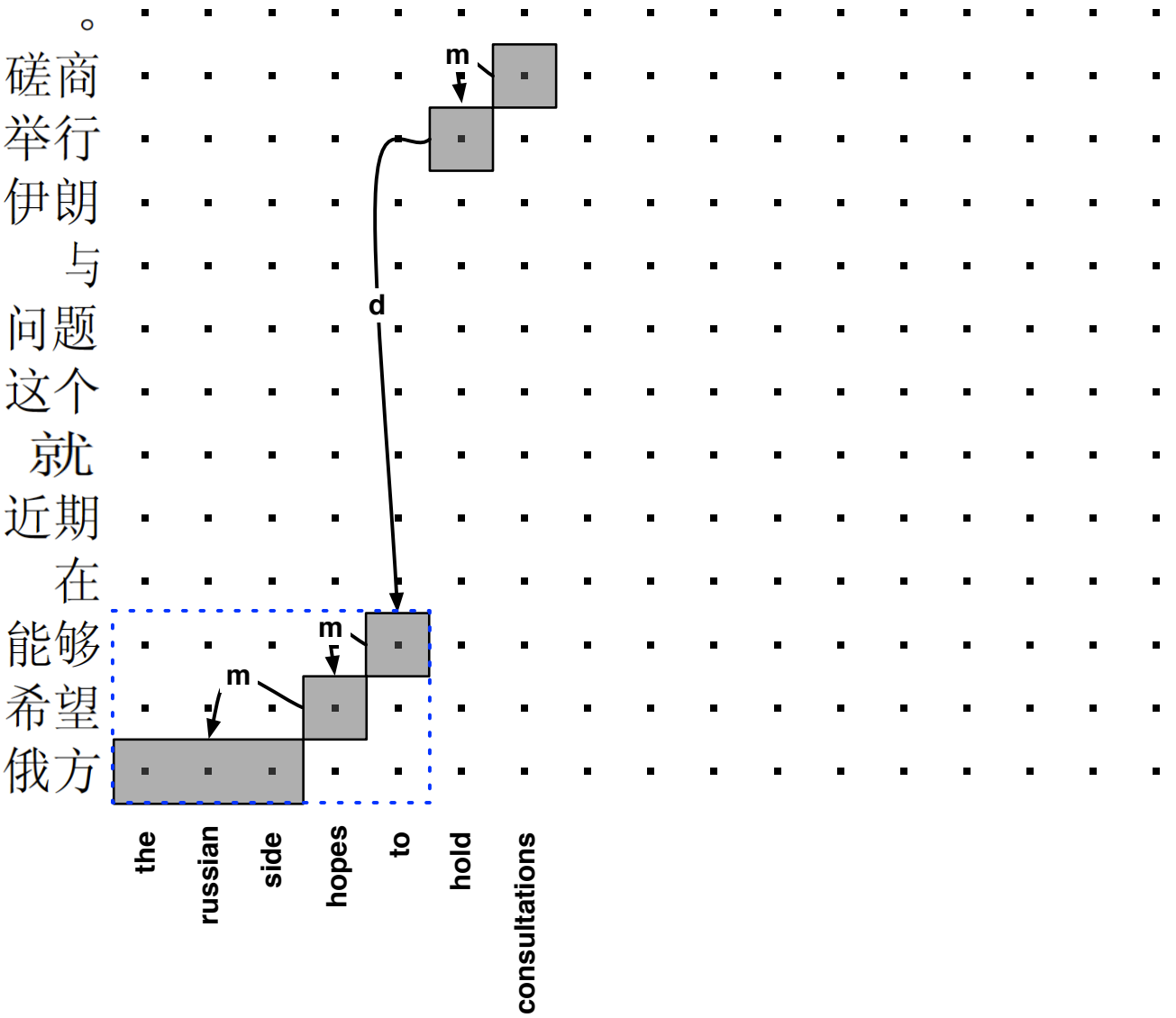
s [1]
s [2], [1]
r [1–2]
s [3], [1–2]
r [1–3]

Hierarchical Reordering: SR-Parsing



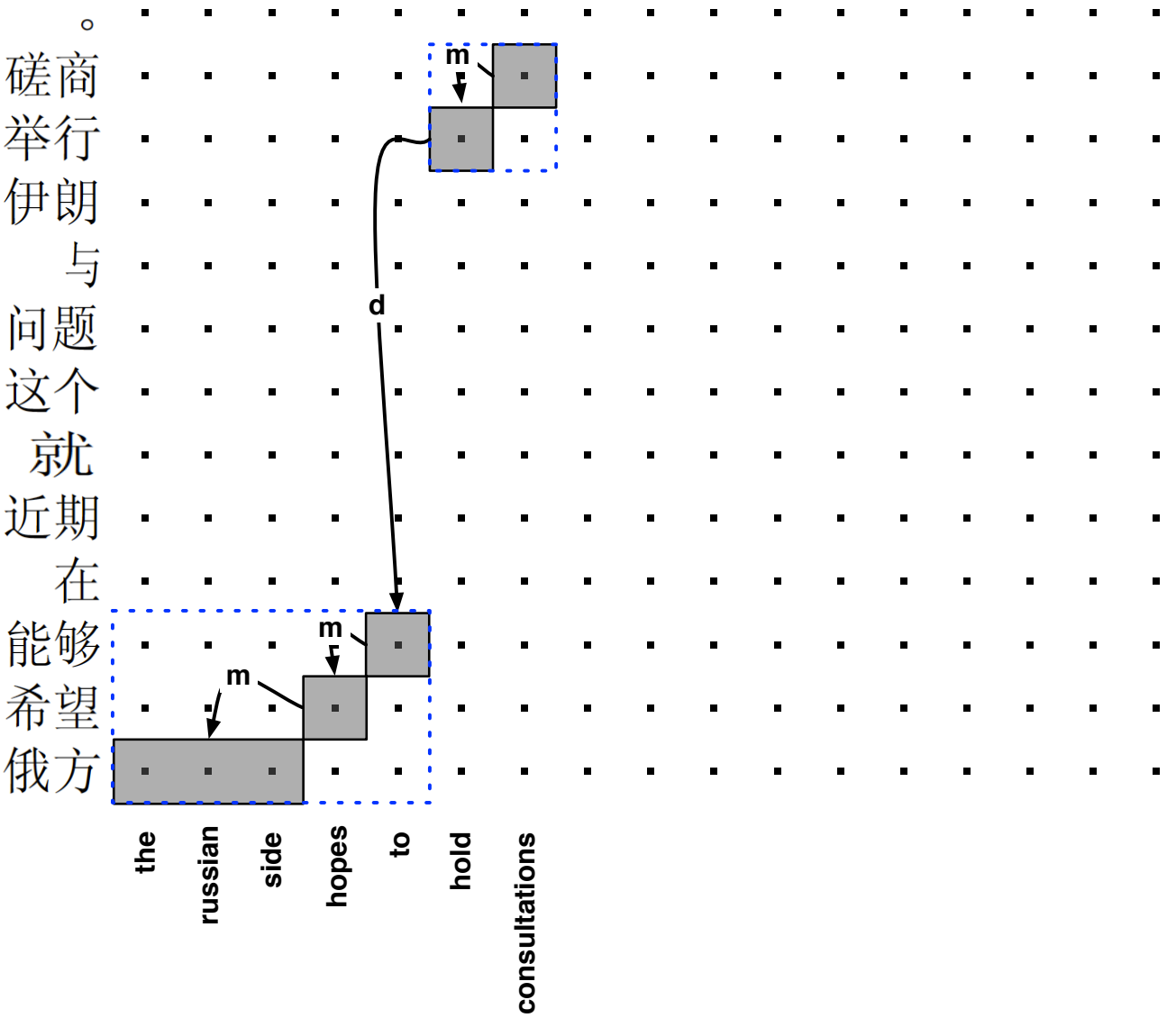
s [1]
s [2], [1]
r [1–2]
s [3], [1–2]
r [1–3]
s [11], [1–3]

Hierarchical Reordering: SR-Parsing



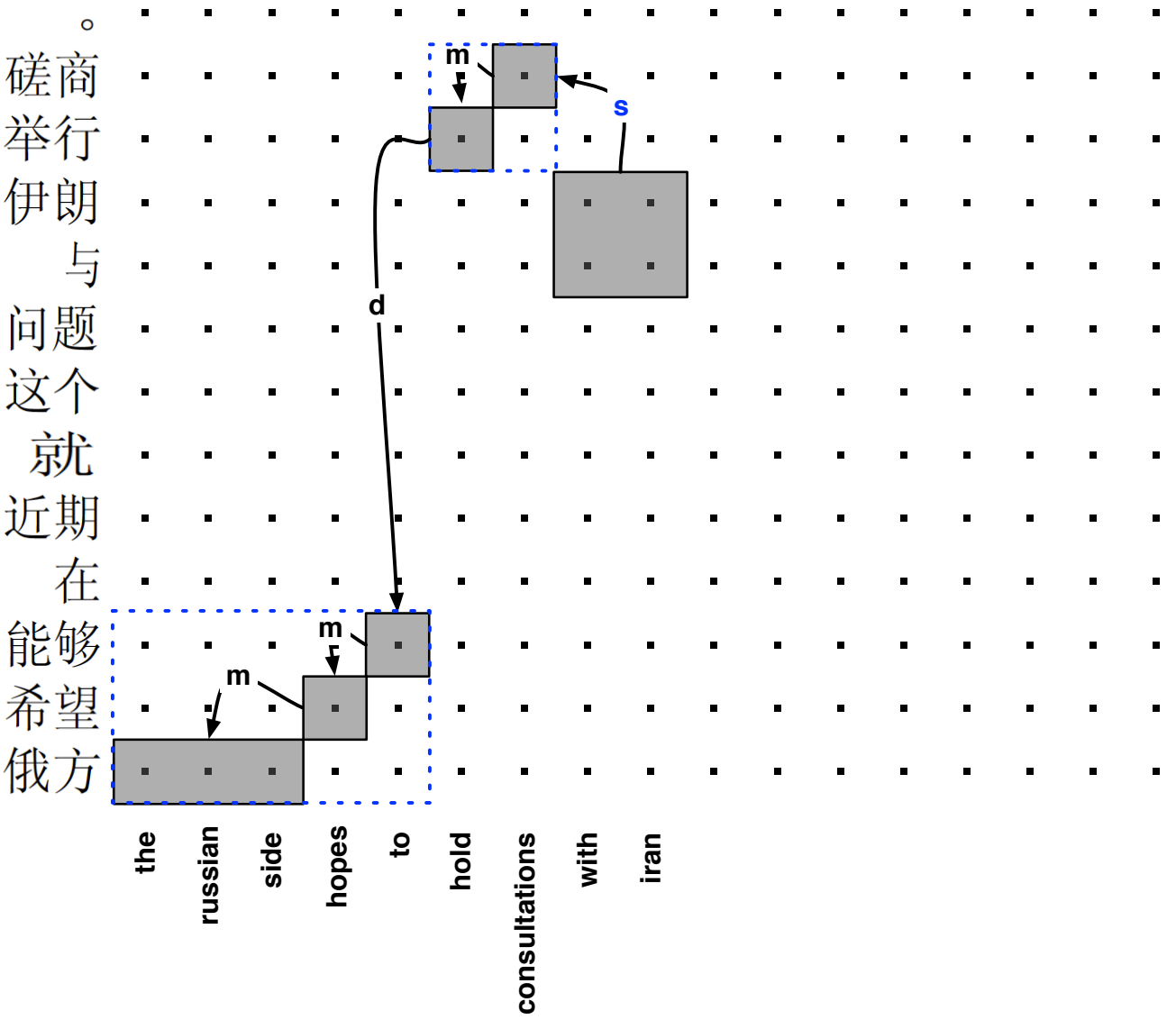
s [1]
s [2], [1]
r [1–2]
s [3], [1–2]
r [1–3]
s [11], [1–3]
s [12], [11], [1–3]

Hierarchical Reordering: SR-Parsing



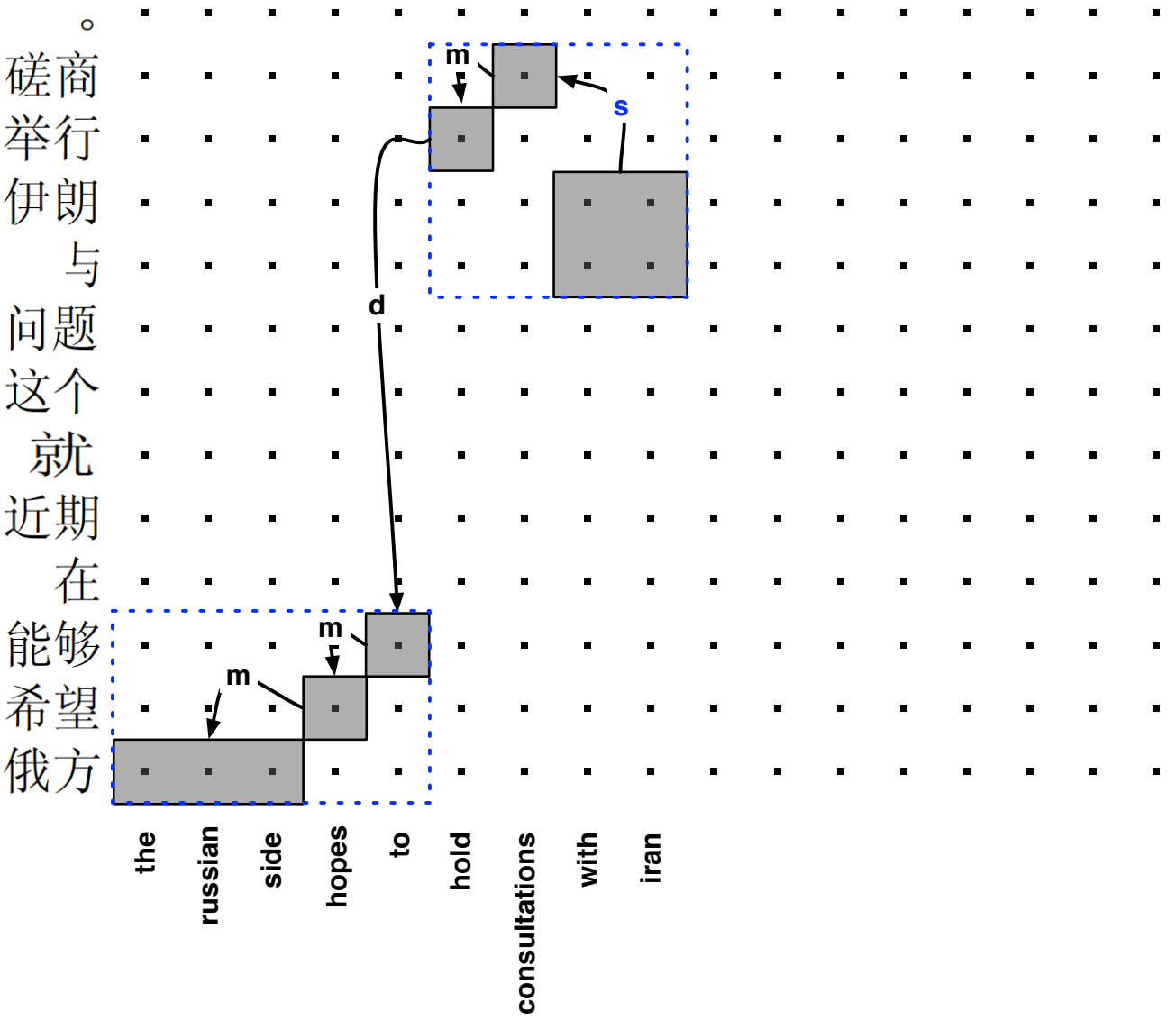
s [1]
s [2], [1]
r [1–2]
s [3], [1–2]
r [1–3]
s [11], [1–3]
s [12], [11], [1–3]
r [11–12], [1–3]

Hierarchical Reordering: SR-Parsing



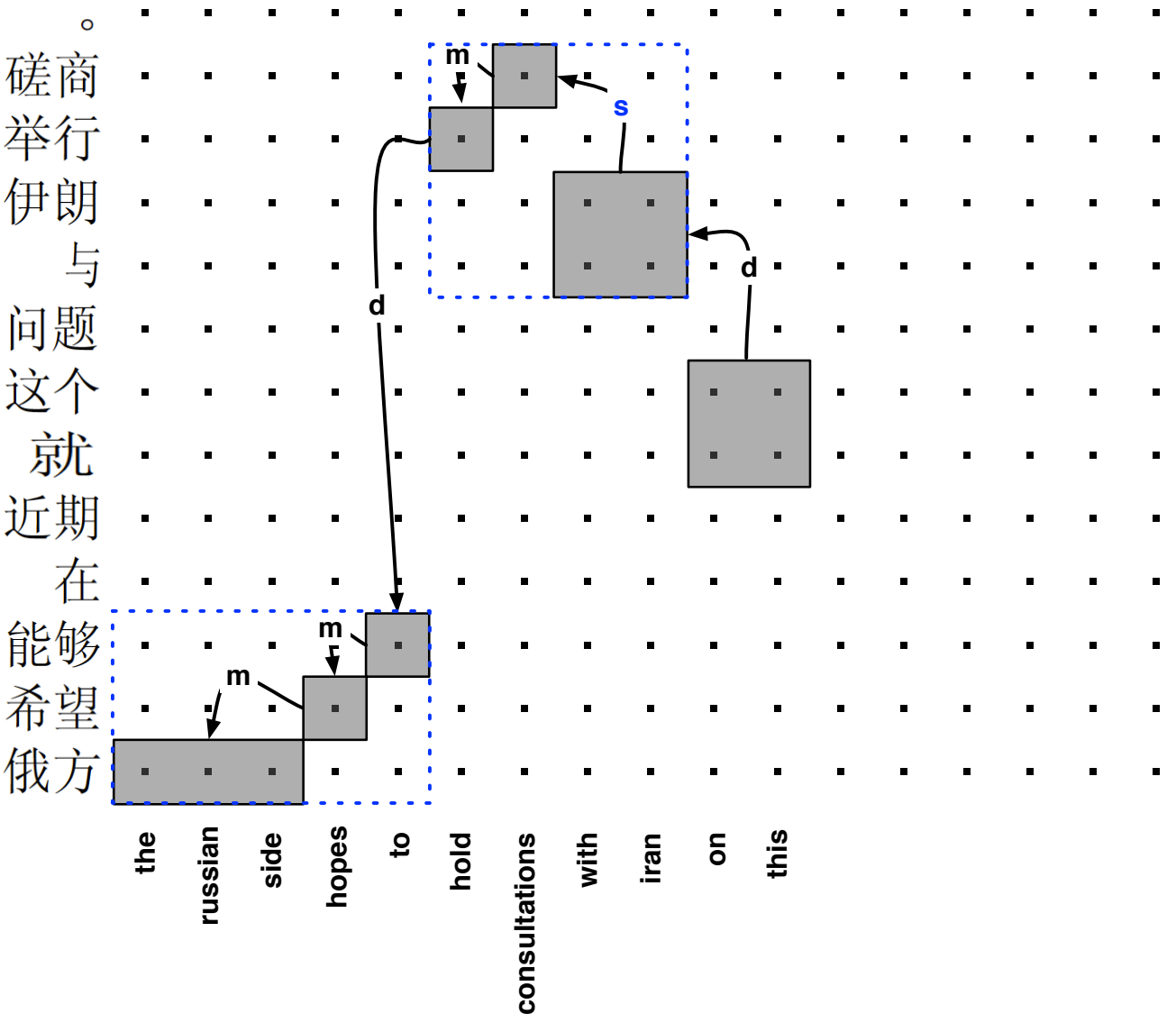
s [1]
s [2], [1]
r [1–2]
s [3], [1–2]
r [1–3]
s [11], [1–3]
s [12], [11], [1–3]
r [11–12], [1–3]
s [9–10], [11–12], [1–3]

Hierarchical Reordering: SR-Parsing



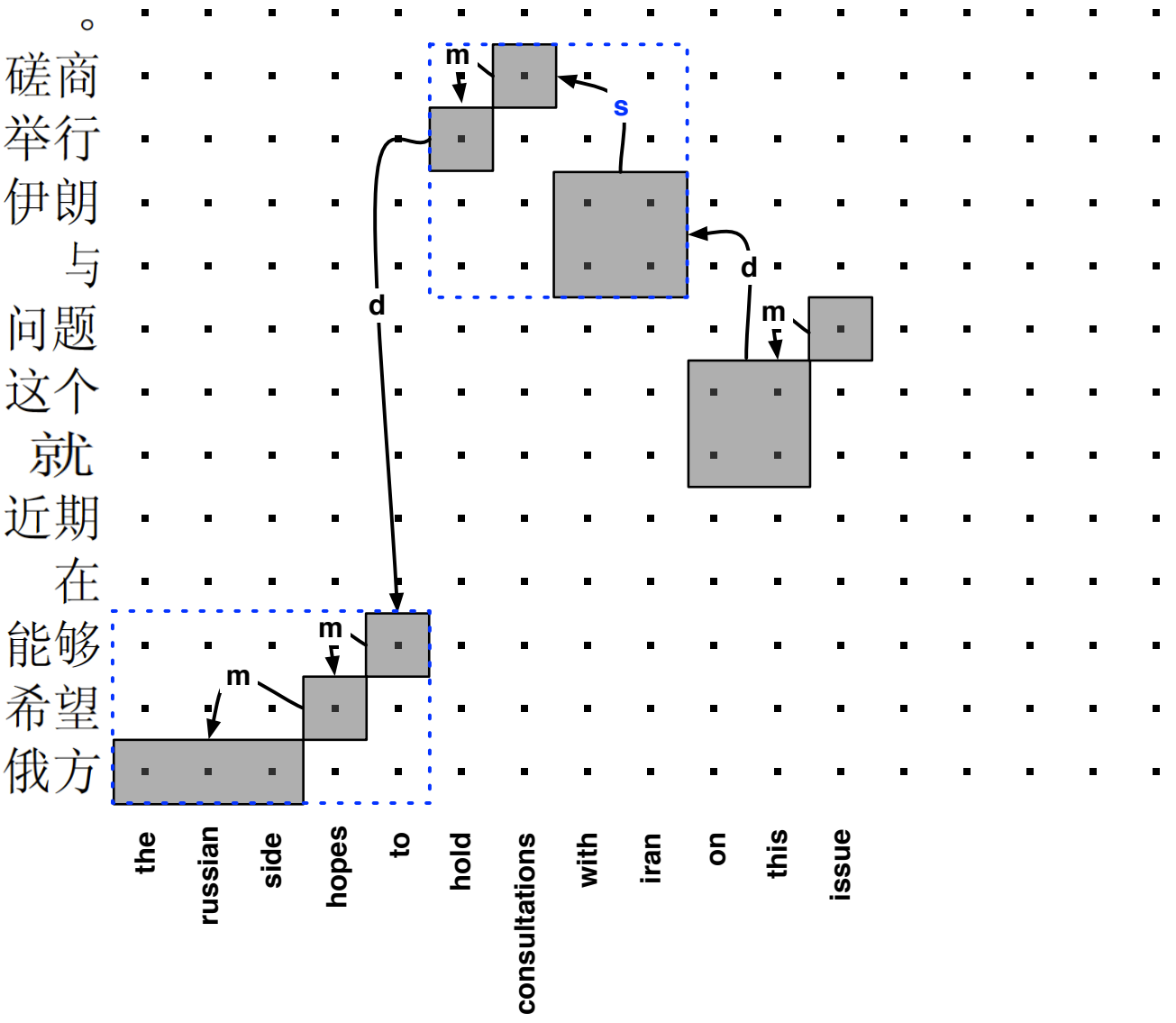
s [1]
s [2], [1]
r [1–2]
s [3], [1–2]
r [1–3]
s [11], [1–3]
s [12], [11], [1–3]
r [11–12], [1–3]
s [9–10], [11–12], [1–3]
r [9–12], [1–3]

Hierarchical Reordering: SR-Parsing



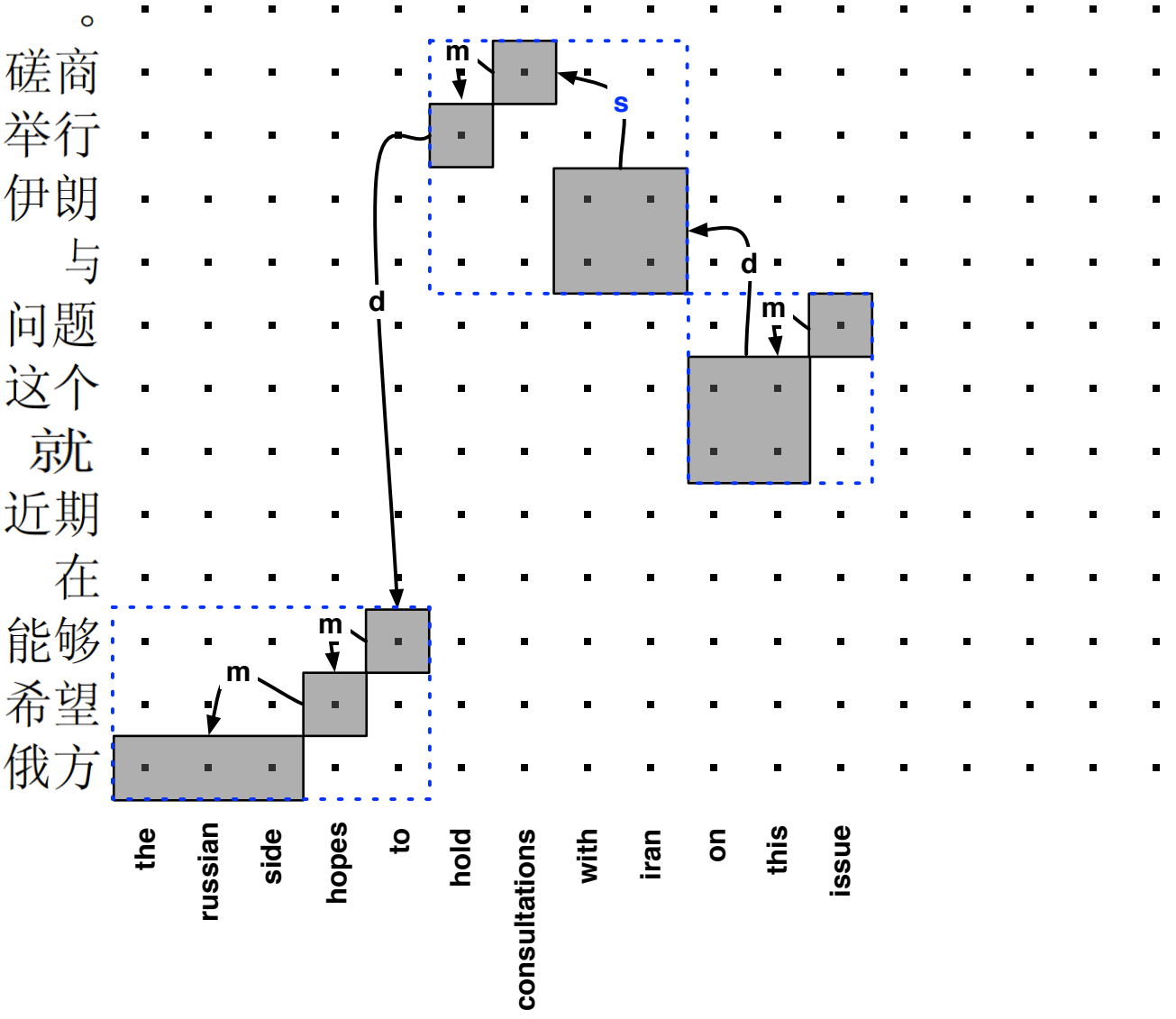
s [1]
s [2], [1]
r [1–2]
s [3], [1–2]
r [1–3]
s [11], [1–3]
s [12], [11], [1–3]
r [11–12], [1–3]
s [9–10], [11–12], [1–3]
r [9–12], [1–3]
s [6–7], [9–12], [1–3]

Hierarchical Reordering: SR-Parsing



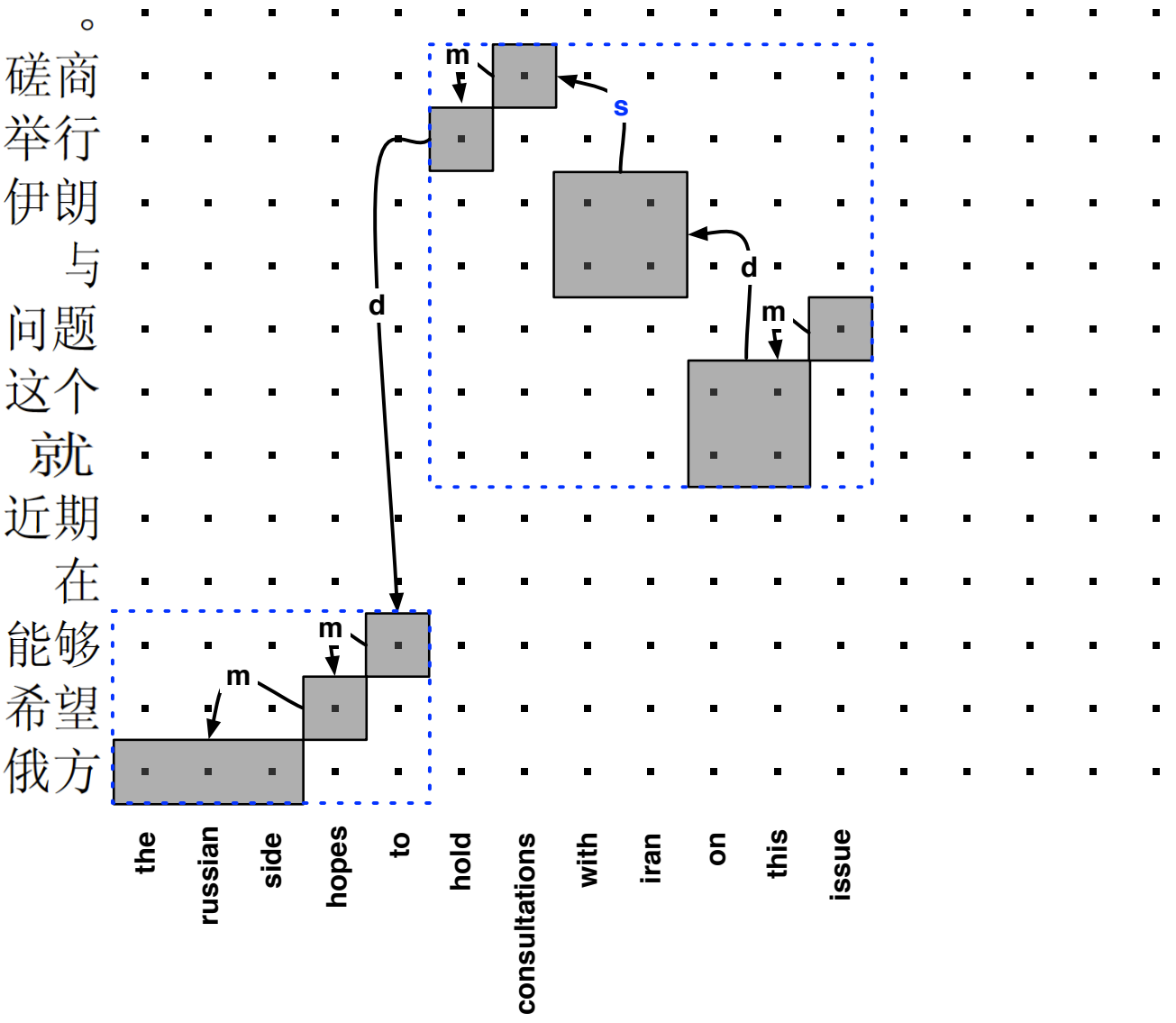
s [1]
s [2], [1]
r [1–2]
s [3], [1–2]
r [1–3]
s [11], [1–3]
s [12], [11], [1–3]
r [11–12], [1–3]
s [9–10], [11–12], [1–3]
r [9–12], [1–3]
s [6–7], [9–12], [1–3]
s [8], [6–7], [9–12], [1–3]

Hierarchical Reordering: SR-Parsing



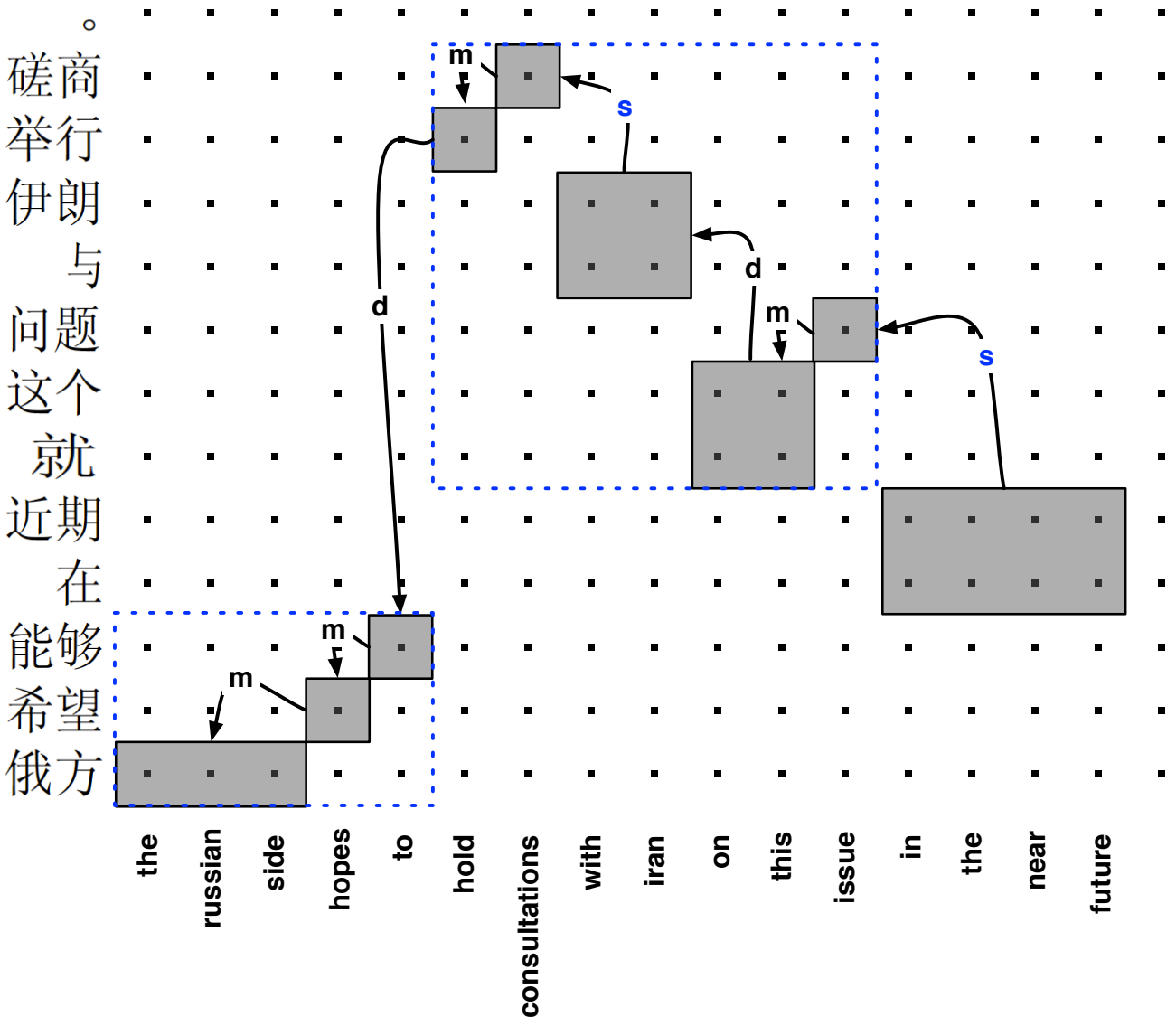
s [1]
s [2], [1]
r [1–2]
s [3], [1–2]
r [1–3]
s [11], [1–3]
s [12], [11], [1–3]
r [11–12], [1–3]
s [9–10], [11–12], [1–3]
r [9–12], [1–3]
s [6–7], [9–12], [1–3]
s [8], [6–7], [9–12], [1–3]
r [6–8], [9–12], [1–3]

Hierarchical Reordering: SR-Parsing



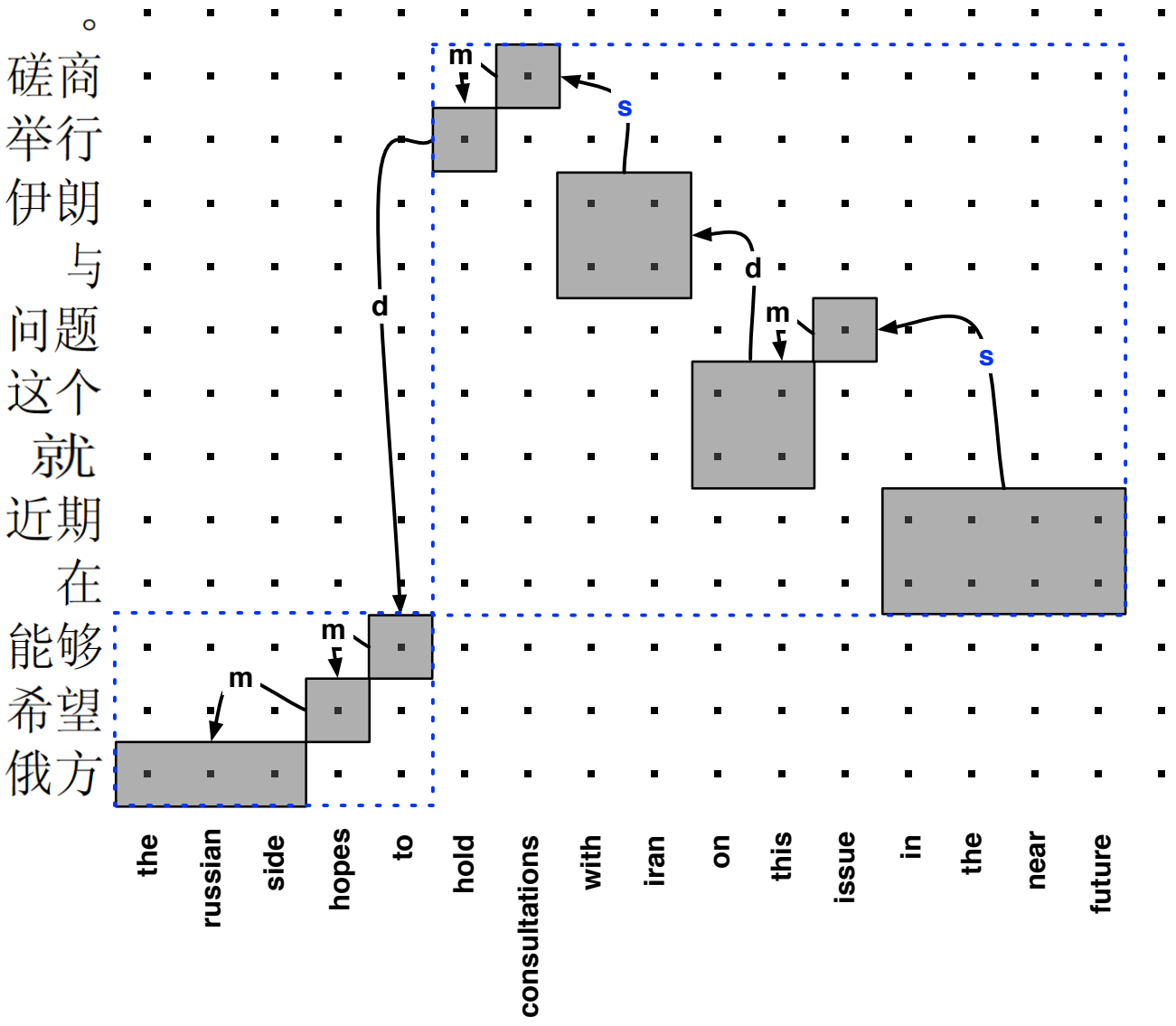
s [1]
s [2], [1]
r [1–2]
s [3], [1–2]
r [1–3]
s [11], [1–3]
s [12], [11], [1–3]
r [11–12], [1–3]
s [9–10], [11–12], [1–3]
r [9–12], [1–3]
s [6–7], [9–12], [1–3]
s [8], [6–7], [9–12], [1–3]
r [6–8], [9–12], [1–3]
r [6–12], [1–3]

Hierarchical Reordering: SR-Parsing



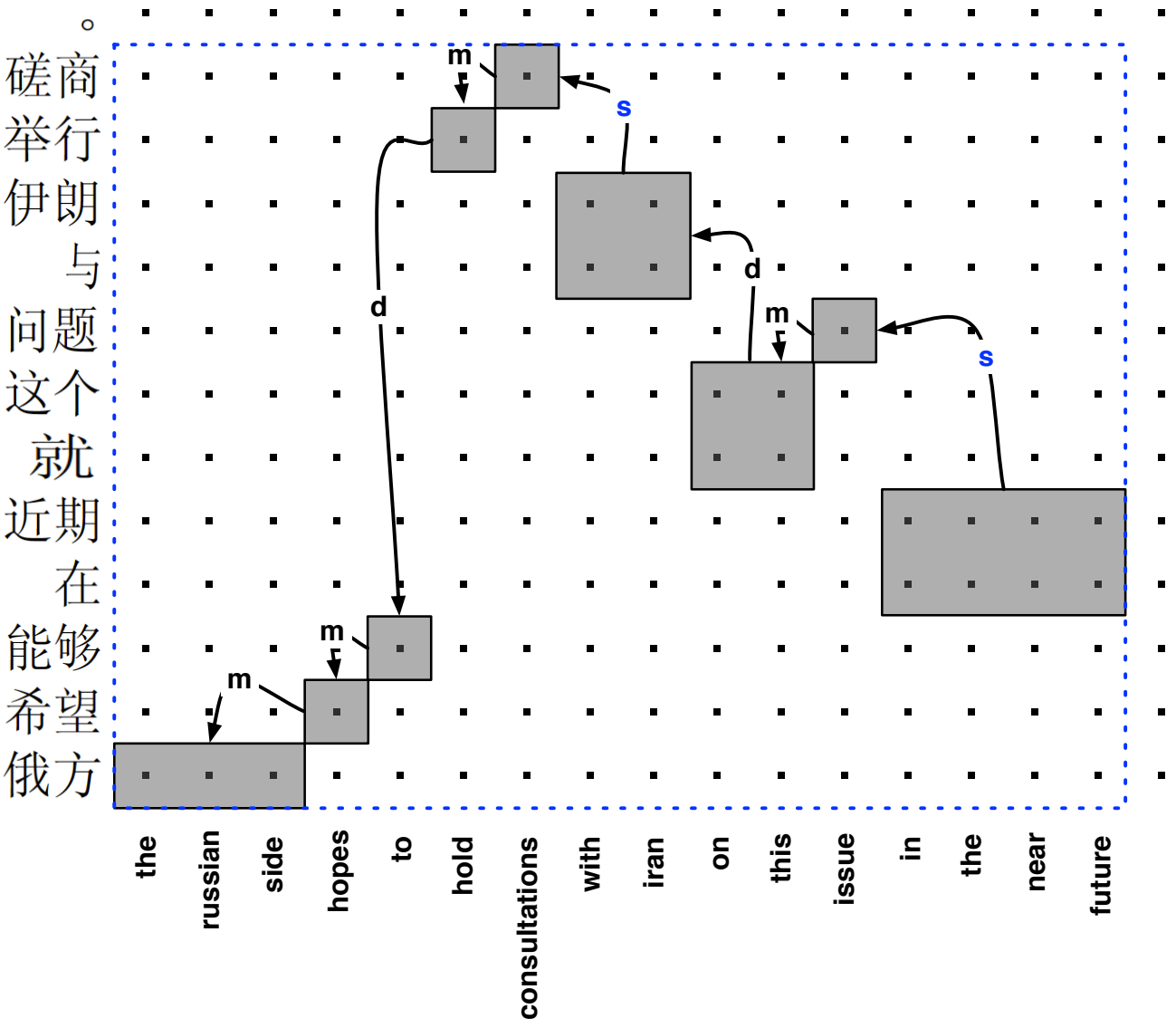
s [1]
s [2], [1]
r [1–2]
s [3], [1–2]
r [1–3]
s [11], [1–3]
s [12], [11], [1–3]
r [11–12], [1–3]
s [9–10], [11–12], [1–3]
r [9–12], [1–3]
s [6–7], [9–12], [1–3]
s [8], [6–7], [9–12], [1–3]
r [6–8], [9–12], [1–3]
r [6–12], [1–3]
s [4–5], [6–12], [1–3]

Hierarchical Reordering: SR-Parsing



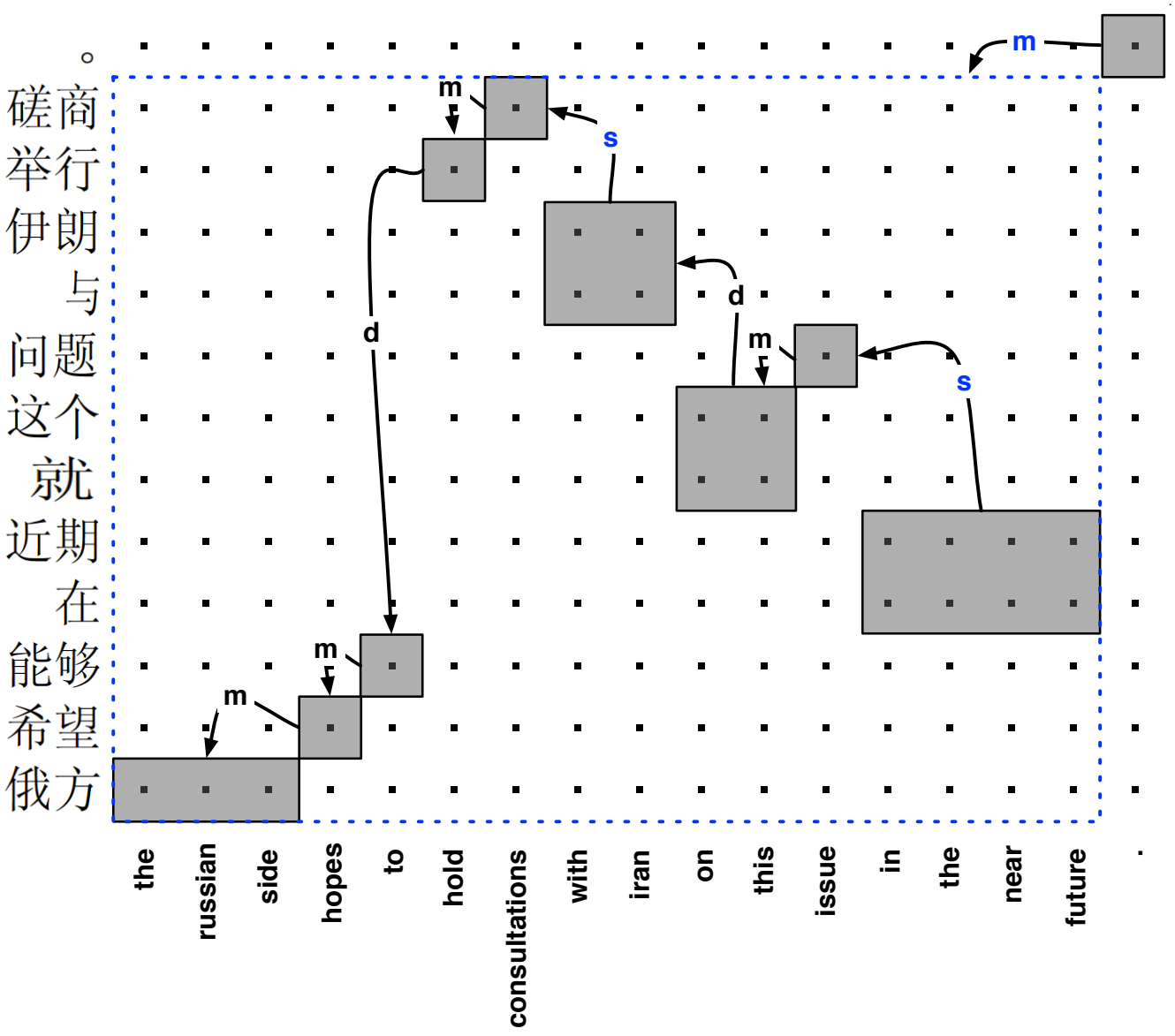
s [1]
s [2], [1]
r [1–2]
s [3], [1–2]
r [1–3]
s [11], [1–3]
s [12], [11], [1–3]
r [11–12], [1–3]
s [9–10], [11–12], [1–3]
r [9–12], [1–3]
s [6–7], [9–12], [1–3]
s [8], [6–7], [9–12], [1–3]
r [6–8], [9–12], [1–3]
r [6–12], [1–3]
s [4–5], [6–12], [1–3]
r [4–12], [1–3]

Hierarchical Reordering: SR-Parsing



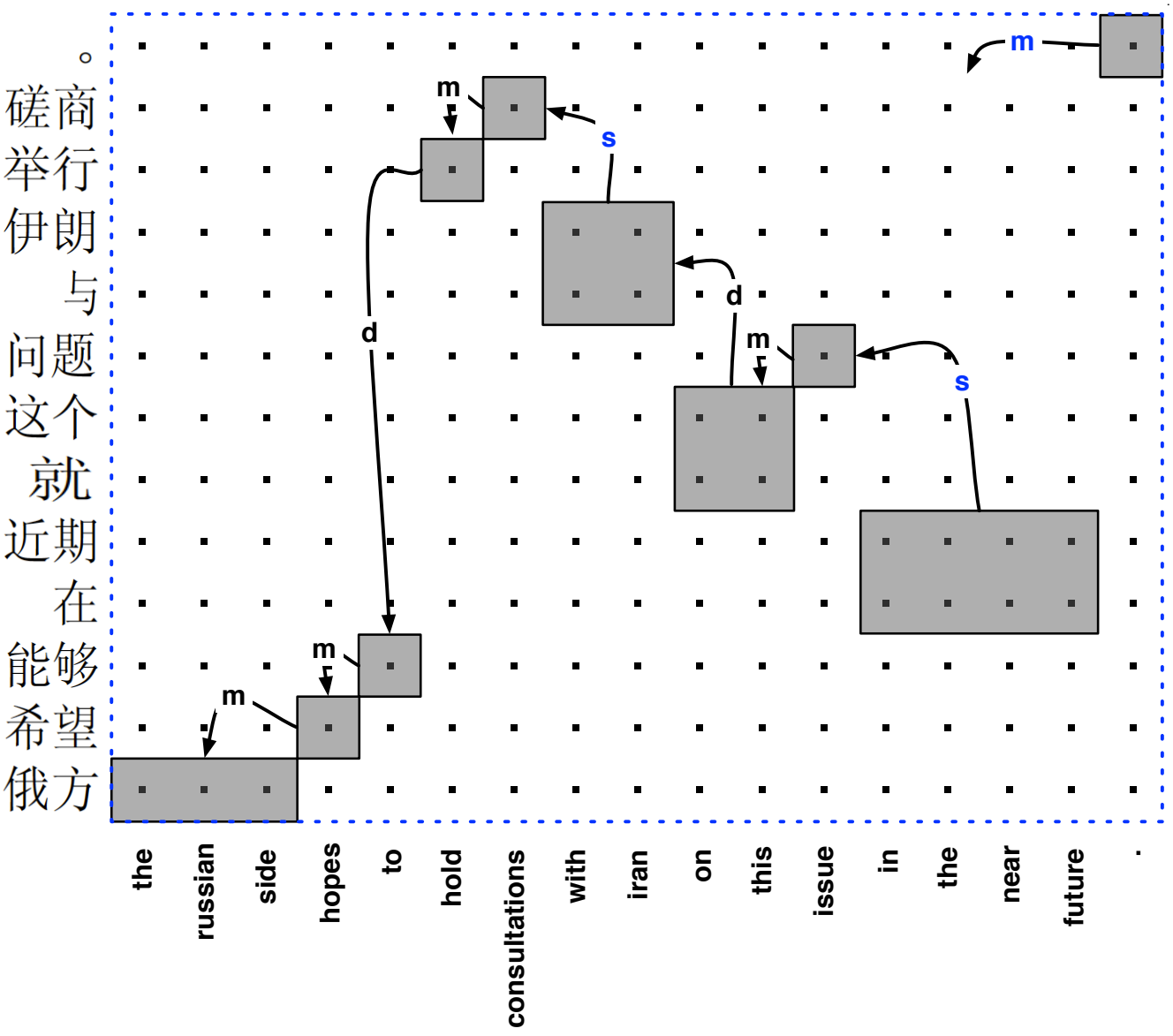
s [1]
s [2], [1]
r [1–2]
s [3], [1–2]
r [1–3]
s [11], [1–3]
s [12], [11], [1–3]
r [11–12], [1–3]
s [9–10], [11–12], [1–3]
r [9–12], [1–3]
s [6–7], [9–12], [1–3]
s [8], [6–7], [9–12], [1–3]
r [6–8], [9–12], [1–3]
r [6–12], [1–3]
s [4–5], [6–12], [1–3]
r [4–12], [1–3]
r [1–12]

Hierarchical Reordering: SR-Parsing



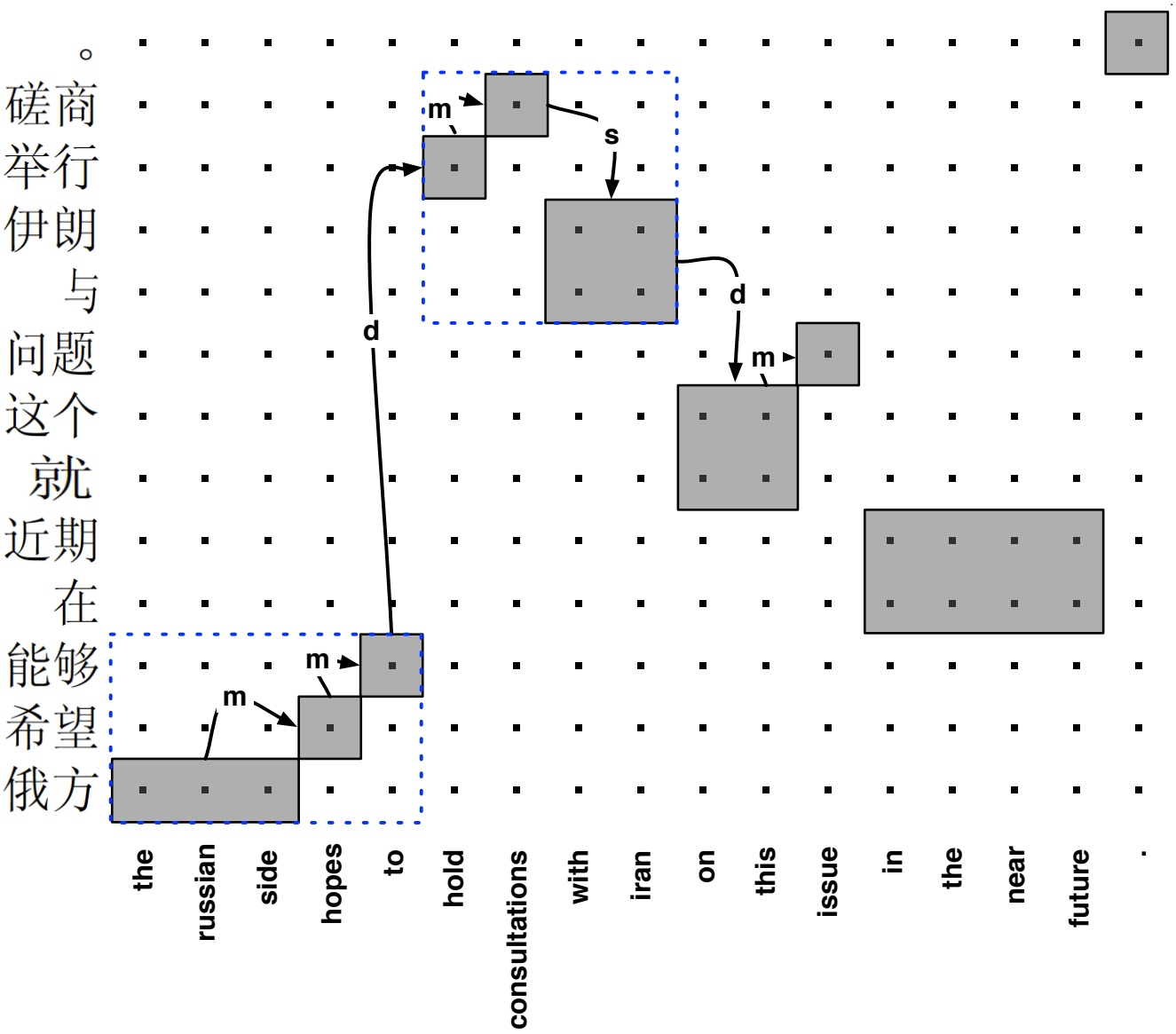
s [1]
s [2], [1]
r [1–2]
s [3], [1–2]
r [1–3]
s [11], [1–3]
s [12], [11], [1–3]
r [11–12], [1–3]
s [9–10], [11–12], [1–3]
r [9–12], [1–3]
s [6–7], [9–12], [1–3]
s [8], [6–7], [9–12], [1–3]
r [6–8], [9–12], [1–3]
r [6–12], [1–3]
s [4–5], [6–12], [1–3]
r [4–12], [1–3]
r [1–12]
s [13], [1–12]

Hierarchical Reordering: SR-Parsing

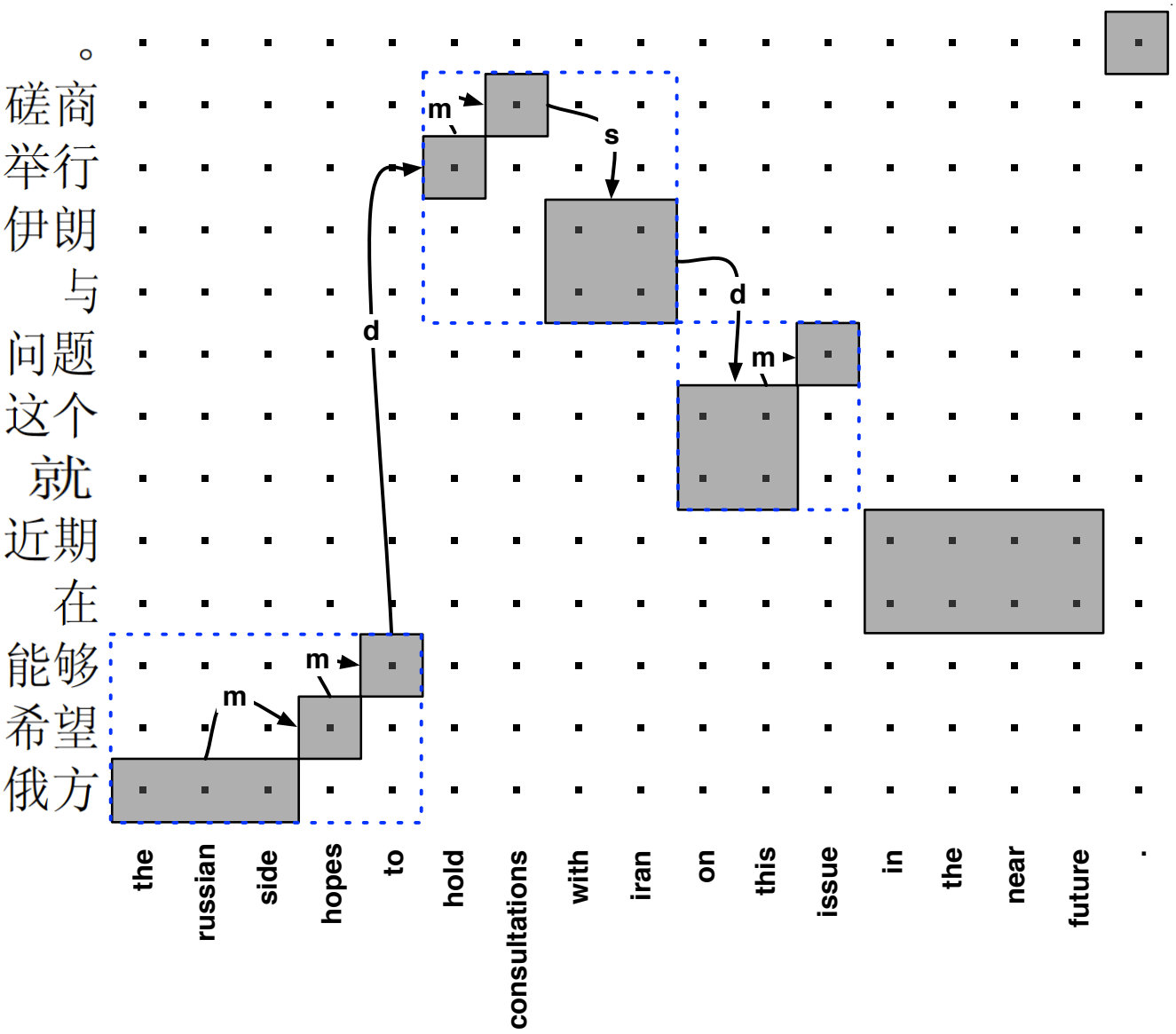


s [1]
s [2], [1]
r [1–2]
s [3], [1–2]
r [1–3]
s [11], [1–3]
s [12], [11], [1–3]
r [11–12], [1–3]
s [9–10], [11–12], [1–3]
r [9–12], [1–3]
s [6–7], [9–12], [1–3]
s [8], [6–7], [9–12], [1–3]
r [6–8], [9–12], [1–3]
r [6–12], [1–3]
s [4–5], [6–12], [1–3]
r [4–12], [1–3]
r [1–12]
s [13], [1–12]
r [1–13]

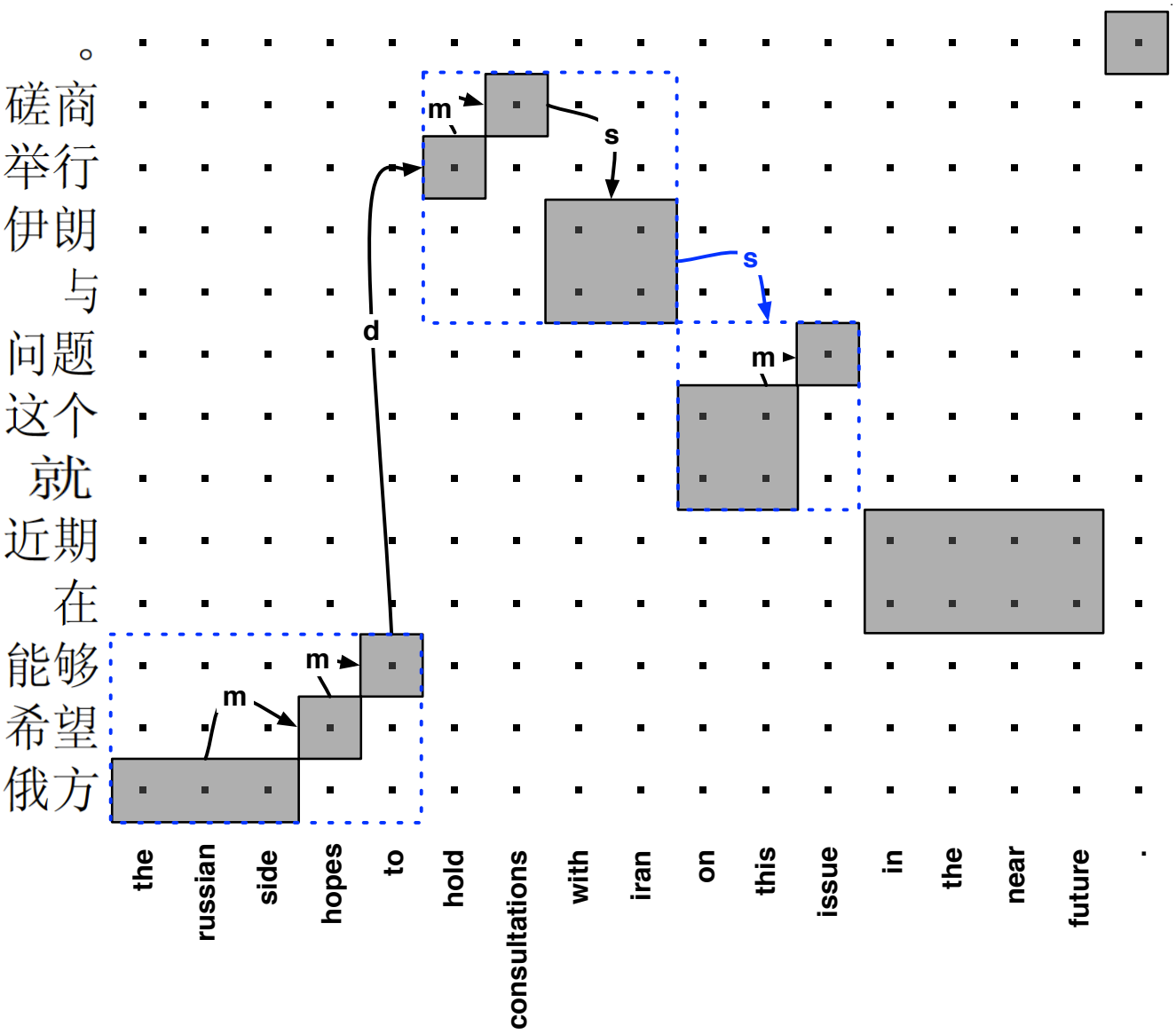
Hierarchical Reordering: Left-to-Right



Hierarchical Reordering: Left-to-Right



Hierarchical Reordering: Left-to-Right



Hierarchical Reordering: Left-to-Right

- Estimation of left-to-right orientations is difficult as we don't know how future phrases applications can be grouped into larger hierarchical phrases
- Left-to-right orientations are approximated
  - Future (block) orientations that are impossible are ignored
  - The remaining orientations are used in the following order of preference:
    - monotone
    - swap
    - discontinuous

Estimation of HRMs

- HRM orientation estimation is achieved similarly to LRM orientation estimation
- For each sentence in the word-aligned parallel corpus we extract two sets of phrase
  - (PP_n): Phrases of maximum length n, which are the phrases that are used during decoding
  - (PP_∞): Phrases of maximum length ∞, which are the phrases wrt which we compute the orientations of the phrases in (PP_n)
- For each phrase pair $(\bar{f}, \bar{e}) \in PP_n$: $c_{r \to l}(o, (\bar{f}, \bar{e})) + +$
  - if **o=m** and $\exists (\bar{f}', \bar{e}') \in PP_\infty : \text{right}(\bar{e}') = \text{left}(\bar{e}) - 1 \wedge \text{right}(\bar{f}') = \text{left}(\bar{f}) - 1$
  - else if **o=s** and $\exists (\bar{f}', \bar{e}') \in PP_\infty : \text{right}(\bar{e}') = \text{left}(\bar{e}) - 1 \wedge \text{left}(\bar{f}') = \text{right}(\bar{f}) + 1$
  - else if **o=d** and $\exists (\bar{f}', \bar{e}') \in PP_\infty : \text{right}(\bar{e}') = \text{left}(\bar{e}) - 1$
- $c_{l \to r}(o, (\bar{f}, \bar{e})) + +$ is estimated analogously

ITG Constraints

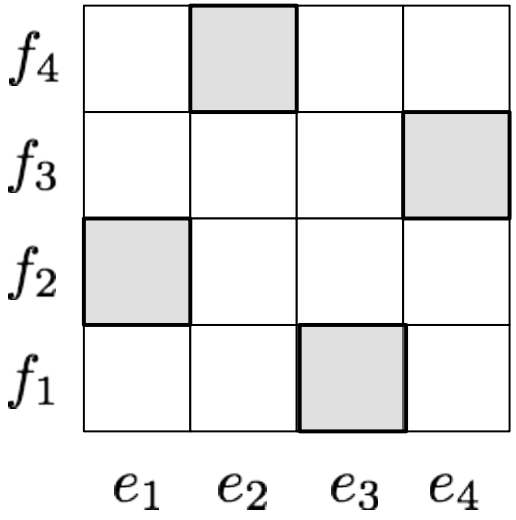
- Inversion Transduction Grammar (ITG) constraints date back to automata theory
- Given a foreign sentence f and a derivation d that yields a translation e of f, we say that derivation d does violate the ITG constraints if d can be rewritten as f: $d \rightarrow_R^* f$ using a rewrite rules of the form
  - $\bar{e} \rightarrow \text{trans}(\bar{e})$
  - $f_1 f_2 \rightarrow f_1 \oplus f_2$
  - $f_1 f_2 \rightarrow f_2 \oplus f_1$
- It is argued that translations should not violate ITG constraints

ITG Constraints

- The shift operation puts one element on the top of the stack
- The reduce operation reduces the top two elements of the stack to one element
  - This can be generalized to n-reduction where the top m $(m \leq n)$ elements can be reduced
  - 2-reduction most commonly used
  - The bigger n the more computations required

ITG Constraints

- Consider the following derivation:



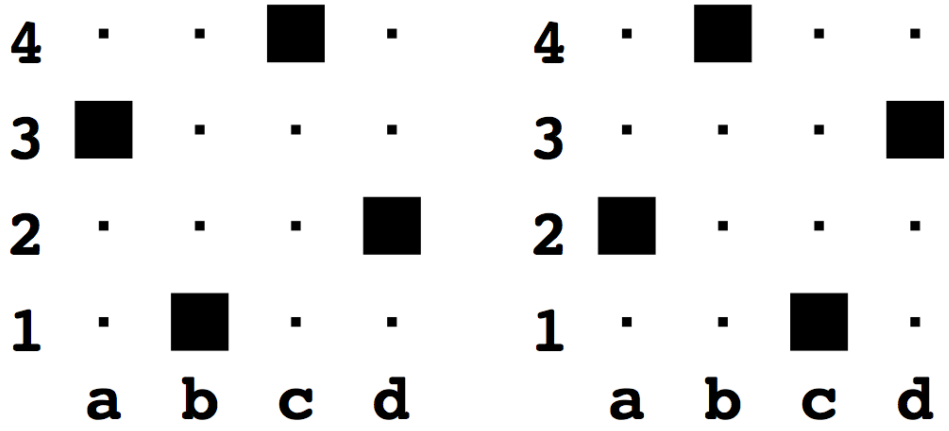
- What n-reduction is required for the decoder to be able to generate this derivation?
- Hierarchical reordering can still produce translations violating ITG constraints
  - Final resulting stack will not be of the form $[0-J-1]$ where J is the length of the foreign sentence

ITG Constraints

- We would like to be able to identify ITG violations as early as possible during decoding
- The approach of Zens et al. (2004) uses the coverage vector
- Let $\bar{f}_i^j$ and $\bar{f}_k^l$ be the previously and currently translated foreign phrases. ITG constraints are violated if
  - $j < k$ and $\exists n : j < n < k-1$ where $\text{cov}(n) = 0$ and $\text{cov}(n+1) = 1$
  - $l < i$ and $\exists n : l < n < i-1$ where $\text{cov}(n) = 1$ and $\text{cov}(n+1) = 0$

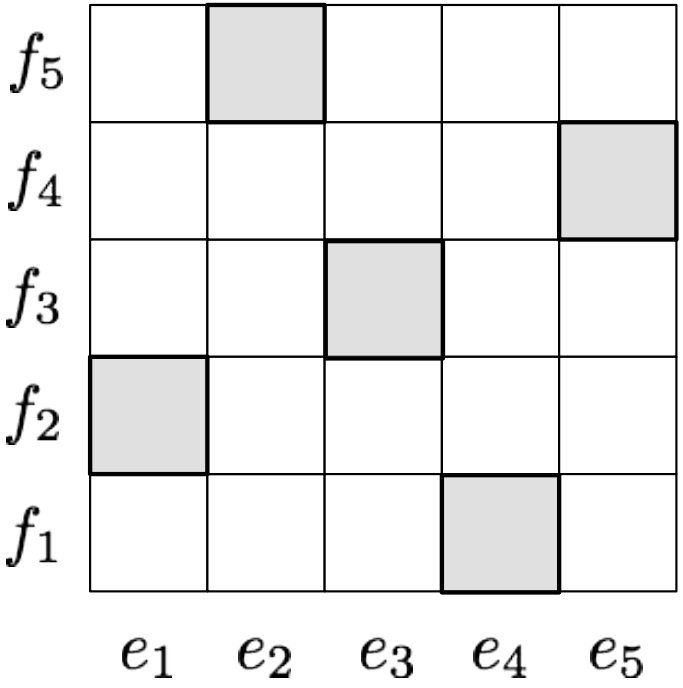
ITG Constraints

- We would like to be able to identify ITG violations as early as possible during decoding
- The approach of Zens et al. (2004) uses the coverage vector
- Let $\bar{f}_i^j$ and $\bar{f}_k^l$ be the previously and currently translated foreign phrases. ITG constraints are violated if
  - $j < k$ and $\exists n : j < n < k-1$ where $\text{cov}(n) = 0$ and $\text{cov}(n+1) = 1$
  - $l < i$ and $\exists n : l < n < i-1$ where $\text{cov}(n) = 1$ and $\text{cov}(n+1) = 0$



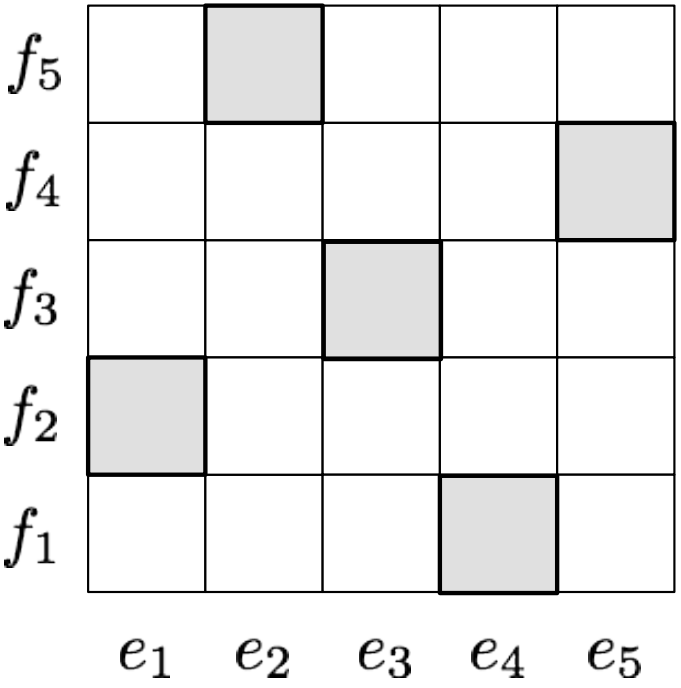
ITG Constraints

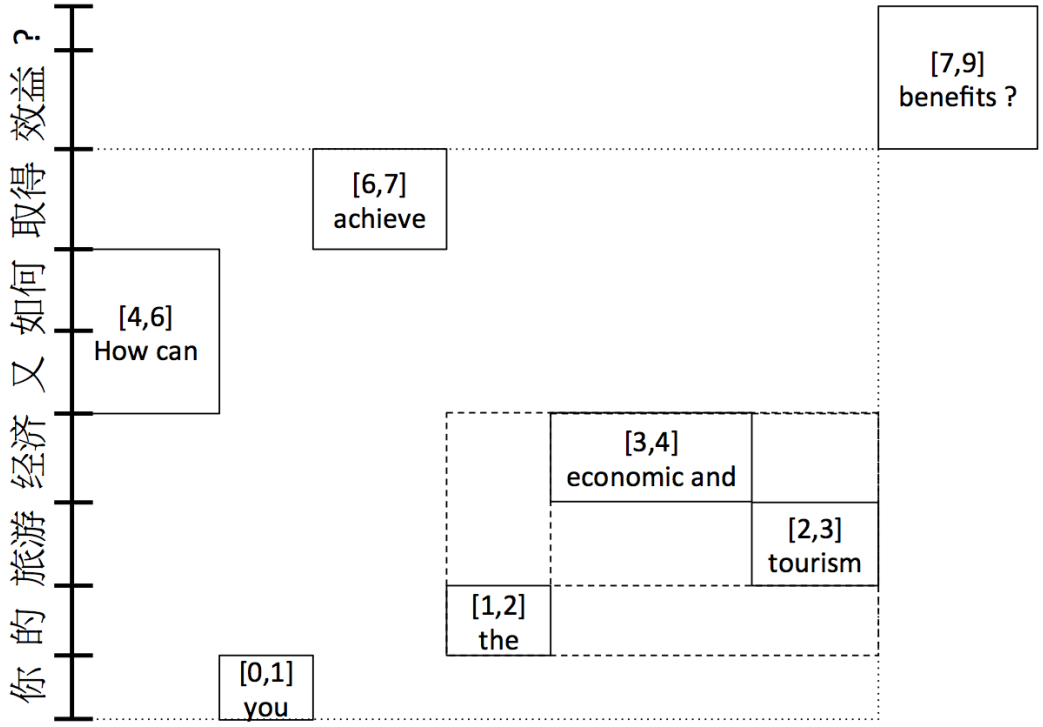
- The coverage vector based approach to ITG violation detection does work correctly for ITG violations involving 4 phrases, but misses higher-order violations



ITG Constraints

- The coverage vector based approach to ITG violation detection does work correctly for ITG violations involving 4 phrases, but misses higher-order violations



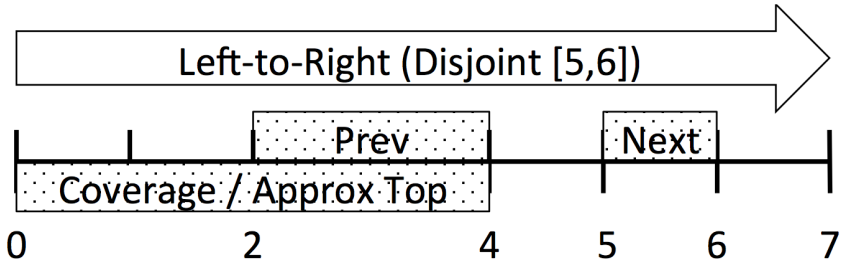


Hierarchical Reordering without Stacks

- Maintaining a stack for each hypothesis and performing shift-reduce parsing further burdens the decoder
- Cherry et al. (2012) proposed an approximation of hierarchical reordering without stacks and shift-reduce parsing
- Reordering orientations are computed with respect to the top element of the stack
  - The top element is a previously shifted element
  - or the result of multiple reduce steps
- The 'true' top element $[i-j]$ can be approximated as the sub-span $[k-l]$ of the coverage vector s.t.
  - for the previously translated foreign phrase $\bar{f}_m^n$: $k \leq m \leq n \leq l$
  - for each n, s.t. $k \leq n \leq l$: $\text{cov}(n) = 1$
  - $k = 0$ or $\text{cov}(k-1) = 0$ and $l = J$ or $\text{cov}(l+1) = 0$ ($J =$ length of foreign sentence -1)
- Most of the times: $k = i$ and $l = j$

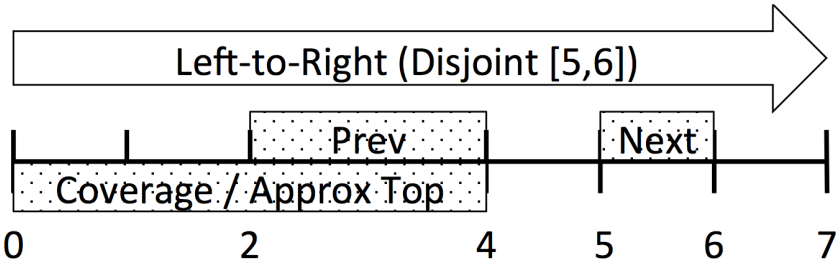
Hierarchical Reordering without Stacks

- Example for right-to-left orientations:

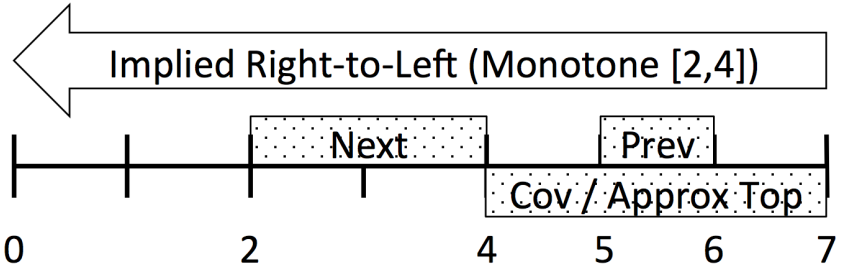


Hierarchical Reordering without Stacks

- Example for right-to-left orientations:



- We can also approximate the left-to-right orientations where the 'true' top element $[i-j]$ can be approximated as the sub-span $[k-l]$ of the coverage vector s.t.
  - for the currently translated foreign phrase $\bar{f}_m^n$: $k \leq m \leq n \leq l$
  - for each s, s.t. $k \leq s < m$ or $n < s \leq l$: $\mathrm{cov}(n) = 0$
  - $k = 0$ or $\mathrm{cov}(k-1) = 1$ and $l = J$ or $\mathrm{cov}(l+1) = 1$
- Example for left-to-right orientations:



Impact of ITG Violations

- Experimental results from Cherry et al. (2012)

	BLEU			NIST 08 Complexity Counts						Speed
Method	nist04	nist06	nist08	> 2	4	5	6	7	≥ 8	sec/sent
LRM	38.00	33.79	27.12	241	146	40	32	12	11	3.187
HRM 2-red	38.53	34.20	27.57	176	113	31	20	8	4	3.353
HRM apprx	38.58	34.09	27.60	280	198	41	26	13	2	3.231
HRM *-red	38.39	34.22	27.41	328	189	71	34	20	14	3.585
HRM itg	38.70	34.26	27.33	0	0	0	0	0	0	3.274

- HRM 2-red (Galley and Manning, 2008) operates as a soft ITG constraint (leading to fewer ITG violations)
- Overall weak correlation between MT quality (as measured by BLEU) and number of ITG violations

Recap

- Distortion limit and distortion constraints
  - linear distortion limit
  - IBM constraints
- Lexicalized reordering modeling
  - Orientation probabilities
  - Estimation of orientation probabilities
- Hierarchical Reordering modeling
  - Shift-Reduce parsing with stacks
  - ITG constraints
  - Approximated HRMs using coverage vectors