A Polynomial-Time Dynamic Programming Algorithm for Phrase-Based Decoding with a Fixed Distortion Limit

Yin-Wen Chang [1]
(Joint work with Michael Collins [1,2])

[1]Google, New York

[2]Columbia University

July 31, 2017

Introduction

Background:

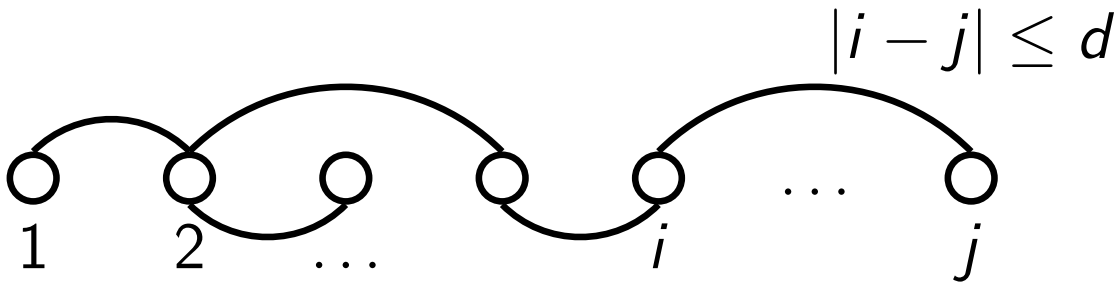
- Phrase-based decoding without further constraints is NP-hard
- Proof: reduction from the travelling salesman problem (TSP)[Knight(1999)]
- Hard distortion limit is commonly imposed in PBMT systems

Question:

- Is phrase-based decoding with a fixed distortion limit NP-hard or not?

Introduction

A related problem: bandwidth-limited TSP



This work: a new decoding algorithm

- Process the source word from left-to-right
- Maintain multiple "tapes" in the target side
- Run time: $O(nd!lh^{d+1})$ n: source sentence length

  d: distortion limit

Overview of the proposed decoding algorithm

1	2	3	4	5	6
das	muss	unsere	sorge	gleichermaßen	sein

$\pi_1 = \epsilon$

$\pi_2 = \epsilon$

- Process the source word from left-to-right
- Maintain multiple "tapes" in the target side

Overview of the proposed decoding algorithm

1	2	3	4	5	6
das	muss	unsere	sorge	gleichermaßen	sein

$$\pi_1 \leftarrow \pi_1 \cdot (1, 2, \text{this must})$$

$$\pi_1 = (1, 2, \text{this must})$$

$$\pi_2 = \epsilon$$

- Process the source word from left-to-right
- Maintain multiple "tapes" in the target side

Overview of the proposed decoding algorithm

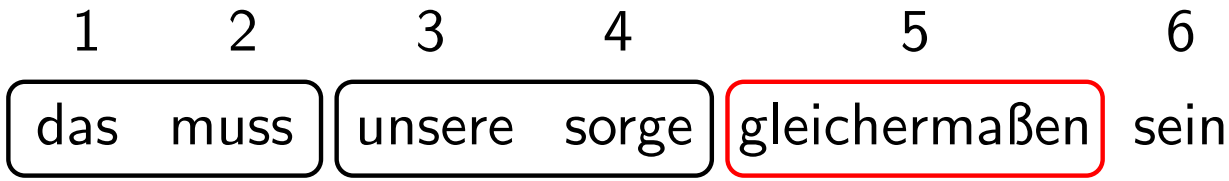
$$\pi_2 \leftarrow \pi_2 \cdot (3, 4, \text{our concern})$$

$\pi_1 = (1, 2, \text{this must})$

$\pi_2 = (3, 4, \text{our concern})$

- Process the source word from left-to-right
- Maintain multiple "tapes" in the target side

Overview of the proposed decoding algorithm



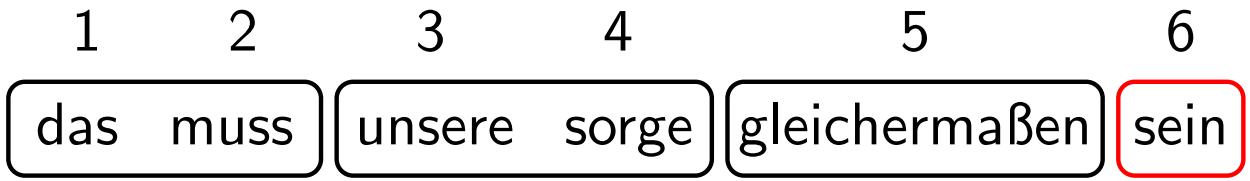
$$\pi_1 \leftarrow \pi_1 \cdot (5, 5, \text{also})$$

$$\pi_1 = (1, 2, \text{this must})(5, 5, \text{also})$$

$$\pi_2 = (3, 4, \text{our concern})$$

- Process the source word from left-to-right
- Maintain multiple "tapes" in the target side

Overview of the proposed decoding algorithm



$$\pi_1 \leftarrow \pi_1 \cdot (6, 6, \text{be}) \cdot \pi_2$$

$$\pi_1 = (1, 2, \text{this must})(5, 5, \text{also})(6, 6, \text{be})(3, 4, \text{our concern})$$

$$\pi_2 = \epsilon$$

- Process the source word from left-to-right
- Maintain multiple "tapes" in the target side

Outline

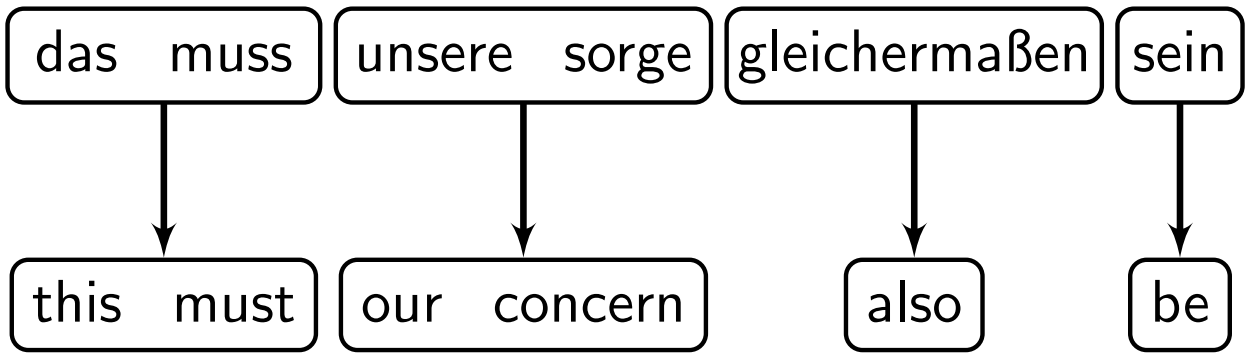
Phrase-based decoding problem

das muss unsere sorge gleichermaßen sein

Phrase-based decoding problem

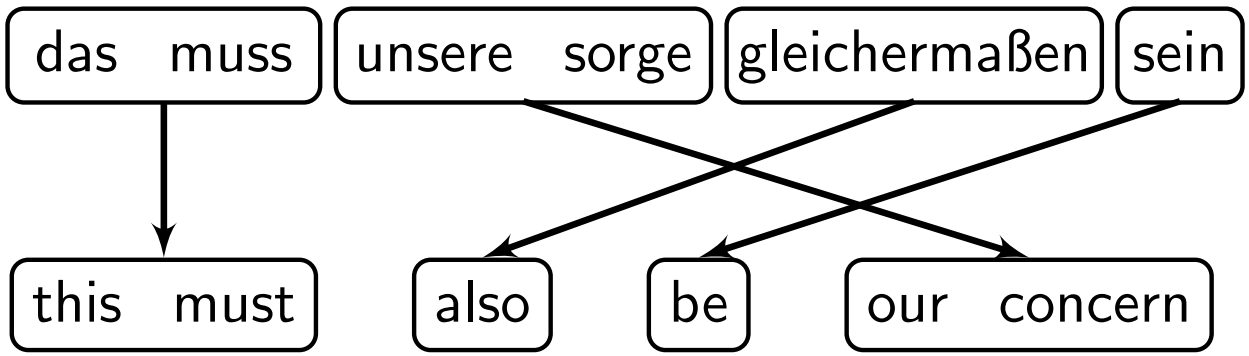
- Segment the German sentence into non-overlapping phrases

Phrase-based decoding problem



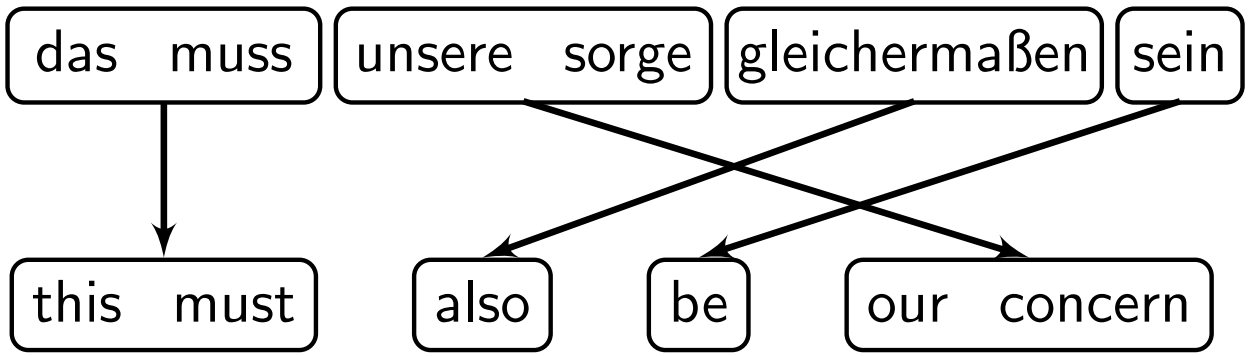
- Segment the German sentence into non-overlapping phrases
- Find an English translation for each German phrase

Phrase-based decoding problem



- Segment the German sentence into non-overlapping phrases
- Find an English translation for each German phrase
- Reorder the English phrases to get a better English sentence

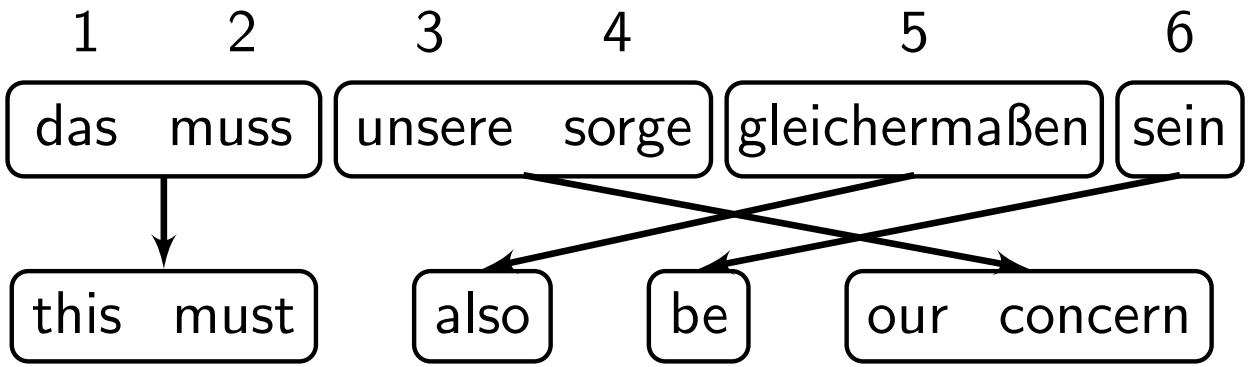
Phrase-based decoding problem



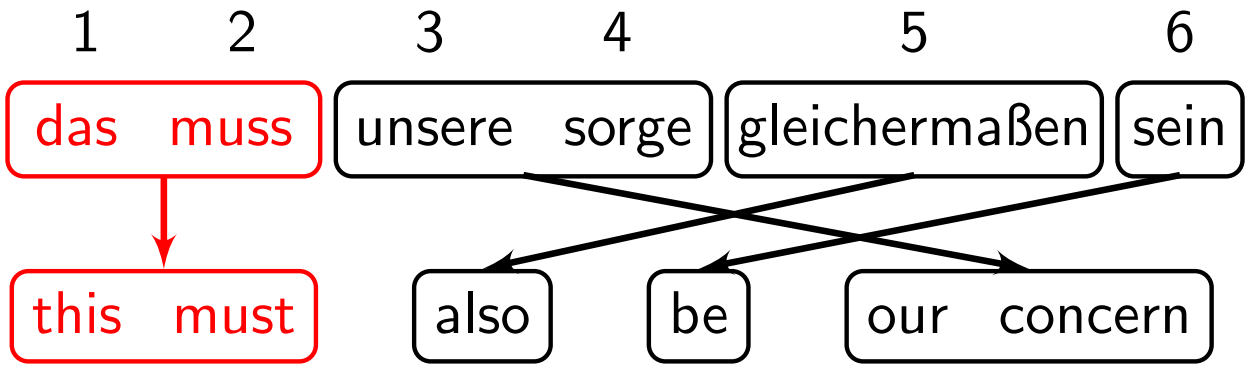
- Segment the German sentence into non-overlapping phrases
- Find an English translation for each German phrase
- Reorder the English phrases to get a better English sentence

Derivation: complete translation with phrase mappings
Sub-derivation: partial translation

Score a derivation

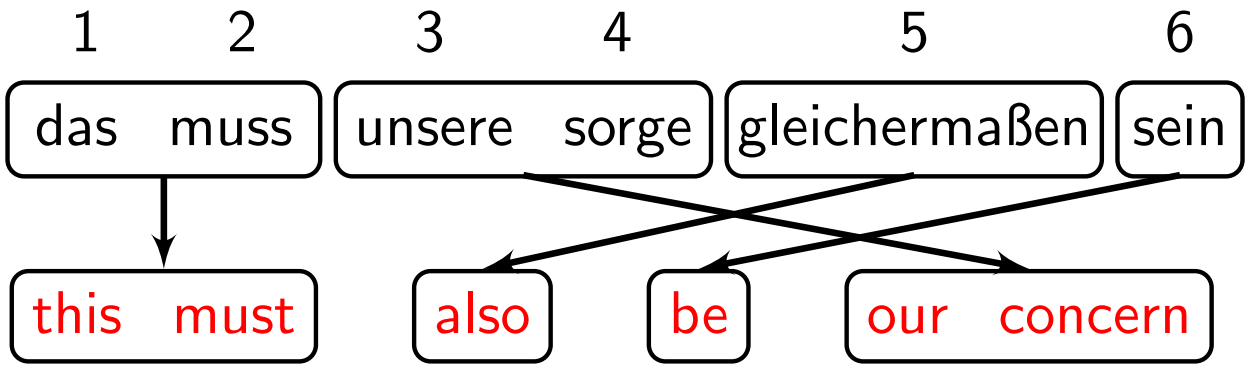


Score a derivation



- Phrase translation score: $score(\text{das muss}, \text{this must}) + \cdots$

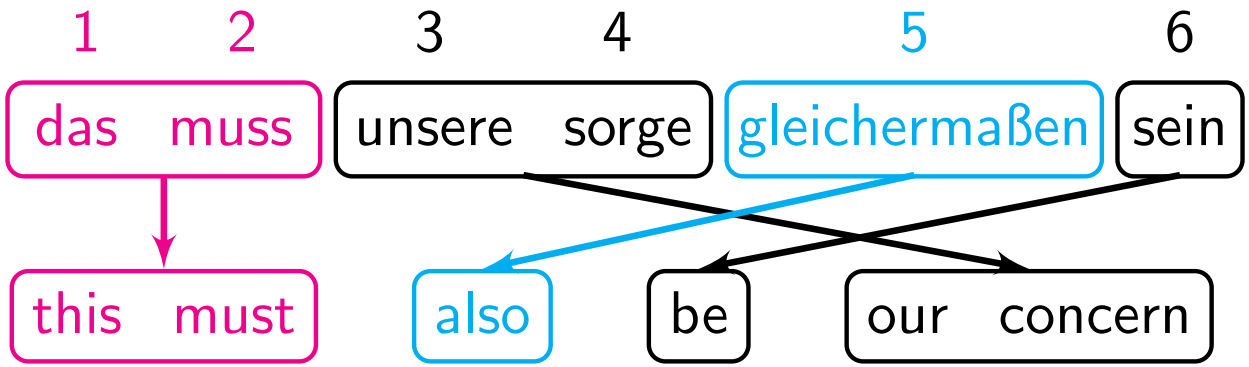
Score a derivation



- Phrase translation score: $score(\text{das muss}, \text{this must}) + \cdots$
- Language model score:

$$score(\text{<s> this must also be our concern </s>})$$
$$= score(\text{this}|\text{<s>}) + score(\text{must}|\text{this}) + \cdots$$

Score a derivation

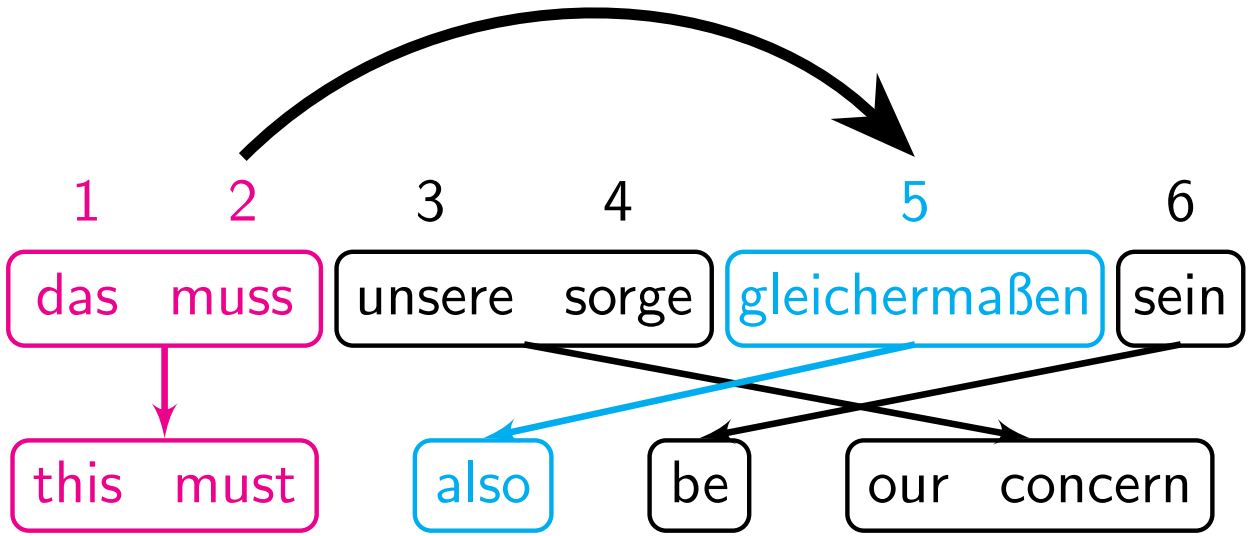


- Phrase translation score: $score(\text{das muss}, \text{this must}) + \cdots$
- Language model score:

$$score(\text{<s> this must also be our concern </s>})$$
$$= score(\text{this}|\text{<s>}) + score(\text{must}|\text{this}) + \cdots$$

- Reordering score: $\eta \cdot |2 + 1 - 5|$

Fixed distortion limit: distortion distance $\leq d$

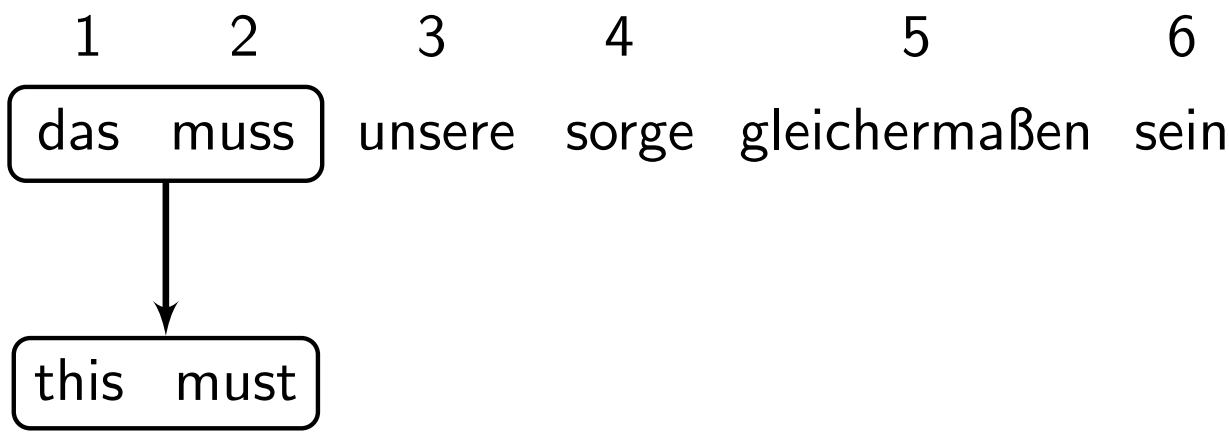


- Distortion distance: $|2 + 1 - 5| = 2$

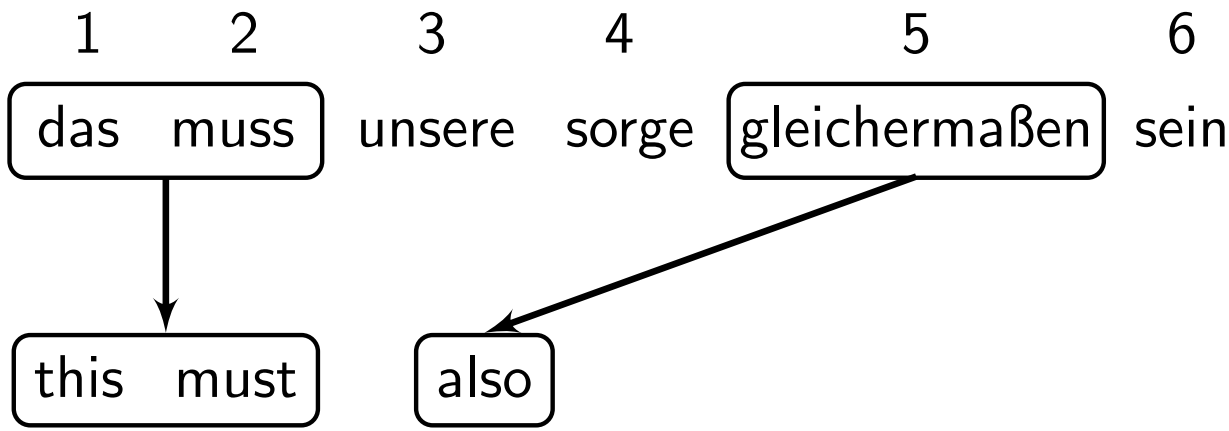
Target-side left-to-right: the usual decoding algorithm

1	2	3	4	5	6
das	muss	unsere	sorge	gleichermaßen	sein

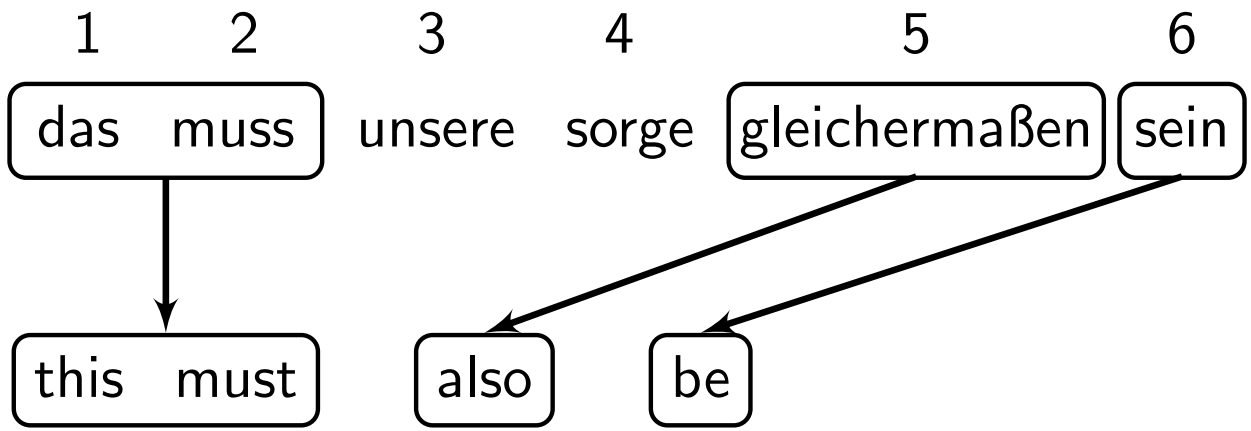
Target-side left-to-right: the usual decoding algorithm



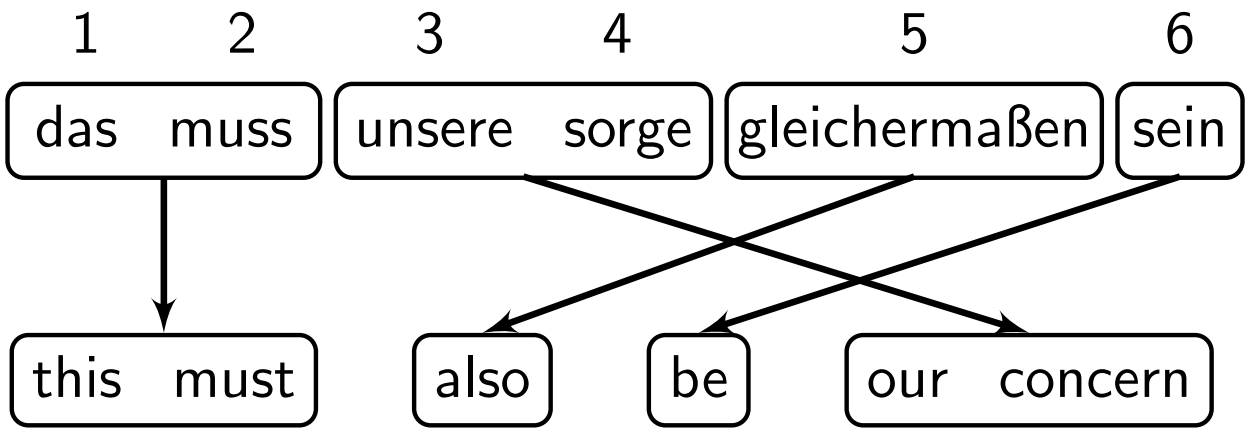
Target-side left-to-right: the usual decoding algorithm



Target-side left-to-right: the usual decoding algorithm



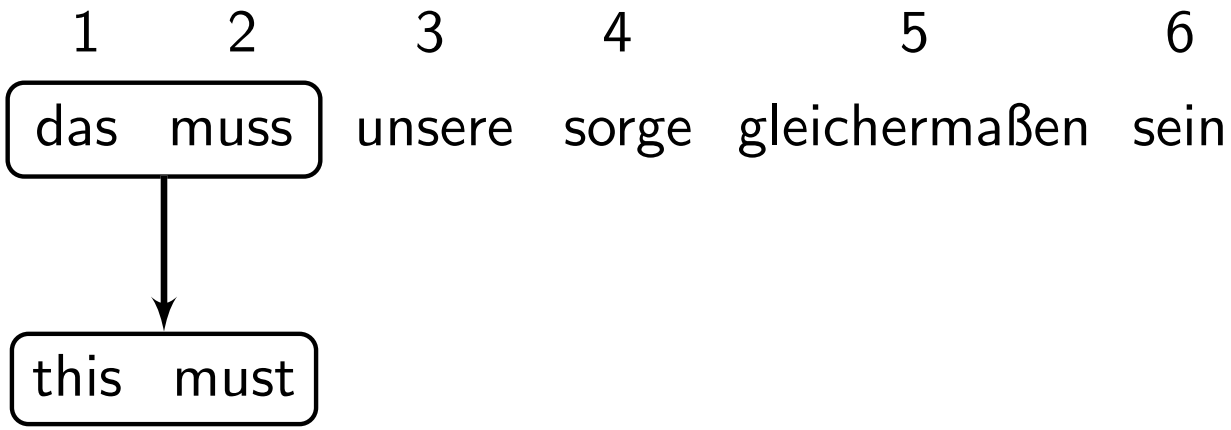
Target-side left-to-right: the usual decoding algorithm



Target-side left-to-right: dynamic programming algorithm

1	2	3	4	5	6
das	muss	unsere	sorge	gleichermaßen	sein

Target-side left-to-right: dynamic programming algorithm

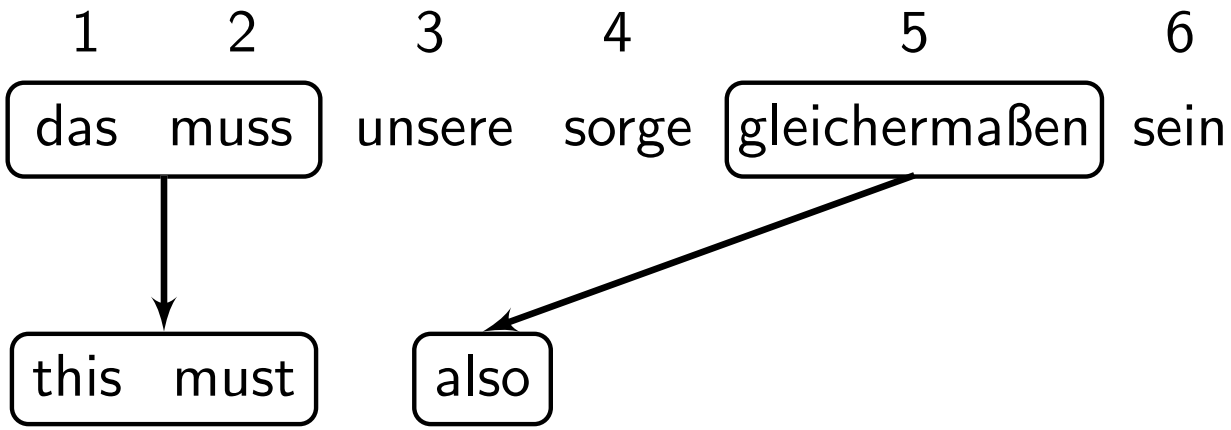


Sub-derivation:

(1, 2, this must)

DP state: (must, 2, 110000)

Target-side left-to-right: dynamic programming algorithm

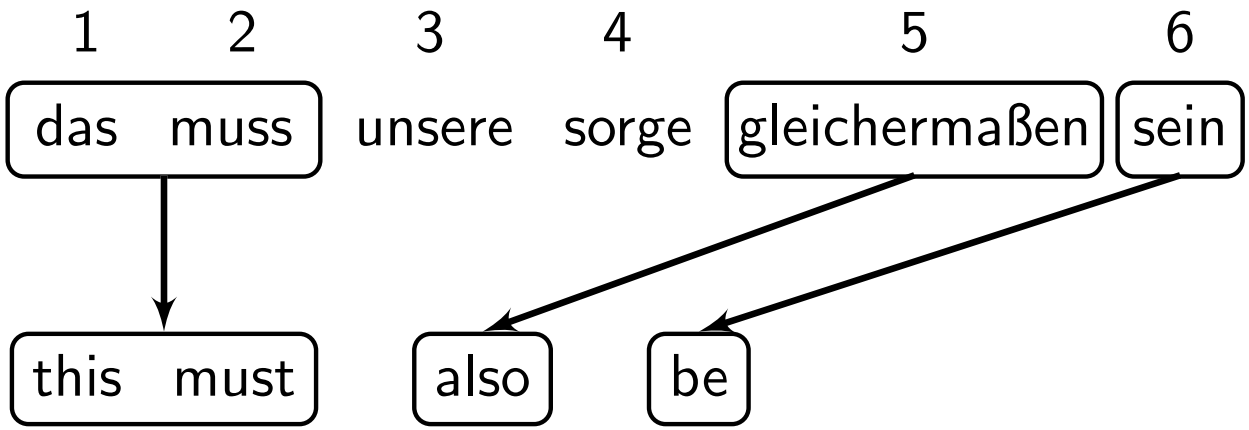


Sub-derivation:

$(1, 2, \text{this must})(5, 5, \text{also})$

DP state: $(\text{also}, 5, 110010)$

Target-side left-to-right: dynamic programming algorithm

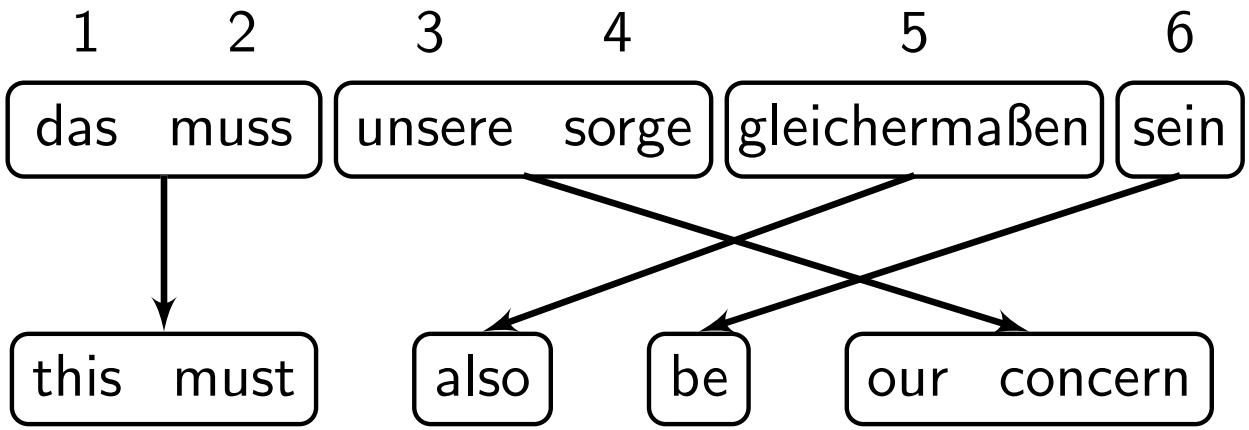


Sub-derivation:

$(1, 2, \text{this must})(5, 5, \text{also})(6, 6, \text{be})$

DP state: $(\text{be}, 6, 110011)$

Target-side left-to-right: dynamic programming algorithm



Sub-derivation:

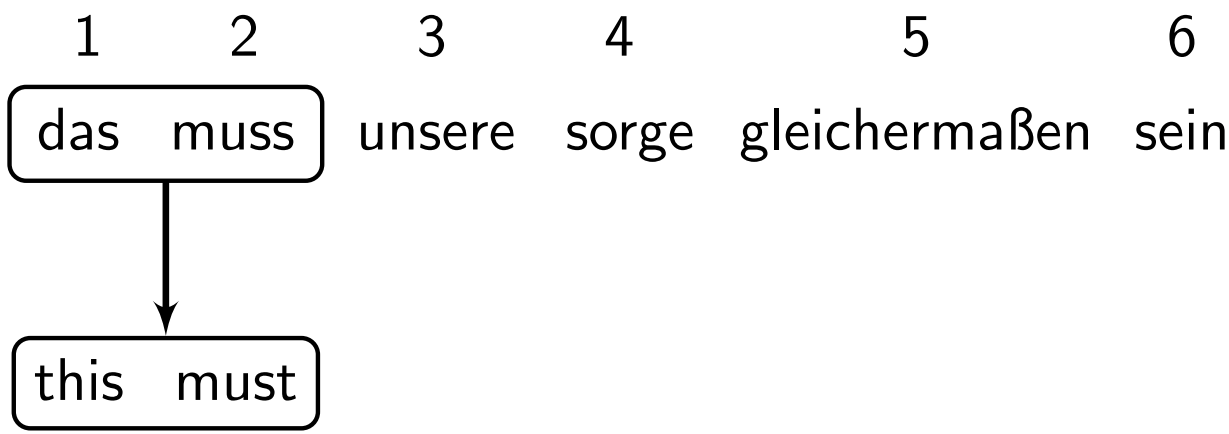
$(1, 2, \text{this must})(5, 5, \text{also})(6, 6, \text{be})(3, 4, \text{our concern})$

DP state: $(\text{concern}, 4, 111111)$

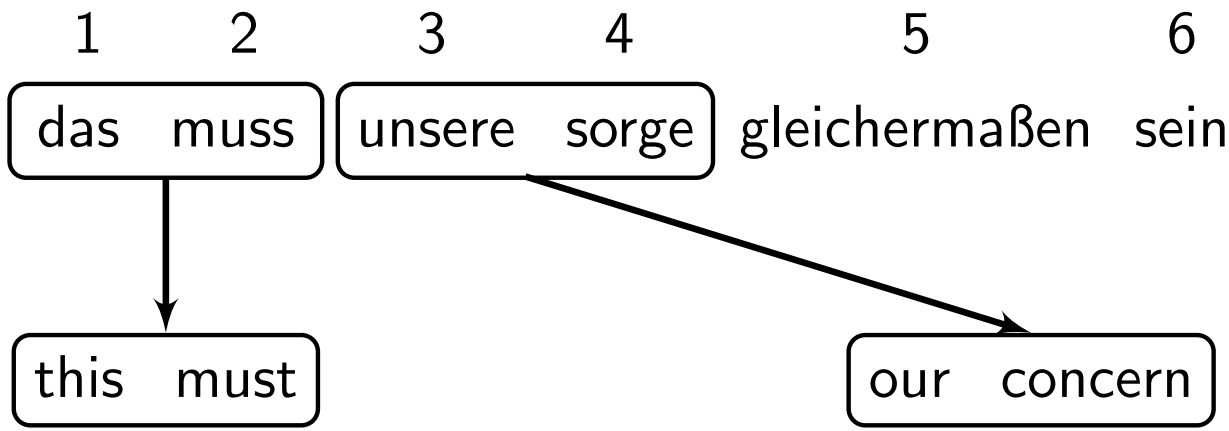
Source-side left-to-right: the proposed algorithm

1	2	3	4	5	6
das	muss	unsere	sorge	gleichermaßen	sein

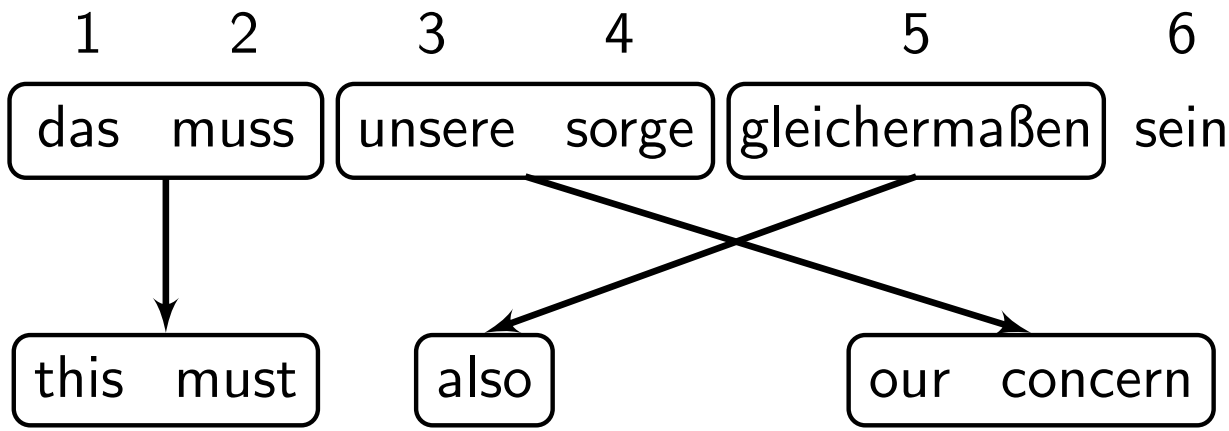
Source-side left-to-right: the proposed algorithm



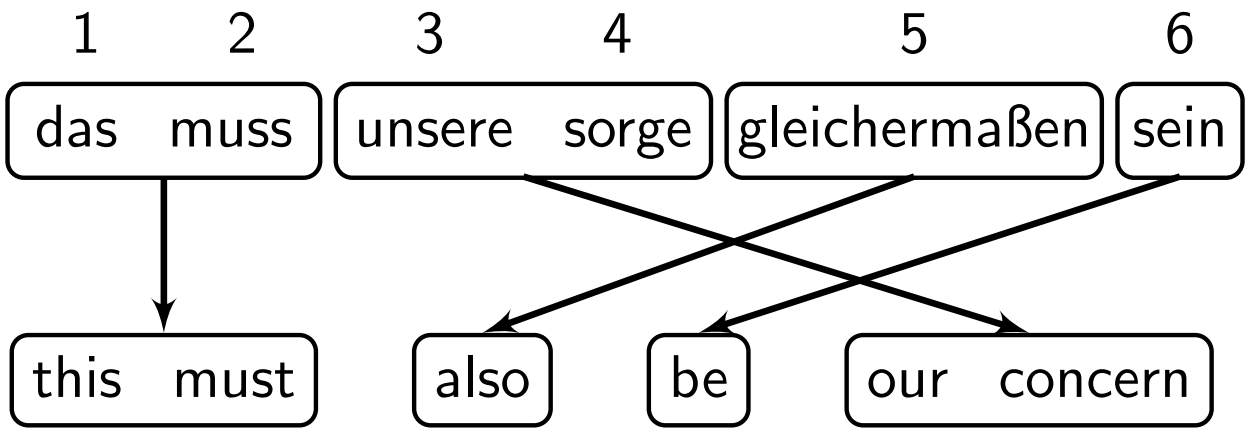
Source-side left-to-right: the proposed algorithm



Source-side left-to-right: the proposed algorithm



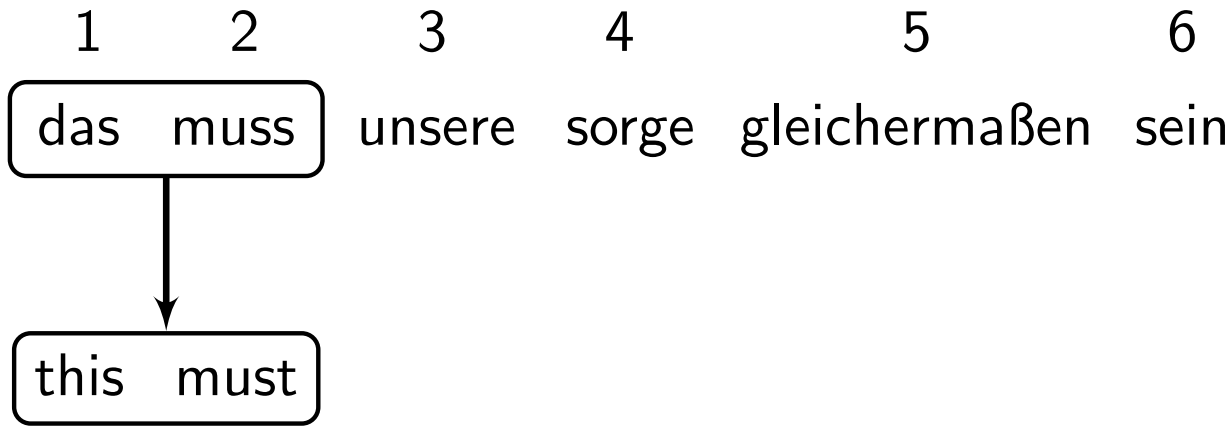
Source-side left-to-right: the proposed algorithm



Source-side left-to-right: dynamic programming state

1	2	3	4	5	6
das	muss	unsere	sorge	gleichermaßen	sein

Source-side left-to-right: dynamic programming state



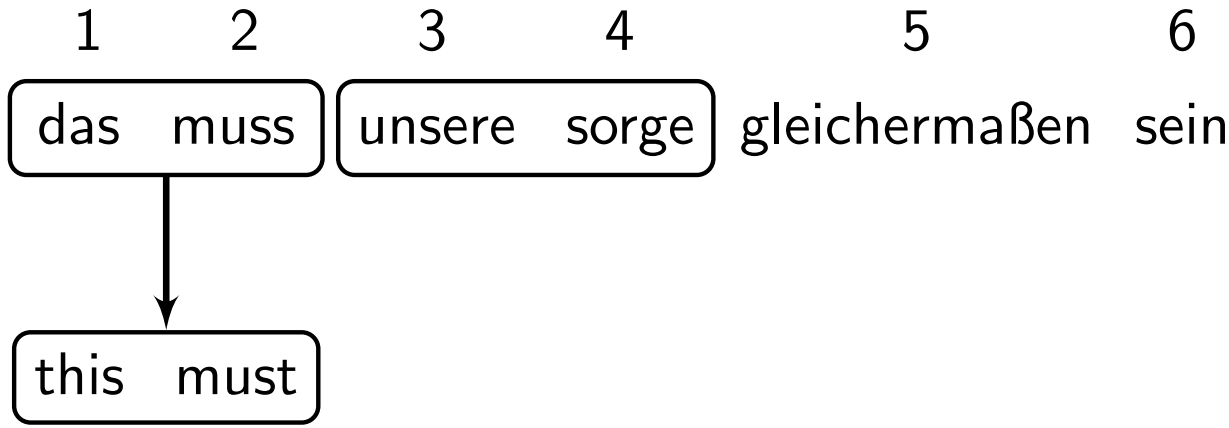
Sub-derivation:

$\pi_1 = (1, 2, \text{this must})$

DP state: $j = 2$,

$\sigma_1 = \langle 1, \text{this}, 2, \text{must} \rangle$

Source-side left-to-right: dynamic programming state



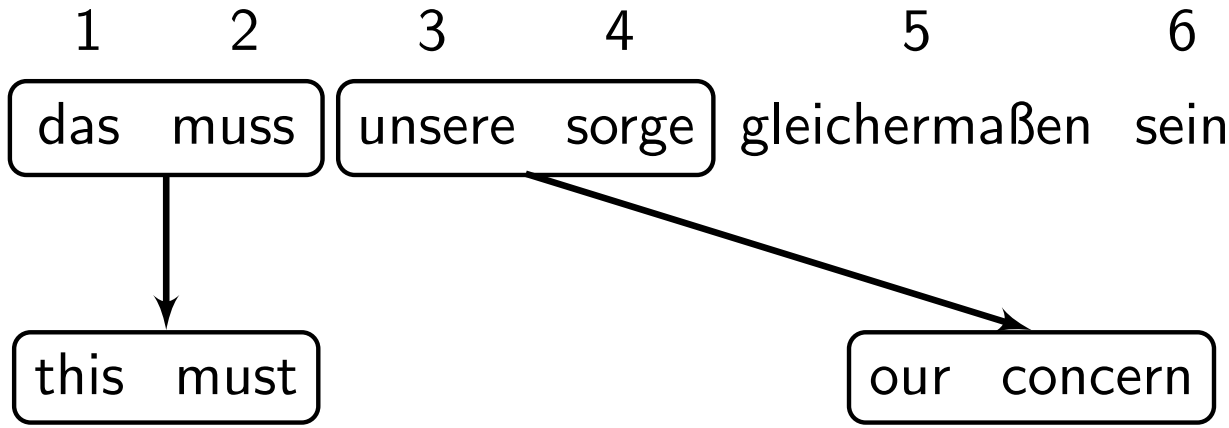
Sub-derivation:

$\pi_1 = (1, 2, \text{this must})$

DP state: $j = 2,$

$\sigma_1 = \langle 1, \text{this}, 2, \text{must} \rangle$

Source-side left-to-right: dynamic programming state



Sub-derivation:

$\pi_1 = (1, 2, \text{this must})$

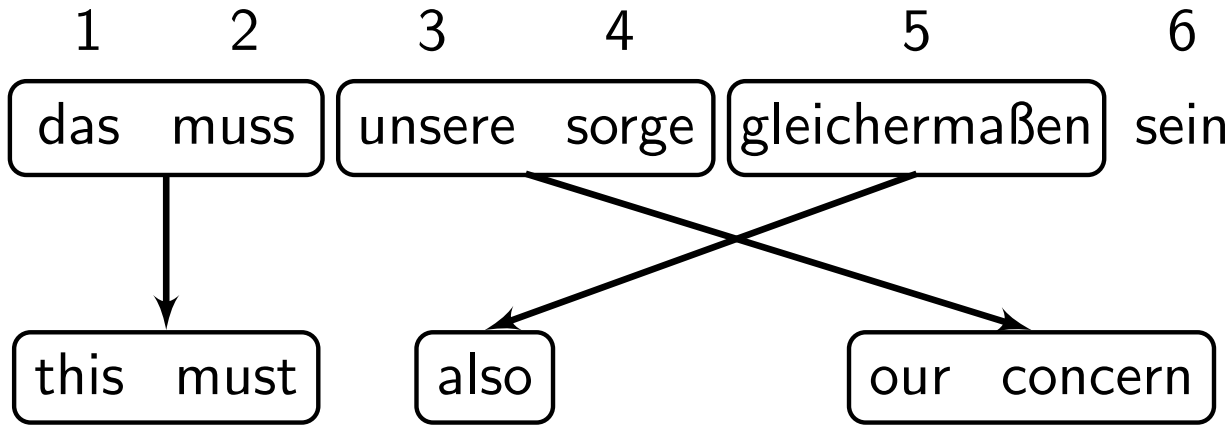
$\pi_2 = (3, 4, \text{our concern})$

DP state: $j = 4$,

$\sigma_1 = \langle 1, \text{this}, 2, \text{must} \rangle$,

$\sigma_2 = \langle 3, \text{our}, 4, \text{concern} \rangle$

Source-side left-to-right: dynamic programming state



Sub-derivation:

$\pi_1 = (1, 2, \text{this must})(5, 5, \text{also})$

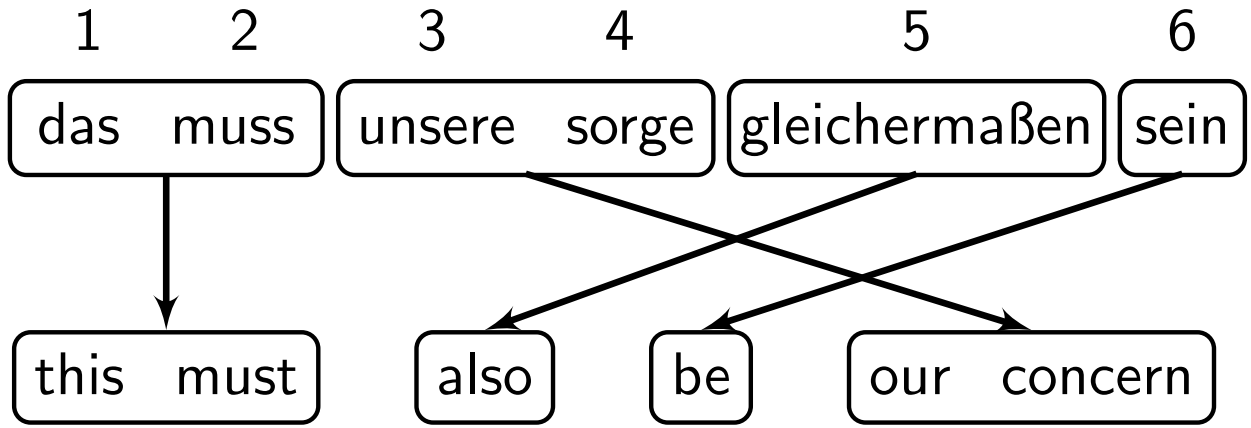
$\pi_2 = (3, 4, \text{our concern})$

DP state: $j = 5,$

$\sigma_1 = \langle 1, \text{this}, 5, \text{also} \rangle,$

$\sigma_2 = \langle 3, \text{our}, 4, \text{concern} \rangle$

Source-side left-to-right: dynamic programming state



Sub-derivation:

$\pi_1 = (1, 2, \text{this must})(5, 5, \text{also})(6, 6, \text{be})(3, 4, \text{our concern})$

DP state: $j = 6$,

$\sigma_1 = \langle 1, \text{this}, 4, \text{concern} \rangle$

The number of DP states (fixed distortion limit d)

State: $\left(j, \{\sigma_1, \sigma_2 \ldots \sigma_r\}\right)$ r: number of "tapes"

- $j \in \{1, \ldots, n\}$ n: source sentence length

The number of DP states (fixed distortion limit d)

State: $\left(j, \{\sigma_1, \sigma_2 \ldots \sigma_r\}\right)$ r: number of "tapes"

- $j \in \{1, \ldots, n\}$ n: source sentence length $\rightarrow O(n)$

The number of DP states (fixed distortion limit d)

State: $\left(j, \{\sigma_1, \sigma_2 \ldots \sigma_r\}\right)$ r: number of "tapes"

- $j \in \{1, \ldots, n\}$ n: source sentence length $\rightarrow O(n)$
- $\sigma = (s, w_s, t, w_t)$ Ex: $\sigma = (1, \text{this}, 5, \text{also})$
  $= (s(\sigma), w_s(\sigma), t(\sigma), w_t(\sigma))$

The number of DP states (fixed distortion limit d)

State: $\left(j, \{\sigma_1, \sigma_2 \ldots \sigma_r\}\right)$ r: number of "tapes"

- $j \in \{1, \ldots, n\}$ n: source sentence length $\rightarrow O(n)$
- $\sigma = (s, w_s, t, w_t)$ Ex: $\sigma = (1, \text{this}, 5, \text{also})$
  $= (s(\sigma), w_s(\sigma), t(\sigma), w_t(\sigma))$
- s, t: source word indices

The number of DP states (fixed distortion limit d)

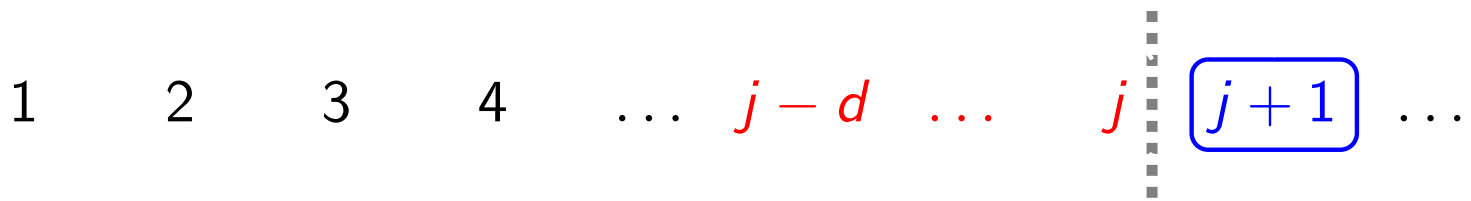
State: $\left(j, \{\sigma_1, \sigma_2 \ldots \sigma_r\}\right)$ r: number of "tapes"

- $j \in \{1, \ldots, n\}$ n: source sentence length $\rightarrow O(n)$
- $\sigma = (s, w_s, t, w_t)$ Ex: $\sigma = (1, \text{this}, 5, \text{also})$
  $= (s(\sigma), w_s(\sigma), t(\sigma), w_t(\sigma))$
- s, t: source word indices
- w_s, w_t: translated target words

The number of DP states (fixed distortion limit d)

State: $\left(j, \{\sigma_1, \sigma_2 \ldots \sigma_r\}\right)$ r: number of "tapes"

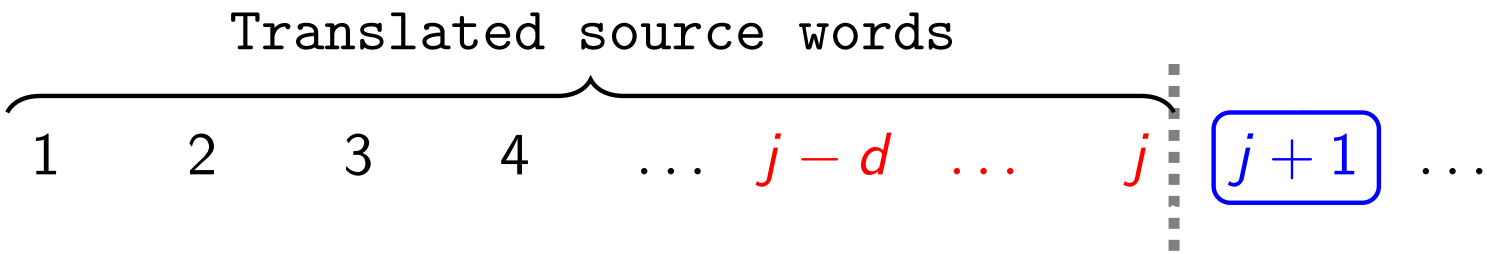
- $j \in \{1, \ldots, n\}$ n: source sentence length $\rightarrow O(n)$
- $\sigma = (s, w_s, t, w_t)$ Ex: $\sigma = (1, \text{this}, 5, \text{also})$
  $= (s(\sigma), w_s(\sigma), t(\sigma), w_t(\sigma))$



The number of DP states (fixed distortion limit d)

State: $\left(j, \{\sigma_1, \sigma_2 \ldots \sigma_r\}\right)$ r: number of "tapes"

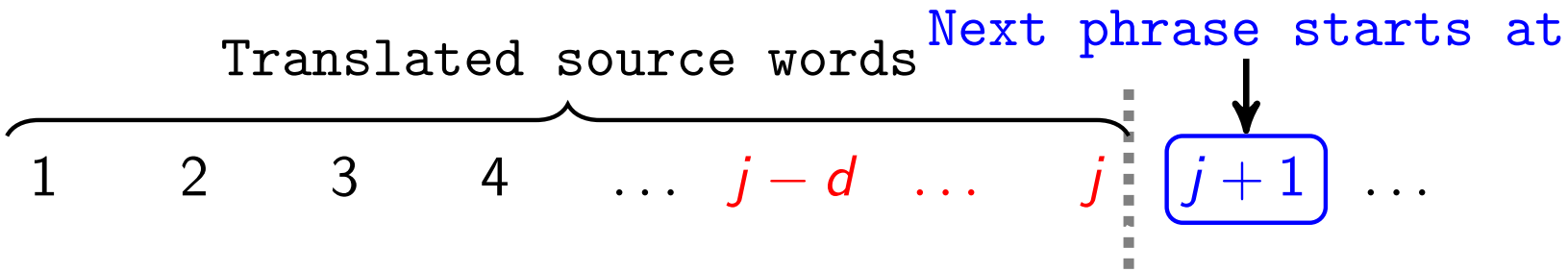
- $j \in \{1, \ldots, n\}$ n: source sentence length $\rightarrow O(n)$
- $\sigma = (s, w_s, t, w_t)$ Ex: $\sigma = (1, \text{this}, 5, \text{also})$
  $= (s(\sigma), w_s(\sigma), t(\sigma), w_t(\sigma))$



The number of DP states (fixed distortion limit d)

State: $\left(j, \{\sigma_1, \sigma_2 \ldots \sigma_r\}\right)$ r: number of "tapes"

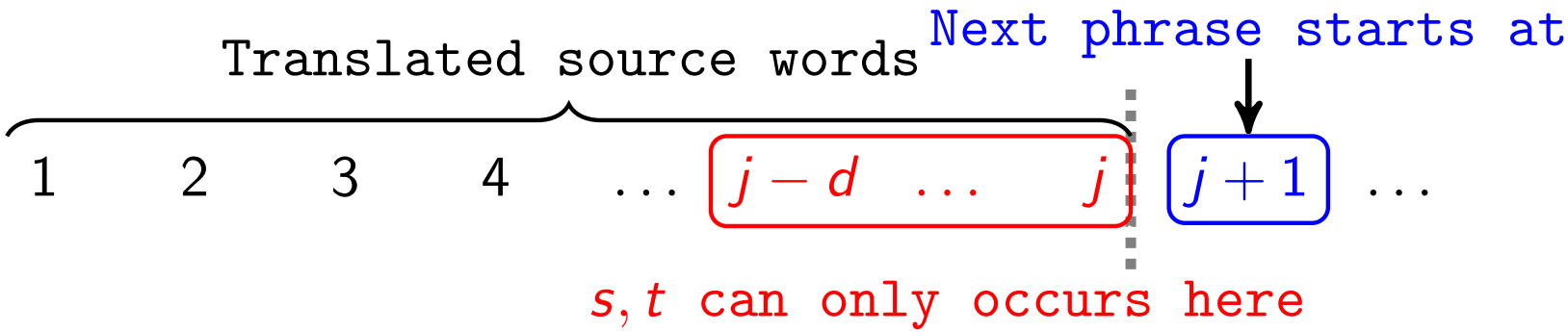
- $j \in \{1, \ldots, n\}$ n: source sentence length $\rightarrow O(n)$
- $\sigma = (s, w_s, t, w_t)$ Ex: $\sigma = (1, \text{this}, 5, \text{also})$
  $= (s(\sigma), w_s(\sigma), t(\sigma), w_t(\sigma))$



The number of DP states (fixed distortion limit d)

State: $\left(j, \{\sigma_1, \sigma_2 \ldots \sigma_r\}\right)$ r: number of "tapes"

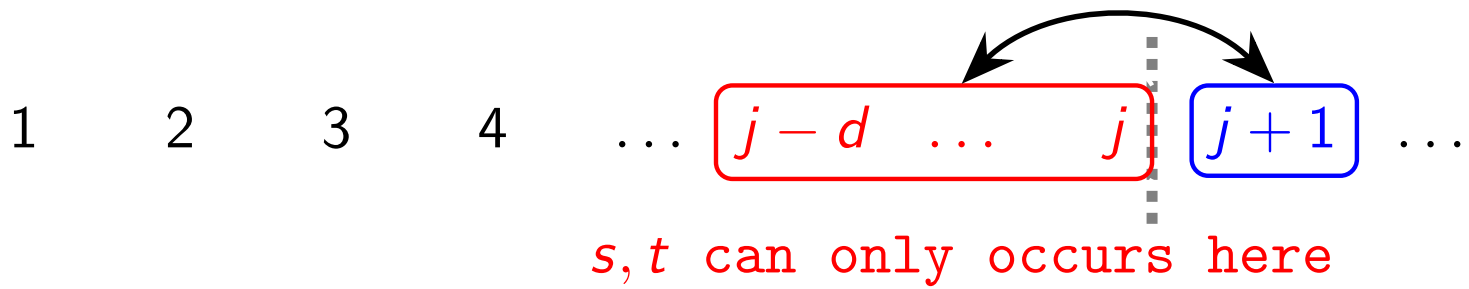
- $j \in \{1, \ldots, n\}$ n: source sentence length $\rightarrow O(n)$
- $\sigma = (s, w_s, t, w_t)$ Ex: $\sigma = (1, \text{this}, 5, \text{also})$
  $= (s(\sigma), w_s(\sigma), t(\sigma), w_t(\sigma))$



The number of DP states (fixed distortion limit d)

State: $\left(j, \{\sigma_1, \sigma_2 \ldots \sigma_r\}\right)$ r: number of "tapes"

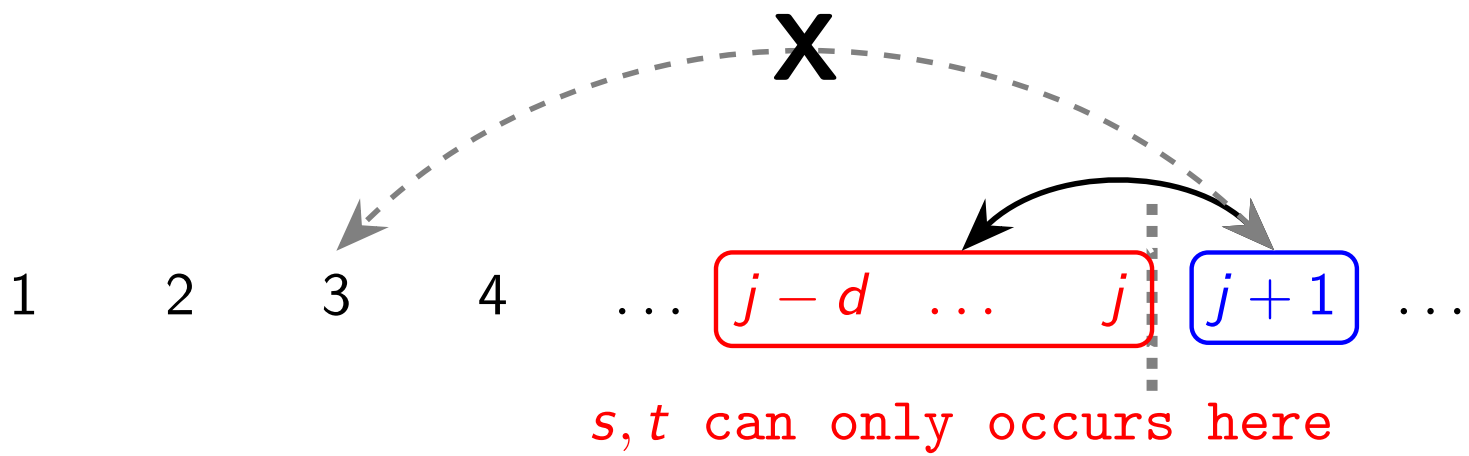
- $j \in \{1, \ldots, n\}$ n: source sentence length $\rightarrow O(n)$
- $\sigma = (s, w_s, t, w_t)$ Ex: $\sigma = (1, \text{this}, 5, \text{also})$
  $= (s(\sigma), w_s(\sigma), t(\sigma), w_t(\sigma))$



The number of DP states (fixed distortion limit d)

State: $\left(j, \{\sigma_1, \sigma_2 \ldots \sigma_r\}\right)$ r: number of "tapes"

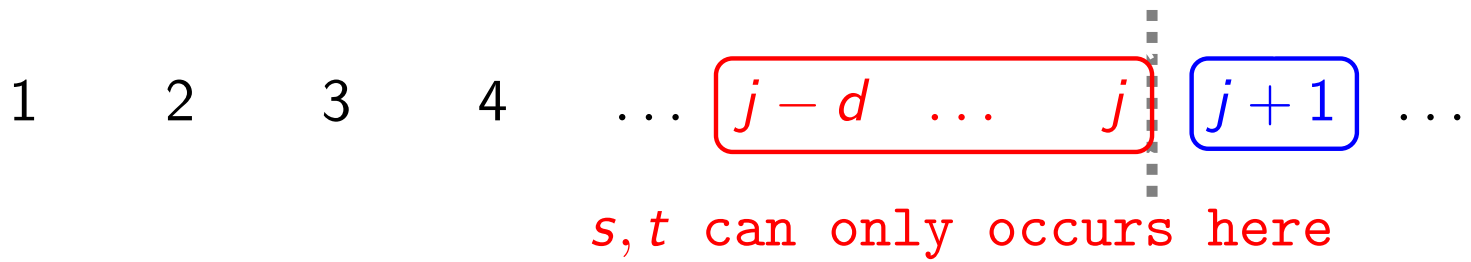
- $j \in \{1, \ldots, n\}$ n: source sentence length $\rightarrow O(n)$
- $\sigma = (s, w_s, t, w_t)$ Ex: $\sigma = (1, \text{this}, 5, \text{also})$
  $= (s(\sigma), w_s(\sigma), t(\sigma), w_t(\sigma))$



The number of DP states (fixed distortion limit d)

State: $\left(j, \{\sigma_1, \sigma_2 \ldots \sigma_r\}\right)$ r: number of "tapes"

- $j \in \{1, \ldots, n\}$ n: source sentence length $\rightarrow O(n)$
- $\sigma = (s, w_s, t, w_t)$ Ex: $\sigma = (1, \text{this}, 5, \text{also})$
  $= (s(\sigma), w_s(\sigma), t(\sigma), w_t(\sigma))$

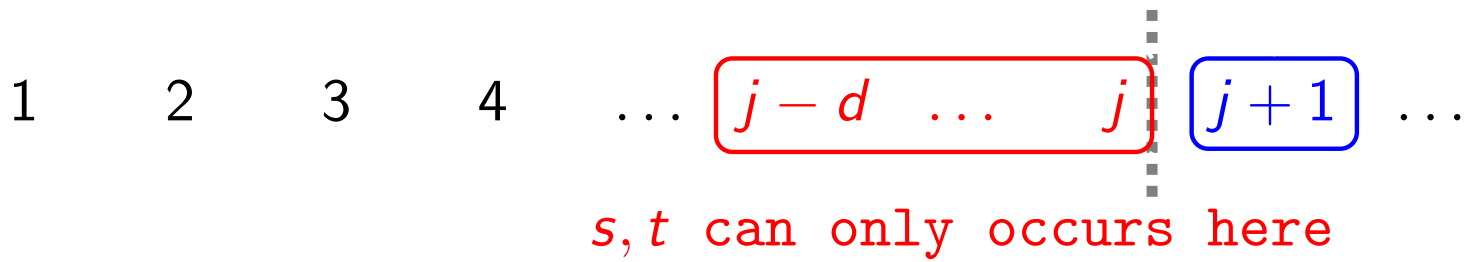


- $s(\sigma_1) = 1$
- $s(\sigma_i) \in \{j - d + 2 \ldots j\} \quad \forall i \in \{2 \ldots r\}$
- $t(\sigma_i) \in \{j - d \ldots j\} \quad \forall i \in \{1 \ldots r\}$

The number of DP states (fixed distortion limit d)

State: $\left(j, \{\sigma_1, \sigma_2 \ldots \sigma_r\}\right)$ r: number of "tapes"

- $j \in \{1, \ldots, n\}$ n: source sentence length $\rightarrow O(n)$
- $\sigma = (s, w_s, t, w_t)$ Ex: $\sigma = (1, \text{this}, 5, \text{also})$
  $= (s(\sigma), w_s(\sigma), t(\sigma), w_t(\sigma))$

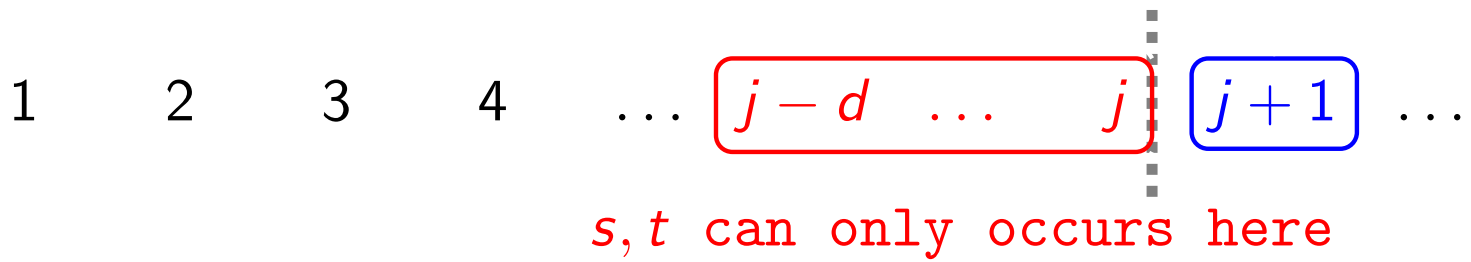


- $s(\sigma_1) = 1$
- $s(\sigma_i) \in \{j - d + 2 \ldots j\} \quad \forall i \in \{2 \ldots r\}$
- $t(\sigma_i) \in \{j - d \ldots j\} \quad \forall i \in \{1 \ldots r\}$
- r is bounded by $d + 1$.

The number of DP states (fixed distortion limit d)

State: $\left(j, \{\sigma_1, \sigma_2 \ldots \sigma_r\}\right)$ r: number of "tapes"

- $j \in \{1, \ldots, n\}$ n: source sentence length $\rightarrow O(n)$
- $\sigma = (s, w_s, t, w_t)$ Ex: $\sigma = (1, \text{this}, 5, \text{also})$
  $= (s(\sigma), w_s(\sigma), t(\sigma), w_t(\sigma)) \rightarrow O(g(d) \cdot h^{d+1})$

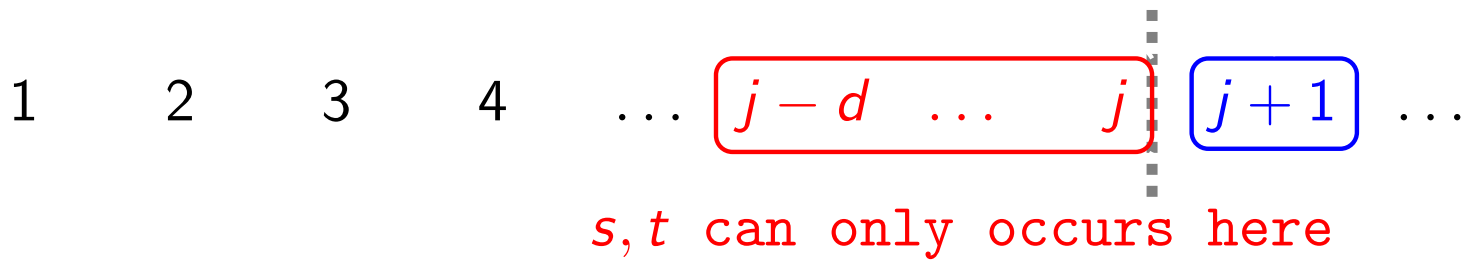


- $s(\sigma_1) = 1$
- $s(\sigma_i) \in \{j - d + 2 \ldots j\} \quad \forall i \in \{2 \ldots r\}$
- $t(\sigma_i) \in \{j - d \ldots j\} \quad \forall i \in \{1 \ldots r\}$
- r is bounded by $d + 1$.

The number of DP states (fixed distortion limit d)

State: $\left(j, \{\sigma_1, \sigma_2 \ldots \sigma_r\}\right)$ r: number of "tapes"

- $j \in \{1, \ldots, n\}$ n: source sentence length $\rightarrow O(n)$
- $\sigma = (s, w_s, t, w_t)$ Ex: $\sigma = (1, \text{this}, 5, \text{also})$
  $= (s(\sigma), w_s(\sigma), t(\sigma), w_t(\sigma)) \rightarrow O(\mathrm{g}(d) \cdot h^{d+1})$



- $s(\sigma_1) = 1$
- $s(\sigma_i) \in \{j - d + 2 \ldots j\} \quad \forall i \in \{2 \ldots r\}$
- $t(\sigma_i) \in \{j - d \ldots j\} \quad \forall i \in \{1 \ldots r\}$
- r is bounded by $d + 1$.

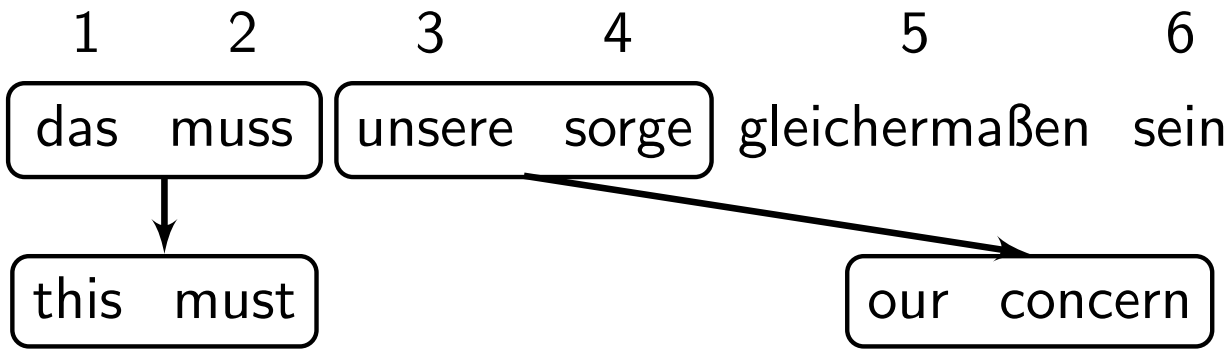
Number of states: $O(n \cdot \mathrm{g}(d) \cdot h^{d+1})$

Extend a sub-derivation by four operations

Current sub-derivation: $j, \langle \pi_1, \pi_2, \ldots \pi_r \rangle$

Consider a new phrase starting at source position $j + 1$

- **New segment** $\pi_{r+1} = \langle p \rangle$
- **Append** $\pi_i = \pi_i, p$
- **Prepend** $\pi_i = p, \pi_i$
- **Concatenate** $\pi_i = \pi_i, p, \pi_{i'}$



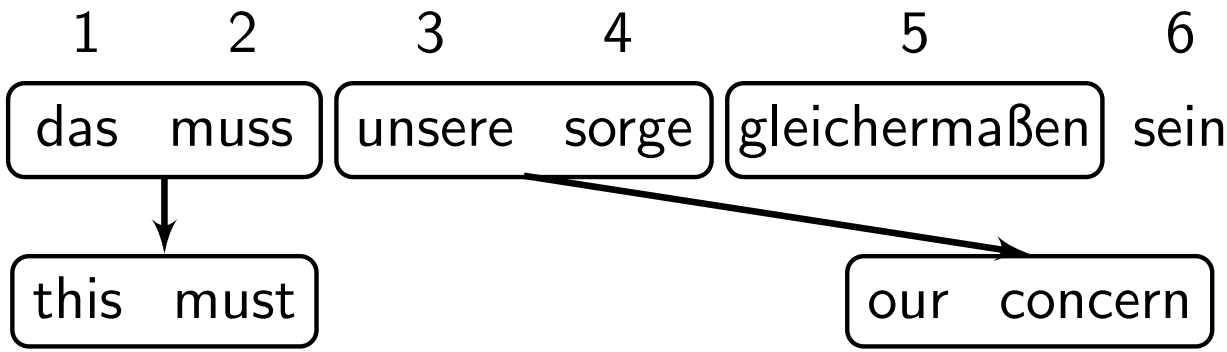
Sub-derivation: $\pi_1 = (1, 2, \text{this must})$
$\pi_2 = (3, 4, \text{our concern})$

Extend a sub-derivation by four operations

Current sub-derivation: $j, \langle \pi_1, \pi_2, \ldots \pi_r \rangle$

Consider a new phrase starting at source position $j + 1$

- **New segment** $\pi_{r+1} = \langle p \rangle$
- **Append** $\pi_i = \pi_i, p$
- **Prepend** $\pi_i = p, \pi_i$
- **Concatenate** $\pi_i = \pi_i, p, \pi_{i'}$



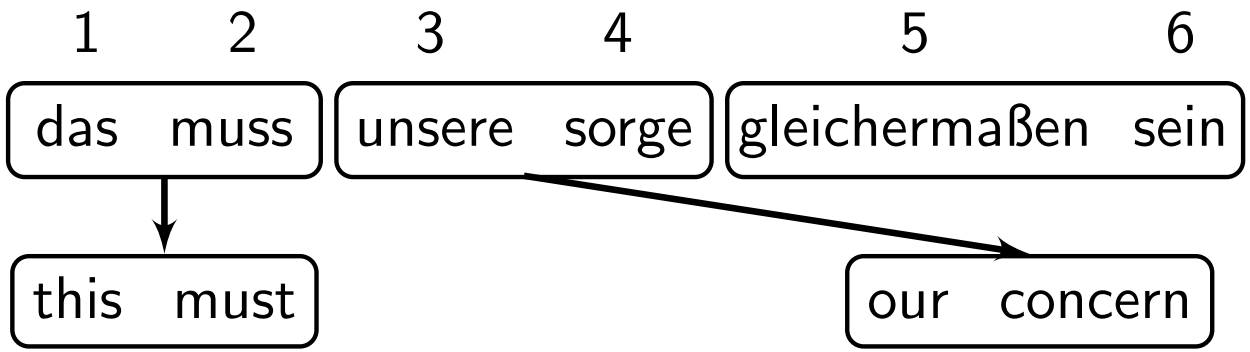
Sub-derivation: $\pi_1 = (1, 2, \text{this must})$
$\pi_2 = (3, 4, \text{our concern})$

Extend a sub-derivation by four operations

Current sub-derivation: $j, \langle \pi_1, \pi_2, \ldots \pi_r \rangle$

Consider a new phrase starting at source position $j + 1$

- **New segment** $\pi_{r+1} = \langle p \rangle$
- **Append** $\pi_i = \pi_i, p$
- **Prepend** $\pi_i = p, \pi_i$
- **Concatenate** $\pi_i = \pi_i, p, \pi_{i'}$



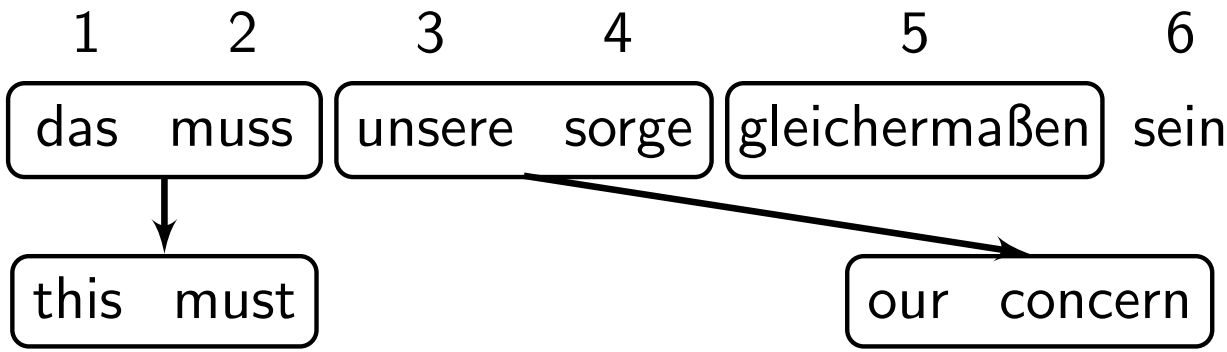
Sub-derivation: $\pi_1 = (1, 2, \text{this must})$
$\pi_2 = (3, 4, \text{our concern})$

Extend a sub-derivation by four operations

Current sub-derivation: $j, \langle \pi_1, \pi_2, \ldots \pi_r \rangle$

Consider a new phrase starting at source position $j + 1$

- **New segment** $\pi_{r+1} = \langle p \rangle$
- **Append** $\pi_i = \pi_i, p$
- **Prepend** $\pi_i = p, \pi_i$
- **Concatenate** $\pi_i = \pi_i, p, \pi_{i'}$



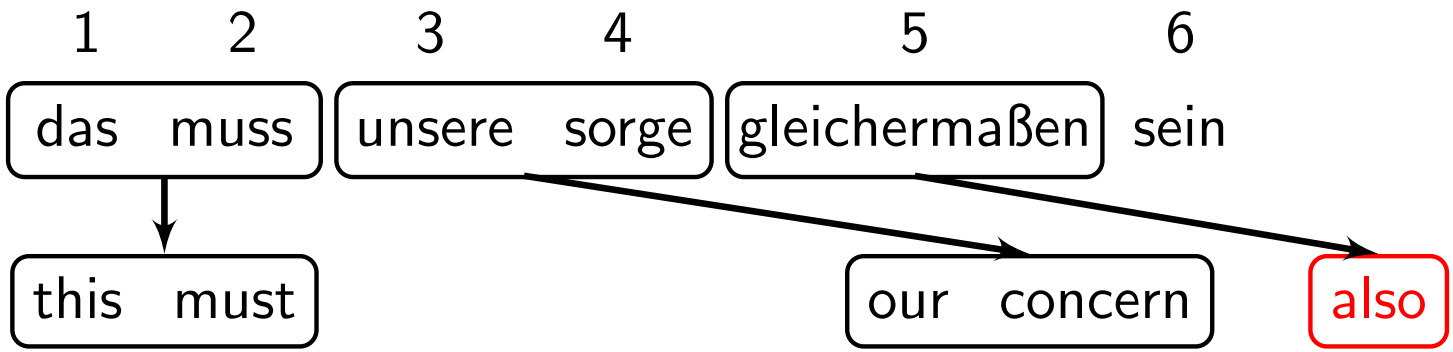
Sub-derivation: $\pi_1 = (1, 2, \text{this must})$
$\pi_2 = (3, 4, \text{our concern})$

Extend a sub-derivation by four operations

Current sub-derivation: $j, \langle \pi_1, \pi_2, \ldots \pi_r \rangle$

Consider a new phrase starting at source position $j + 1$

- **New segment** $\pi_{r+1} = \langle p \rangle$
- **Append** $\pi_i = \pi_i, p$
- **Prepend** $\pi_i = p, \pi_i$
- **Concatenate** $\pi_i = \pi_i, p, \pi_{i'}$



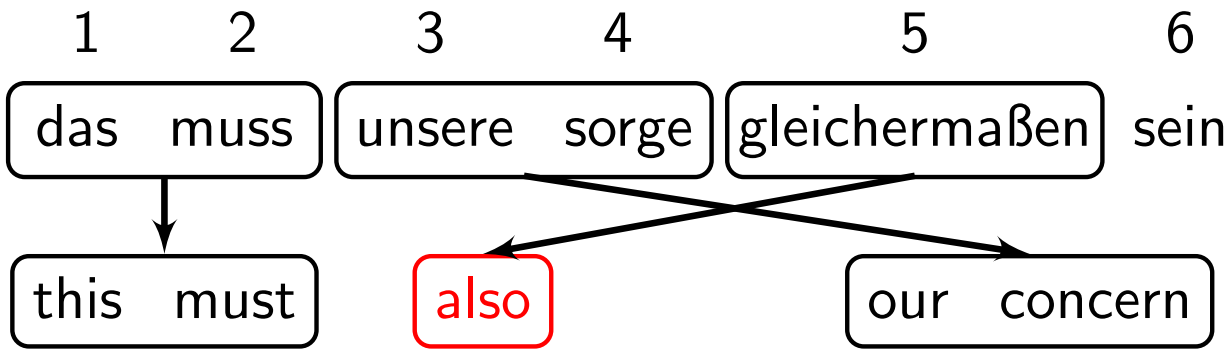
Sub-derivation: $\pi_1 = (1, 2, \text{this must})$
$\pi_2 = (3, 4, \text{our concern})$
$\pi_3 = (5, 5, \text{also})$

Extend a sub-derivation by four operations

Current sub-derivation: $j, \langle \pi_1, \pi_2, \ldots \pi_r \rangle$

Consider a new phrase starting at source position $j + 1$

- **New segment** $\pi_{r+1} = \langle p \rangle$
- **Append** $\pi_i = \pi_i, p$
- **Prepend** $\pi_i = p, \pi_i$
- **Concatenate** $\pi_i = \pi_i, p, \pi_{i'}$



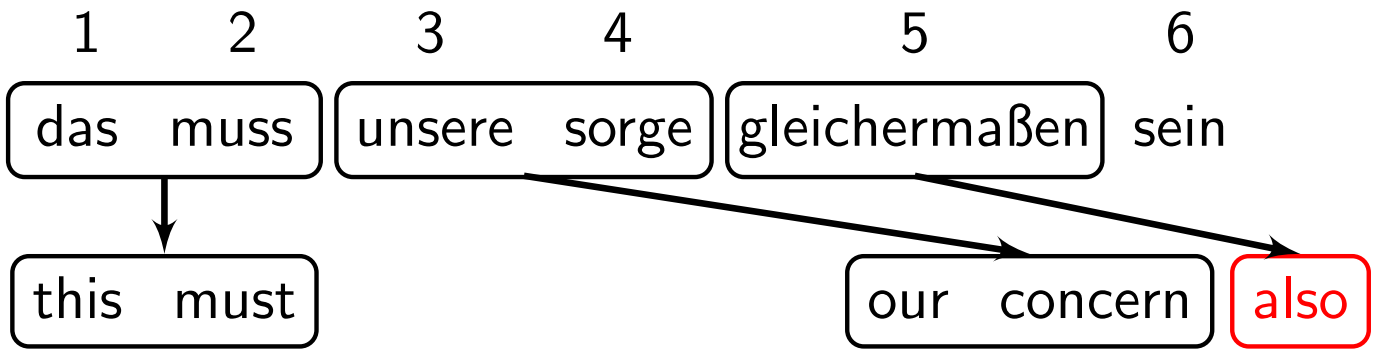
Sub-derivation: $\pi_1 = (1, 2, \text{this must})(5, 5, \text{also})$
$\pi_2 = (3, 4, \text{our concern})$

Extend a sub-derivation by four operations

Current sub-derivation: $j, \langle \pi_1, \pi_2, \ldots \pi_r \rangle$

Consider a new phrase starting at source position $j + 1$

- **New segment** $\pi_{r+1} = \langle p \rangle$
- **Append** $\pi_i = \pi_i, p$
- **Prepend** $\pi_i = p, \pi_i$
- **Concatenate** $\pi_i = \pi_i, p, \pi_{i'}$



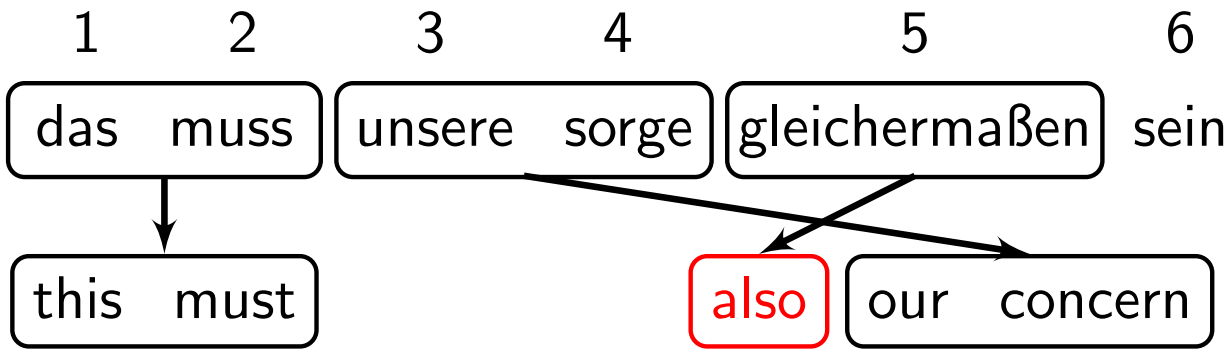
Sub-derivation: $\pi_1 = (1, 2, \text{this must})$
$\pi_2 = (3, 4, \text{our concern})(5, 5, \text{also})$

Extend a sub-derivation by four operations

Current sub-derivation: $j, \langle \pi_1, \pi_2, \ldots \pi_r \rangle$

Consider a new phrase starting at source position $j + 1$

- **New segment** $\pi_{r+1} = \langle p \rangle$
- **Append** $\pi_i = \pi_i, p$
- **Prepend** $\pi_i = p, \pi_i$
- **Concatenate** $\pi_i = \pi_i, p, \pi_{i'}$



Sub-derivation: $\pi_1 = (1, 2, \text{this must})$

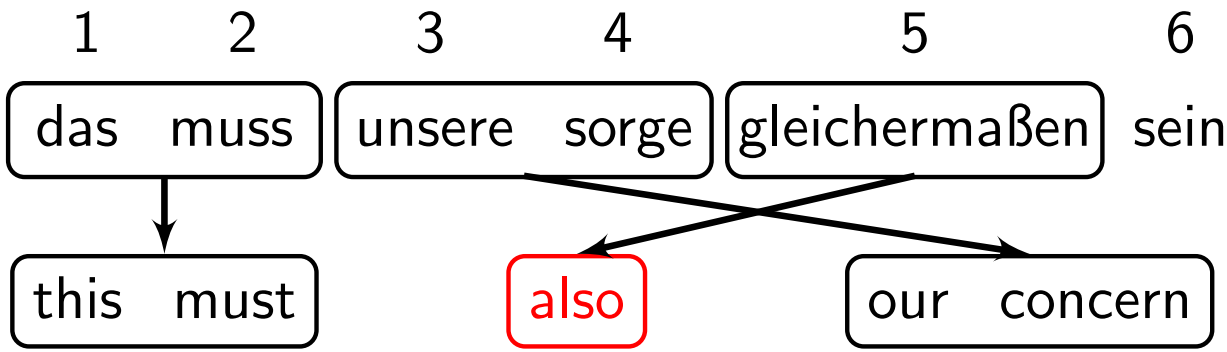
$\pi_2 = (5, 5, \text{also})(3, 4, \text{our concern})$

Extend a sub-derivation by four operations

Current sub-derivation: $j, \langle \pi_1, \pi_2, \ldots \pi_r \rangle$

Consider a new phrase starting at source position $j + 1$

- **New segment** $\pi_{r+1} = \langle p \rangle$
- **Append** $\pi_i = \pi_i, p$
- **Prepend** $\pi_i = p, \pi_i$
- **Concatenate** $\pi_i = \pi_i, p, \pi_{i'}$



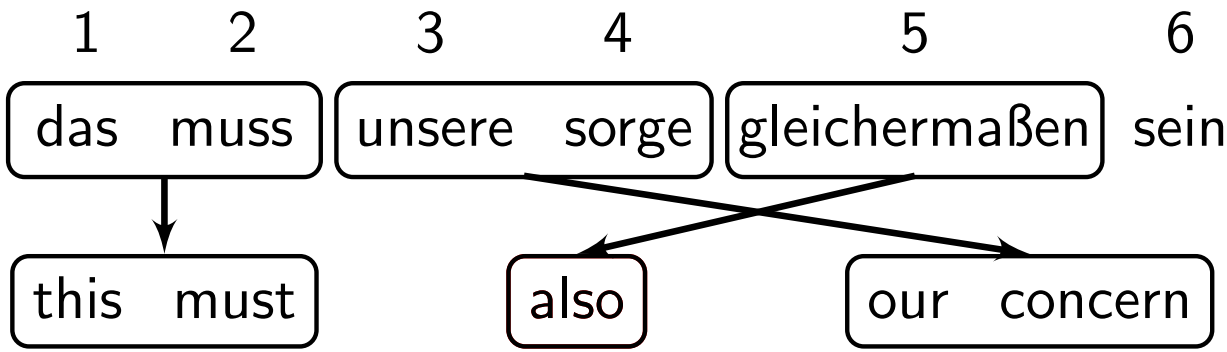
Sub-derivation: $\pi_1 = (1, 2, \text{this must})(5, 5, \text{also})(3, 4, \text{our concern})$

Extend a sub-derivation by four operations

Current sub-derivation: $j, \langle \pi_1, \pi_2, \ldots \pi_r \rangle$

Consider a new phrase starting at source position $j + 1 \rightarrow O(l)$

- **New segment** $\pi_{r+1} = \langle p \rangle$
- **Append** $\pi_i = \pi_i, p$
- **Prepend** $\pi_i = p, \pi_i$
- **Concatenate** $\pi_i = \pi_i, p, \pi_{i'} \rightarrow O(r^2) = O(d^2)$



Sub-derivation: $\pi_1 = (1, 2, \text{this must})(5, 5, \text{also})(3, 4, \text{our concern})$

Bound on running time $O(nd!lh^{d+1})$

 # DP states: $O(n \cdot g(d) \cdot h^{d+1})$
 # transition: $O(d^2 \cdot l)$

- n: source sentence length
- d: distortion limit
- l: bound on the number of phrases starting at any position
- h: bound on the maximum number of target translations for any source word

Summary

Problem: Phrase-based decoding with a fixed distortion limit

- A new decoding algorithm with $O(nd!lh^{d+1})$ time
- Operate from left to right on the source side
- Maintain multiple "tapes" on the target side

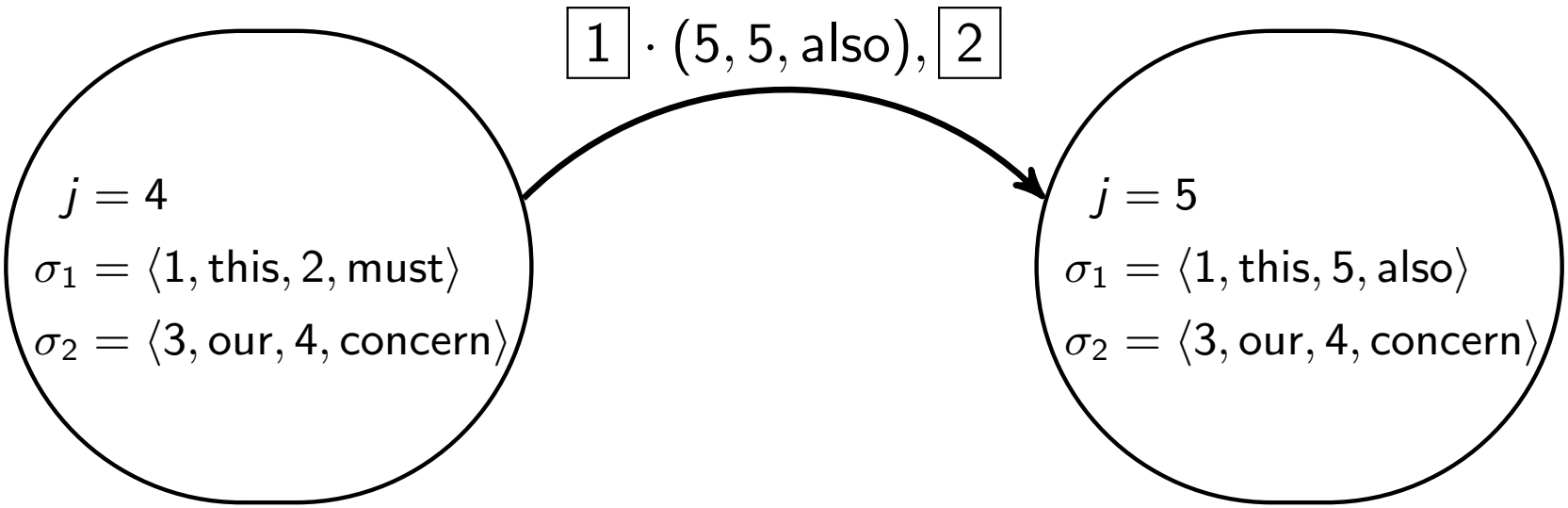
Follow-up paper in EMNLP discussing experimental results

To appear in EMNLP 2017:
"Source-side left-to-right or target-side left-to-right?
An empirical comparison of two phrase-based decoding algorithms"

- Beam search with a trigram language model
- Constraints on the number of "tapes"
- Achieve similar efficiency and accuracy as Moses

Future work

Finite state transducer (FST) formulation



Neural machine translation

- An NMT system using this kind of approach?
- Replace the attention model by absolving source words strictly left-to-right?