

# **Softwareprojektpraktikum Maschinelle Übersetzung**

**Matthias Huck, Markus Freitag**  
**{huck,freitag}@i6.informatik.rwth-aachen.de**

**Vorbesprechung 4. Aufgabe 26. Mai 2011**

**Human Language Technology and Pattern Recognition  
Lehrstuhl für Informatik 6  
Computer Science Department  
RWTH Aachen University, Germany**

# Outline

<b>1</b>	<b>Wiederholung</b>	<b>3</b>
<b>2</b>	<b>Phrasenbasierte Übersetzung</b>	<b>5</b>
<b>3</b>	<b>Phrasenextraktion: Implementierung</b>	<b>12</b>
<b>4</b>	<b>Erweiterung des Decoders und der A*-Suche auf Phrasen</b>	<b>21</b>
<b>5</b>	<b>Log-lineare Modellkombination</b>	<b>23</b>
<b>6</b>	<b>Übung 4</b>	<b>26</b>

# 1 Wiederholung

$$\begin{aligned}\hat{e}_1^I &= \arg \max_{e_1^I} \left\{ p(e_1^I | f_1^J) \right\} \\ &= \arg \max_{e_1^I} \left\{ p(e_1^I) \cdot p(f_1^J | e_1^I) \right\}\end{aligned}$$

**Wo stehen wir?**

# Verlauf des Praktikums

1. Extraktion einer Wort-zu-Wort-Übersetzungstabelle (Alignment vorgegeben)
2. Implementieren eines einzelwortbasierten Decoders,  $A^*$ -Suche für  $n$ -best Listen
3. Automatische Metriken WER, PER und BLEU
4. Phrasenextraktion, phrasenbasiertes Decoding, log-lineare Modellierung
5. Optimierung der Modellgewichte: Downhill-Simplex und MERT
6. Reranking mit  $n$ -gram Sprachmodell

## 2 Phrasenbasierte Übersetzung

- ▶ Segmentierung in zweidimensionale Blöcke
  - ▷ Wörter innerhalb einer Phrase können nicht zu Wörtern außerhalb der Phrase aligniert sein
- ▶ Ziel: Zerlegung eines Satzpaares  $(f_1^J, e_1^I)$  in Phrasenpaare  $(\tilde{f}_k, \tilde{e}_k), k = 1, \dots, K$ :

$$p(e_1^I | f_1^J) = p(\tilde{e}_1^K | \tilde{f}_1^K) = \prod_k p(\tilde{e}_k | \tilde{f}_k)$$

meal	•	•	•	■	•	•
toddler	•	•	•	•	■	■
a	•	•	■	•	•	•
order	•	■	•	•	•	•
you	■	•	•	•	•	•
did	■	•	•	•	•	•
	ha	ordinato	un	piatto	per	bambini

# Extraktion: Gültige Phrasen

- ▶ Gegeben: ein Quellsatz  $f_1^J$ , ein Zielsatz  $e_1^I$  und ein zugehöriges Alignment  $A$ .
- ▶ Eine Phrasenpaar  $(f_{j_1}^{j_2}, e_{i_1}^{i_2})$  wird als gültig angesehen, wenn
  - ▷ es mindestens ein Alignment zwischen den Phrasen gibt
  - ▷ alle Alignments nur innerhalb der Phrasen liegen, und
  - ▷ keines links, rechts, oben oder unten außerhalb
- ▶ Formal: Menge der bilingualen Phrasen  $\mathcal{BP}(f_1^J, e_1^I, A)$  des Satzpaares  $(f_1^J, e_1^I)$  bei gegebener Alignment-Matrix  $A \subseteq J \times I$  ist definiert als:

$$\mathcal{BP}(f_1^J, e_1^I, A) = \left\{ (f_{j_1}^{j_2}, e_{i_1}^{i_2}) : \forall (j, i) \in A : j_1 \leq j \leq j_2 \leftrightarrow i_1 \leq i \leq i_2 \right. \\ \left. \wedge \exists (j, i) \in A : j_1 \leq j \leq j_2 \wedge i_1 \leq i \leq i_2 \right\}$$

# Beispiel

## ► Beispiel-Alignment für das Sprachpaar Italienisch-Englisch (IWSLT)

meal	•	•	•	■	•	•
toddler	•	•	•	•	■	■
a	•	•	■	•	•	•
order	•	■	•	•	•	•
you	■	•	•	•	•	•
did	■	•	•	•	•	•
	ha	ordinato	un	piatto	per	bambini

# Beispiel








## ► Einzelwort-Paare wie in Übung 1

meal	•	•	•	<input type="checkbox"/>	•	•
toddler	•	•	•	•	<input type="checkbox"/>	<input type="checkbox"/>
a	•	•	<input type="checkbox"/>	•	•	•
order	•	<input type="checkbox"/>	•	•	•	•
you	<input type="checkbox"/>	•	•	•	•	•
did	<input type="checkbox"/>	•	•	•	•	•
	ha	ordinato	un	piatto	per	bambini



# Beispiel

## ► Jetzt ungültige Einzelwort-Paare

meal	•	•	•		•	•
toddler	•	•	•	•		
a	•	•		•	•	•
order	•		•	•	•	•
you		•	•	•	•	•
did		•	•	•	•	•
	ha	ordinato	un	piatto	per	bambini

# Beispiel

## ► Gültiges Phrasenpaar

meal	•		■	•	•
toddler	•		•	■	■
a		•	■		
order		■	•		
you	■		•	•	•
did	■		•	•	•
ha		ordinato	un	piatto	per bambini

# Beispiel

## ► Ungültiges Phrasenpaar

meal	•	•	•	■	•	•
toddler	•	•	•	•	■	■
a	•	•	■	•	•	•
order	•	■	•	•	•	•
you	■	•	•	•	•	•
did	■	•	•	•	•	•
ha		ordinato	un	piatto	per	bambini

### 3 Phrasenextraktion: Implementierung

- ▶ Verschachtelte Schleife (j1, j2) über den Quellsatz
- ▶ Ermitteln des minimalen (i1) und maximalen (i2) Wortes auf Zielseite
- ▶ Überprüfen, ob i1 und i2 ihrerseits nicht über j1 und j2 hinausgehen

```
for j1 := 0 to J-1
  for j2 := j1 to J-1
    i1 = getMinZielAlignment(j1, j2)
    i2 = getMaxZielAlignment(j1, j2)
    if (getMinQuellAlignment(i1, i2) == j1 &&
        getMaxQuellAlignment(i1, i2) == j2)
      outputGeltigePhrase(j1, j2, i1, i2)
```

# Beispiel

## ► Beispiel für $j_1 = 0$ und $j_2 = 2$

meal	•	•	•	■	•	•
toddler	•	•	•	•	■	■
a	•	•	■	•	•	•
order	•	■	•	•	•	•
you	■	•	•	•	•	•
did	■	•	•	•	•	•
ha		ordinato	un	piatto	per	bambini
	↑		↑			
	j1		j2			

# Beispiel

- ▶ Beispiel für  $j_1 = 0$  und  $j_2 = 2$
- ▶ Ermitteln von  $i_1$  und  $i_2$

meal	•	•	•	■	•	•
toddler	•	•	•	•	■	■
a	•	•	■	•	•	•
order	•	■	•	•	•	•
you	■	•	•	•	•	•
did	■	•	•	•	•	•
	ha	ordinato	un	piatto	per	bambini
	↑		↑			
	j1		j2			

# Beispiel

- ▶ Beispiel für  $j_1 = 0$  und  $j_2 = 2$
- ▶ Ermitteln von  $i_1$  und  $i_2$
- ▶ Gültige Phrase: ha ordinato un # did you order a

	meal			■	•	•
	toddler			•	■	■
i2 →	a	•	•	■		
	order	•	■	•		
	you	■	•	•		
i1 →	did	■	•	•		
	ha		ordinato	un	piatto	per bambini
		↑		↑		
		j1		j2		

# Beispiel

## ► Nächste Iteration:

$$j_1 = 0 \text{ und } j_2 = j_2 + 1 = 3$$

meal	•	•	•	■	•	•
toddler	•	•	•	•	■	■
a	•	•	■	•	•	•
order	•	■	•	•	•	•
you	■	•	•	•	•	•
did	■	•	•	•	•	•
	ha	ordinato	un	piatto	per	bambini
	↑			↑		
	j1			j2		



# Beispiel

► Nächste Iteration:

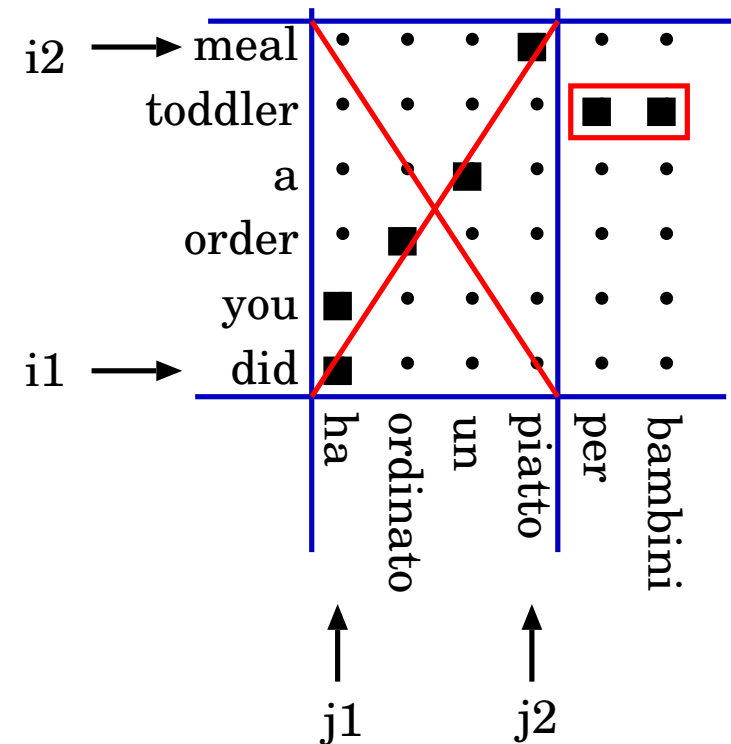
$$j_1 = 0 \text{ und } j_2 = j_2 + 1 = 3$$

► Ermitteln von  $i_1$  und  $i_2$

i2 →	meal	•	•	•	■	•	•
	toddler	•	•	•	•	■	■
	a	•	•	■	•	•	•
	order	•	■	•	•	•	•
	you	■	•	•	•	•	•
i1 →	did	■	•	•	•	•	•
	ha		ordinato	un	piatto	per	bambini
		↑			↑		
		j1			j2		

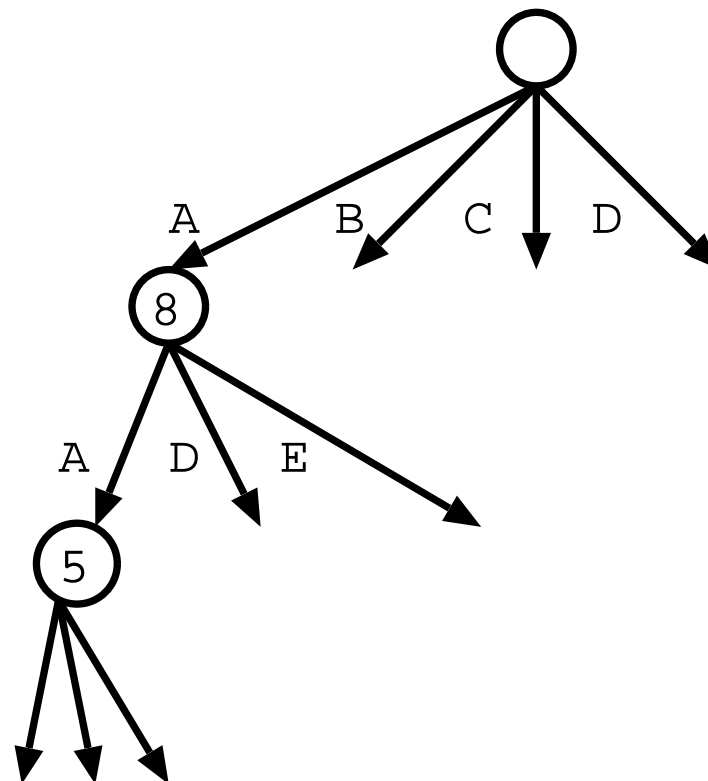
# Beispiel

- ▶ **Nächste Iteration:**  
 $j_1 = 0$  und  $j_2 = j_2 + 1 = 3$
- ▶ **Ermitteln von  $i_1$  und  $i_2$**
- ▶ **Ungültige Phrase:**  
 ha ordinato un piatto # did you order a toddler meal



# Präfixbaum

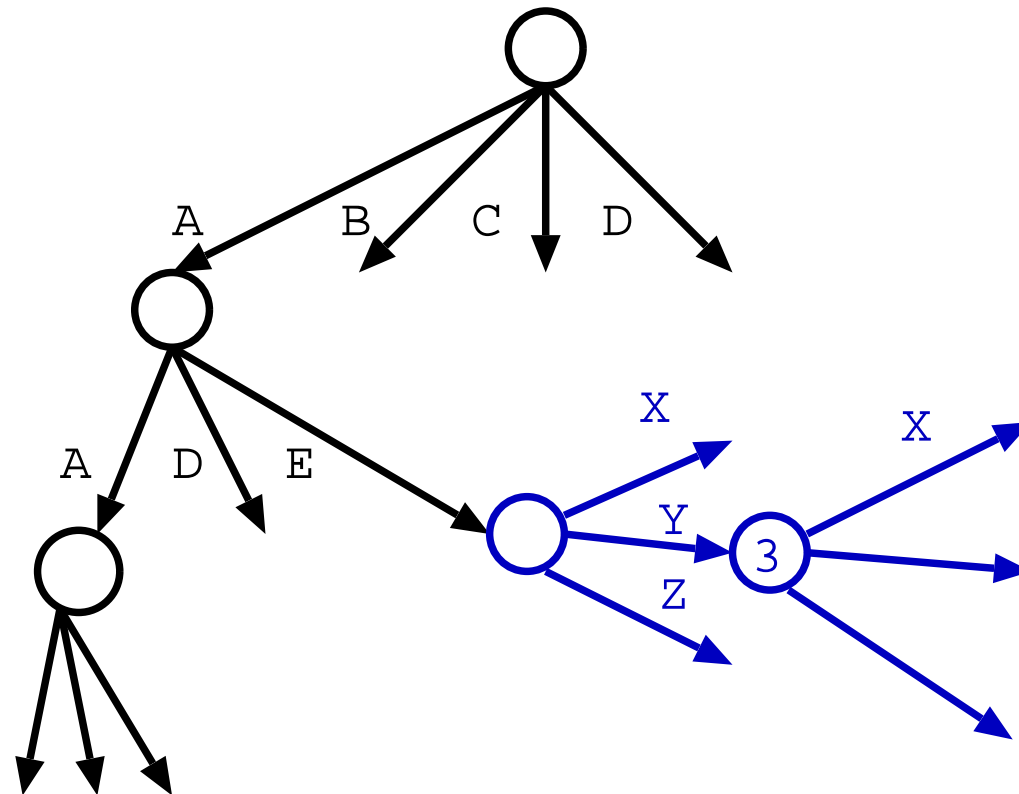
- ▶ Die Counts der Phrasen werden in Präfixbäumen abgespeichert



- ▶ Die Phrase A-A wurde 5 mal gesehen

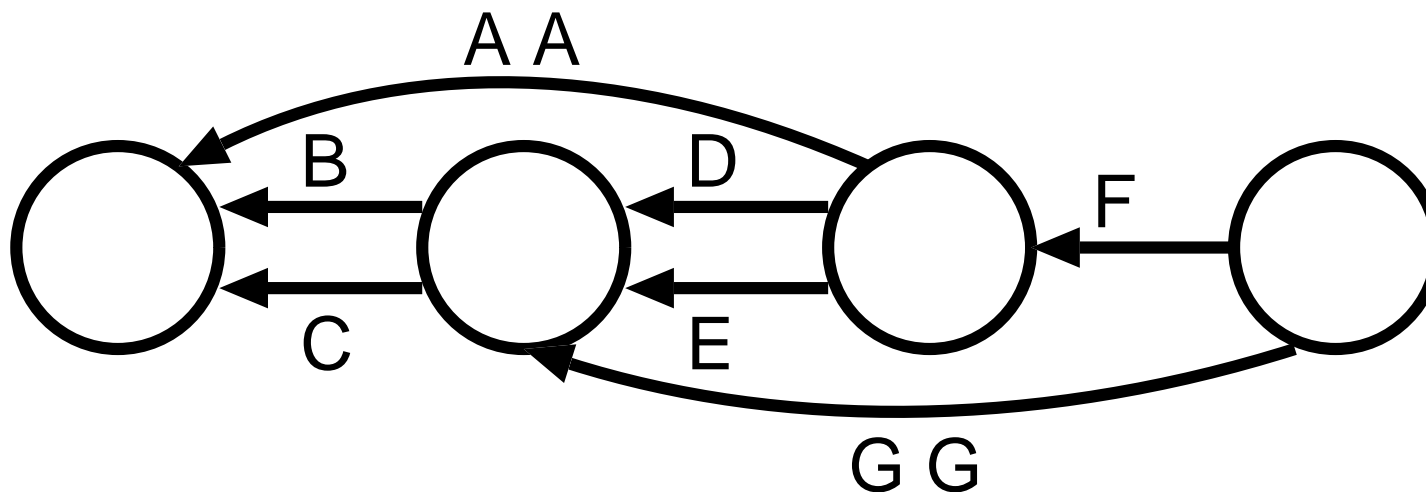
# Präfixbaum von Präfixbäumen

- ▶ Die Counts der Phrasenpaare werden in Präfixbäumen von Präfixbäumen abgespeichert



- ▶ Das Phrasenpaar A-E # Y wurde 3 mal gesehen
- ▶ Erinnerung: Template-Mechanismus in C++!

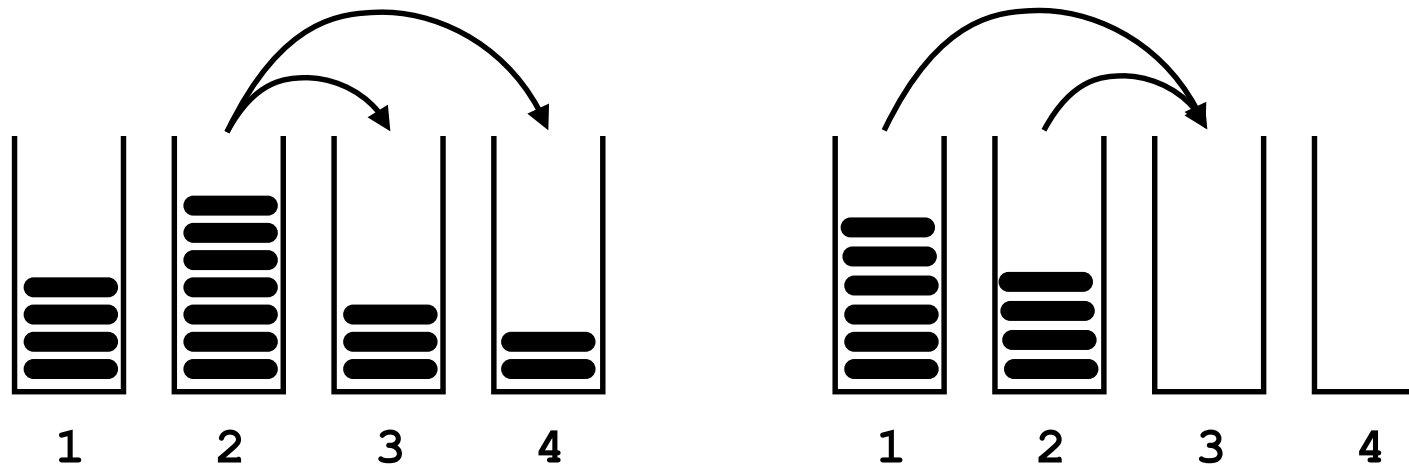
## 4 Erweiterung des Decoders und der A\*-Suche auf Phrasen



Wenn ihr alles richtig gemacht habt, ist das trivial :-)

# Phrasenbasiertes Decoding: Expansion von Teilhypothesen

Zwei Varianten:



(Illustration von Philipp Koehn)

**Besser: Variante 2**

- Warum?
- Hinweis: Pruning

## 5 Log-lineare Modellkombination

### Motivation:

- ▶ möglichst viele Wissensquellen hinzunehmen
- ▶ aber: nicht alle Modelle werden gleich zuverlässig sein
- ▶ Gewichtung der einzelnen Modelle durch Skalierungsfaktoren

### Mathematisch:

$$\hat{e}_1^I = \arg \max_{e_1^I} \{p(e_1^I | f_1^J)\} \quad (1)$$

$$= \arg \max_{e_1^I} \left\{ \frac{\exp \left( \sum_{m=1}^M \lambda_m h_m(e_1^I, f_1^J) \right)}{\sum_{\tilde{e}_1^I} \exp \left( \sum_{m=1}^M \lambda_m h_m(\tilde{e}_1^I, f_1^J) \right)} \right\} \quad (2)$$

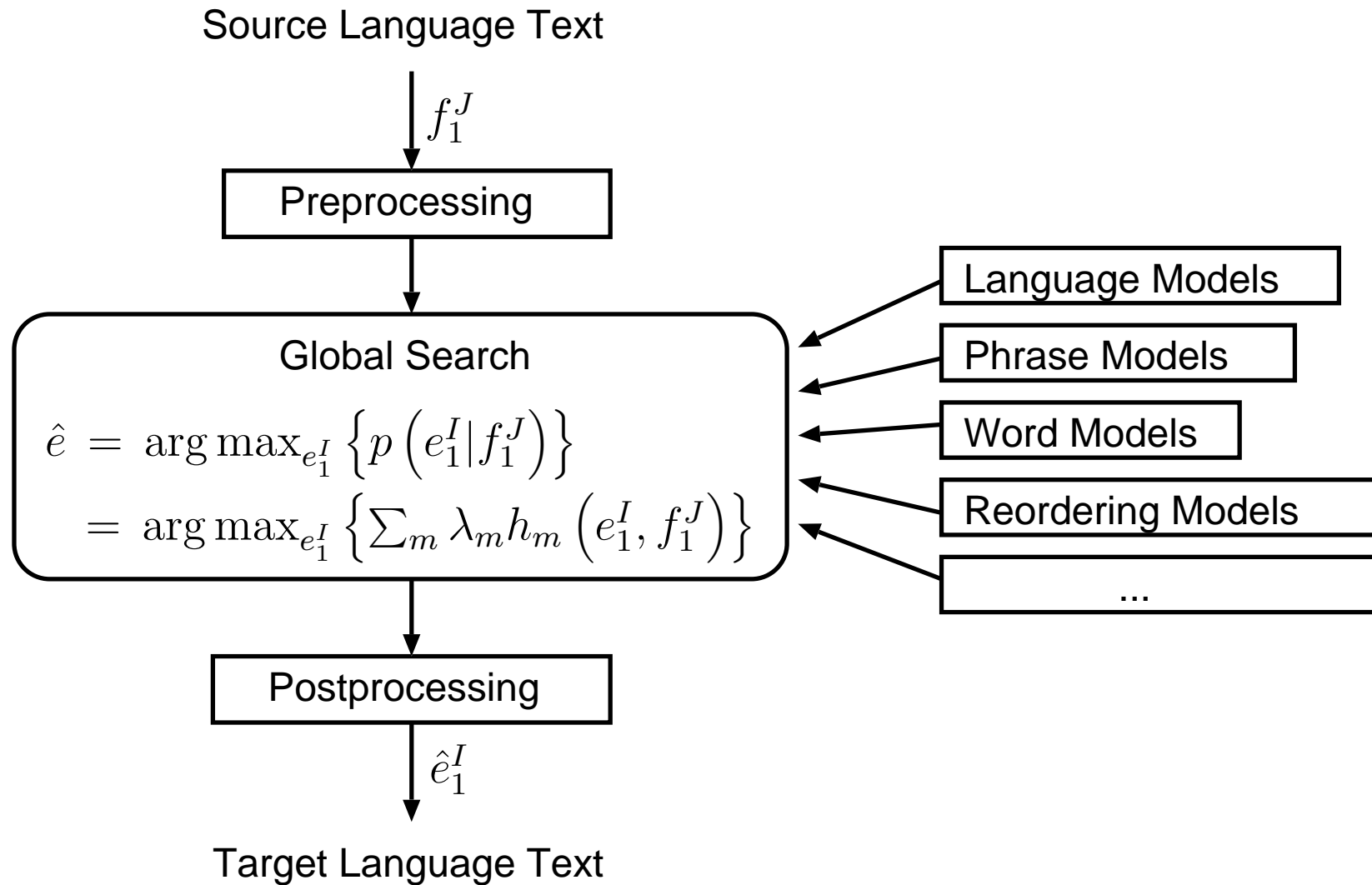
$$= \arg \max_{e_1^I} \left\{ \sum_{m=1}^M \lambda_m h_m(e_1^I, f_1^J) \right\} \quad (3)$$

für Skalierungsfaktoren  $\lambda_m$  und Funktionen  $h_m(e_1^I, f_1^J)$ ,  $m = 1, \dots, M$ .  
**Algorithmen zur Optimierung der Skalierungsfaktoren: nächstes Blatt**

# Einfache Zusatzmodelle

- ▶ Anzahl der bei der Übersetzung benutzten Phrasen (Phrase Penalty)
- ▶ Anzahl der Wörter auf Ziel-Seite (Word Penalty)
- ▶ Count-Heuristiken (z.B. Phrasenvorkommen  $>1$ ,  $>2$ ,  $>5$ )
- ▶ Source-Target Ratio
- ▶ Signifikanztests
- ▶ ...





## 6 Übung 4

- ▶ **Extraktion einer Phrasen-Übersetzungstabelle mit relativen Häufigkeiten, Implementierung eines (einfachen) phrasenbasierten Übersetzers**
  - ▷ **Präfixbaum zur Speicherung der Phrasentabelle**
  - ▷ **phrasenbasiertes Decoding**
  - ▷ **monoton von links nach rechts, keine Umordnungen in der Phrasenabfolge**
  - ▷ **log-lineare Kombination mehrerer Einzelmodelle**
  - ▷ **simples Unigramm-Sprachmodell**
- ▶ **Eingabe: unbekannte Sätze in Französisch**
- ▶ **Ausgabe: Übersetzung in Englisch**

# Fragen?

## Viel Erfolg!

