

Softwareprojektpraktikum Maschinelle Übersetzung

Matthias Huck, Daniel Stein
{huck,stein}@i6.informatik.rwth-aachen.de

Besprechung 2. Aufgabe 06. Mai 2010

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

Was ist statistische maschinelle Übersetzung?

„It must be recognized that the notion of a *probability of a sentence* is an entirely useless one, under any interpretation of this term.“

– Noam Chomsky, 1969

Was ist statistische maschinelle Übersetzung?

► Gegeben:

- ▷ parallele Sätze von vorübersetztem Trainingsmaterial
- ▷ Beispiel: politische Reden im Europäischen Parlament, Bücher, die bereits in verschiedene Sprachen übersetzt wurden (z.B. die Bibel, Handbücher, Patente),
...

► Typische Größen (in Anzahl der parallelen Sätze):

- ▷ klein: 40 K (IWSLT)
- ▷ mittel: 3 M (Europarl)
- ▷ groß: 10 M (GALE)



► Ziel:

- ▷ Die beste (d.h.: wahrscheinlichste) Übersetzung eines unbekannten Satzes zu finden

Ansatz

- ▶ Idee: die Übersetzung als Entschlüsselungsproblem zu sehen
- ▶ Für einen gegebenen Quellsatz $f_1^J = f_1 \dots f_j \dots f_J$
 - ▷ berechne $Pr(e_1^I | f_1^J)$ für alle möglichen Zielsätze $e_1^I = e_1 \dots e_i \dots e_I$
 - ▷ wähle den Satz \hat{e}_1^I , der die Wahrscheinlichkeit maximiert

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



Bestandteile der Maschinellen Übersetzung

► Fünf integrale Bestandteile:

- ▷ **Training:** Wort-Alignment für $f_j \# e_i$ Wortpaare
- ▷ **Extraktion:** Suche nach Fragmenten in einem bilingualen Trainings-Corpus
- ▷ **Log-linear model:** Kombination möglicher Abhängigkeiten zwischen f_1^J und e_1^I
- ▷ **Suche:** Finden des wahrscheinlichsten, „plausibelsten“ Zielsatzes e_1^I
- ▷ **Optimierung:** automatische Bewertung der Ausgabe und Optimierung

► Ähnlichkeiten zur Spracherkennung, aber

- ▷ Nicht-monotone Suche
- ▷ Ausgabe schwerer zu interpretieren (z.B. bei Synonymen, anderer Grammatikstruktur, ...)

Wort-Alignment

- ▶ **Eingeführt von IBM 1989–1993**
 - ▷ Sequenz von IBM-1, ..., IBM-5 Modellen
 - ▷ Berechnung durch den EM-Algorithmus
- ▶ **Erweiterungen der RWTH**
 - ▷ Zusätzliches Modell (HMM), statt IBM-2
 - ▷ Effiziente Implementierung
 - ▷ Open Source Toolkit: GIZA++

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

Annahmen der Zero-Order Alignment-Modelle

$$\begin{aligned} Pr(f_1^J | e_1^I) &= p(J|I) \cdot Pr(f_1^J | J, e_1^I) \\ &= p(J|I) \cdot \sum_{a_1^J} Pr(f_1^J, a_1^J | J, e_1^I) \\ &= p(J|I) \cdot \sum_{a_1^J} \prod_{j=1}^J p(a_j | j, J, I) \cdot p(f_j | e_{a_j}) \end{aligned}$$

Annahmen der Zero-Order Alignment-Modelle

$$Pr(f_1^J | e_1^I) = p(J|I) \cdot \sum_{a_1^J} \prod_{j=1}^J p(a_j | j, J, I) \cdot p(f_j | e_{a_j})$$

► **Längenmodell:**

Abhängigkeit der Länge J nur vom Zielsatz e_1^I :

$$Pr(J | e_1^I) = p(J|I)$$

► **Alignment Model:**

Abhängigkeit nur an der absoluten Position j (und den Längen J und I):

$$Pr(a_j | a_1^{j-1}, J, e_1^I) = p(a_j | j, J, I)$$

► **Lexikon Wahrscheinlichkeit:**

Abhängigkeit nur von e_i in der Position $i = a_j$:

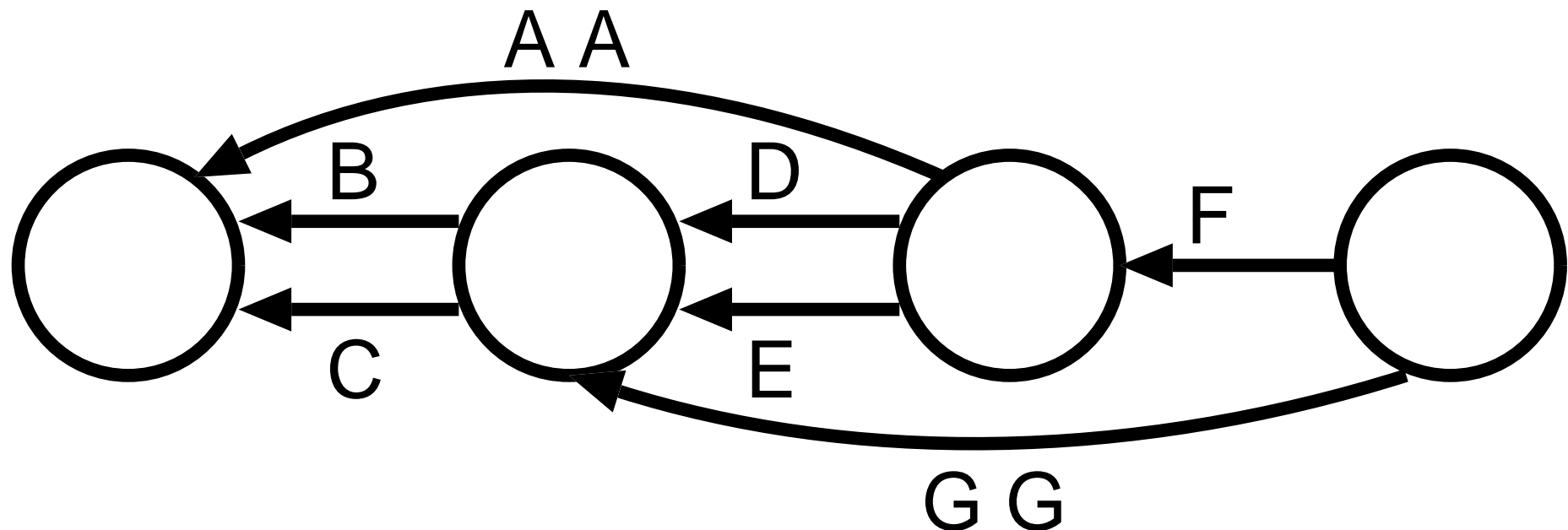
$$Pr(f_j | f_1^{j-1}, a_1^J, J, e_1^I) = p(f_j | e_{a_j})$$

Übung 2

- ▶ **Ziel: Konstruktion eines (einfachen) Übersetzers für Italienisch-Englisch**
- ▶ **Eingabe: unbekannte Sätze in Italienisch**
- ▶ **Ausgabe: Übersetzung in Englisch**

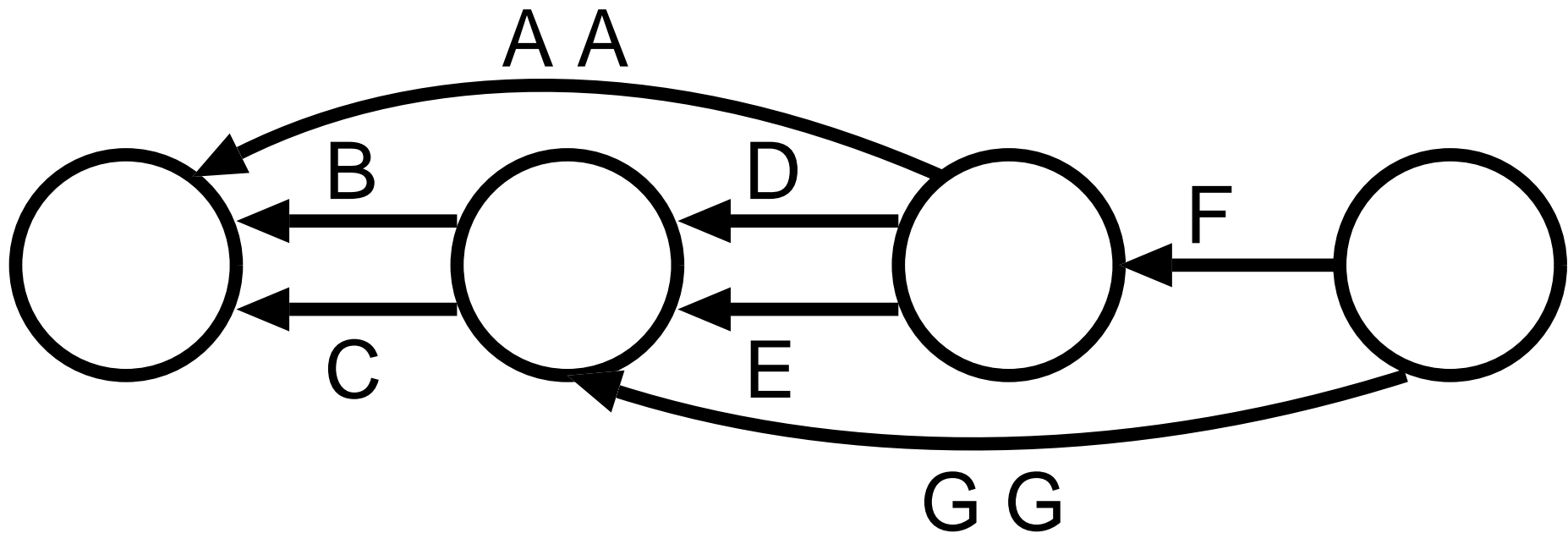
Berechnung der minimalen Kosten

- ▶ **Einfach:** Berechnung des Satzes mit den minimalen Kosten
- ▶ **Semi-Einfach:** Berechnung der nächsten Alternative für einfache Übergänge
- ▶ **Problematisch:** Berechnung der nächsten Alternative für beliebig lange Übergänge



Berechnung der minimalen Kosten

- ▶ Einfach: (langweilig)
- ▶ Semi-Einfach: (langweilig)
- ▶ Problematisch: Eure Aufgabe



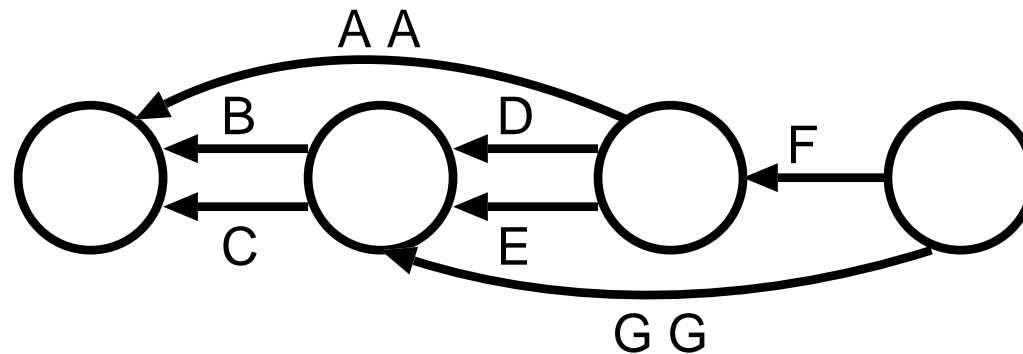
A*-Suche

- ▶ **single-best Berechnung einfach: jeweils besten Pfad abspeichern**
- ▶ **n-best Berechnung durch den A*-Suchalgorithmus**
 - ▷ **informierter Suchalgorithmus**
 - ▷ **untersucht Knoten zuerst, die am vielversprechendsten sind**
 - ▷ **benötigt *optimistische* Schätzungsfunktion $f(x)$**
- ▶ **in unserer Übersetzung: $f(x) = g(x) + h(x)$, mit**
 - ▷ **$g(x)$ sind die Übersetzungskosten**
 - ▷ **$h(x)$ sind die besten Pfade zum Knoten**

N-Best Berechnung durch A*-Suche

Tabelle:

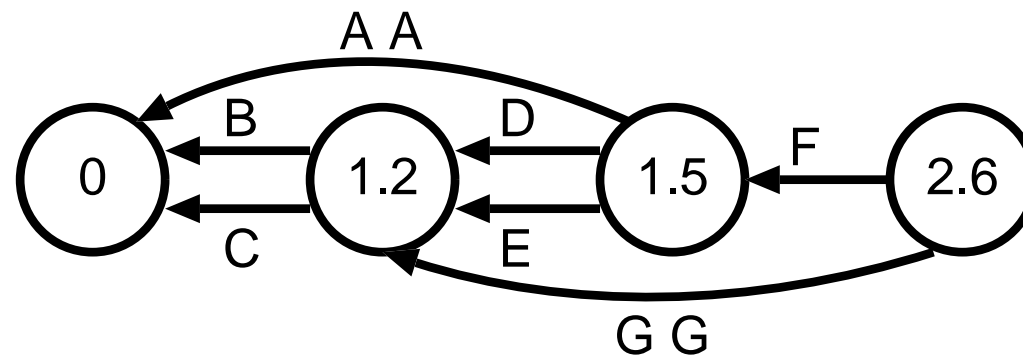
1.5	#	X	Y	#	A	A
1.4	#	X		#	B	
1.2	#	X		#	C	
1.2	#	Y		#	D	
1.3	#	Y		#	E	
2.5	#	Y	Z	#	G	G
1.1	#	Z		#	F	



N-Best Berechnung durch A*-Suche

Tabelle:

1.5	#	X Y	#	A A
1.4	#	X	#	B
1.2	#	X	#	C
1.2	#	Y	#	D
1.3	#	Y	#	E
2.5	#	Y Z	#	G G
1.1	#	Z	#	F



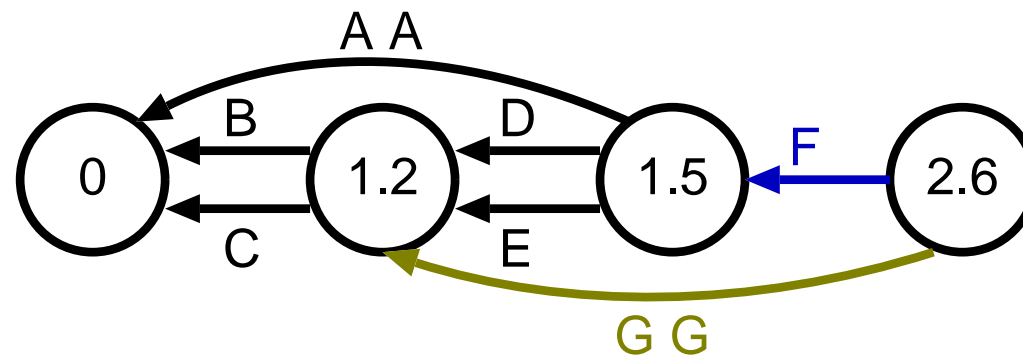
N-Best Berechnung durch A*-Suche

Tabelle:

1.5	#	X Y	#	A A
1.4	#	X	#	B
1.2	#	X	#	C
1.2	#	Y	#	D
1.3	#	Y	#	E
2.5	#	Y Z	#	G G
1.1	#	Z	#	F

hyp	f(x)	g(x)	h(x)
F	2.6	1.1	1.5
G G	2.7	2.5	1.2

⇒ expandiere F



N-Best Berechnung durch A*-Suche

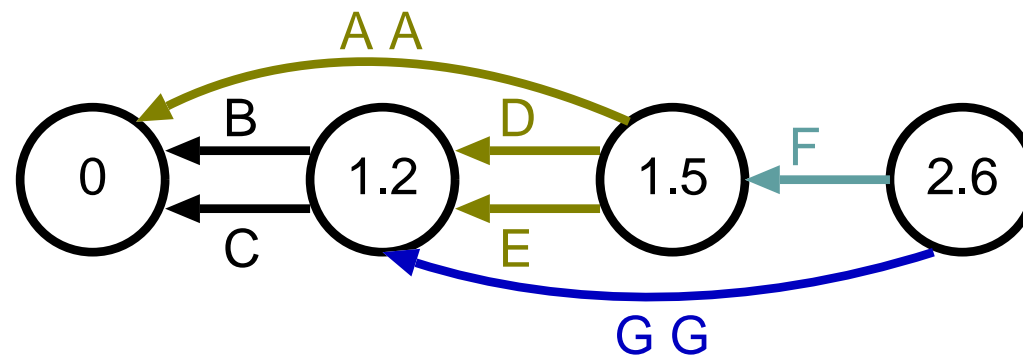
Tabelle:

1.5	#	X Y	#	A A
1.4	#	X	#	B
1.2	#	X	#	C
1.2	#	Y	#	D
1.3	#	Y	#	E
2.5	#	Y Z	#	G G
1.1	#	Z	#	F

hyp	f(x)	g(x)	h(x)
A A F	2.6	2.6	0
D F	3.5	2.3	1.2
E F	3.6	2.4	1.2
G G	2.7	2.5	1.2

⇒ gib A A F aus

⇒ expandiere G G



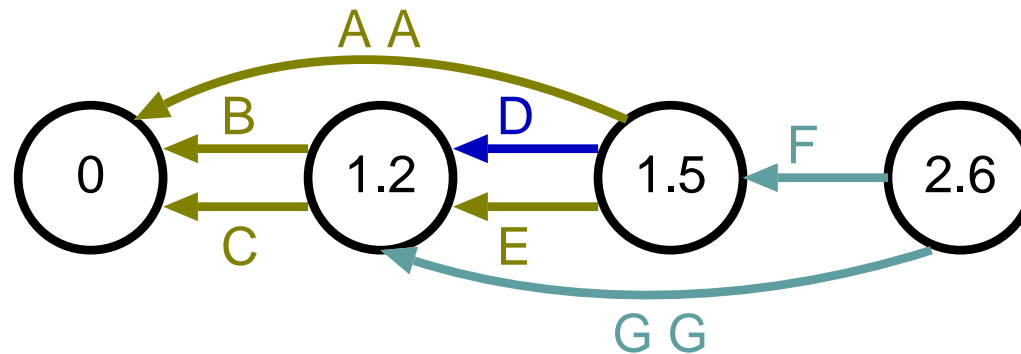
N-Best Berechnung durch A*-Suche

Tabelle:

1.5	#	X Y	#	A A
1.4	#	X	#	B
1.2	#	X	#	C
1.2	#	Y	#	D
1.3	#	Y	#	E
2.5	#	Y Z	#	G G
1.1	#	Z	#	F

hyp	f(x)	g(x)	h(x)
D F	3.5	2.3	1.2
E F	3.6	2.4	1.2
B G G	2.9	2.9	0
C G G	2.7	2.7	0

- ⇒ gib C G G aus
- ⇒ gib B G G aus
- ⇒ expandiere D F



Fragen?

Ihr wisst wo unser Büro ist :-)

