
Approximatives Textalignment

Sven Schrinner

14. Mai 2014

Inhalt

Inhalt:

- ① Global Alignment
 - Spezialfall des Needleman-Wunsch-Algorithmus
- ② Semiglobales Alignment

Inhalt

Inhalt:

- ① Global Alignment
 - Spezialfall des Needleman-Wunsch-Algorithmus
- ② Semiglobales Alignment

Verwendete Quellen:

- Skript zu „Algorithmen auf Sequenzen (2013)“
- (((Wikipedia-Artikel zu „Needleman-Wunsch-Algorithmus“)))

Globales Alignment

Gegeben Zwei Strings $s, t \in \Sigma^*$

Gesucht „Optimales“ Alignment zwischen s und t

Globales Alignment

Gegeben Zwei Strings $s, t \in \Sigma^*$

Gesucht „Optimales“ Alignment zwischen s und t

Was ist ein Alignment?

Globales Alignment

Gegeben Zwei Strings $s, t \in \Sigma^*$

Gesucht „Optimales“ Alignment zwischen s und t

Was ist ein Alignment?

Definition (nach Def 4.8, Skript A.a.S.) Ein Alignment A von $s, t \in \Sigma^*$ ist ein String über $(\Sigma \cup \{-\})^2$ (ohne $(-, -)$), sodass die erste Komponente ohne „-“ den String s ergibt und die zweite Komponente entsprechend t .

Globales Alignment

Gegeben Zwei Strings $s, t \in \Sigma^*$

Gesucht „Optimales“ Alignment zwischen s und t

Was ist ein Alignment?

Definition (nach Def 4.8, Skript A.a.S.) Ein Alignment A von $s, t \in \Sigma^*$ ist ein String über $(\Sigma \cup \{-\})^2$ (ohne $(-, -)$), sodass die erste Komponente ohne „-“ den String s ergibt und die zweite Komponente entsprechend t .

Beispiel $s = \text{RABABA}$, $t = \text{BARBARA}$

R	A	B	A	-	B	A	-	-
-	-	B	A	R	B	A	R	A

Globales Alignment

Was ist ein optimales Alignment?

Globales Alignment

Was ist ein optimales Alignment?

Jede Spalte (a, b) eines Alignment besitzt Kosten $d(a, b)$, die wir über das gesamte Alignment minimieren wollen.

Globales Alignment

Was ist ein optimales Alignment?

Jede Spalte (a, b) eines Alignment besitzt Kosten $d(a, b)$, die wir über das gesamte Alignment minimieren wollen.

Beispiel: Edit-Distanz / Levenshtein-Distanz

R A B A - B A - -

- - B A R B A R A (Distanz 5)

$$d(a, b) = \begin{cases} 0, & \text{falls } a = b \\ 1, & \text{falls } a \neq b \end{cases}$$

Globales Alignment

Was ist ein optimales Alignment?

Jede Spalte (a, b) eines Alignment besitzt Kosten $d(a, b)$, die wir über das gesamte Alignment minimieren wollen.

Beispiel: Edit-Distanz / Levenshtein-Distanz

R A B A - B A - -
 - - B A R B A R A (Distanz 5)

$$d(a, b) = \begin{cases} 0, & \text{falls } a = b \\ 1, & \text{falls } a \neq b \end{cases}$$

Optimales Alignment:

R A - B A B A
 B A R B A R A (Distanz 3)

Globales Alignment mit dynamischer Programmierung

Definiere Matrix D mit $|s| * |t|$ Einträgen:

$$D[i, j] := \text{Distanz zwischen } s[0, i - 1] \text{ und } t[0, j - 1]$$

Globales Alignment mit dynamischer Programmierung

Definiere Matrix D mit $|s| * |t|$ Einträgen:

$$D[i, j] := \text{Distanz zwischen } s[0, i - 1] \text{ und } t[0, j - 1]$$

Initialisierung:

$$D[i, 0] = i$$

$$D[0, j] = j$$

Globales Alignment mit dynamischer Programmierung

Definiere Matrix D mit $|s| * |t|$ Einträgen:

$$D[i, j] := \text{Distanz zwischen } s[0, i - 1] \text{ und } t[0, j - 1]$$

Initialisierung:

$$D[i, 0] = i$$

$$D[0, j] = j$$

Rekursion:

$$D[i, j] = \min \begin{cases} D[i - 1, j - 1] & +d(s[i - 1], t[j - 1]) \\ D[i - 1, j] & +1 \\ D[i, j - 1] & +1 \end{cases}$$

Globales Alignment (Beispiel)

	C	A	G	A	C
A					
G					
A					
T					

Globales Alignment (Beispiel)

		C	A	G	A	C
	0	1	2	3	4	5
A	1					
G	2					
A	3					
T	4					

Globales Alignment (Beispiel)

		C	A	G	A	C
	0	1	2	3	4	5
A	1	1				
G	2					
A	3					
T	4					

Globales Alignment (Beispiel)

		C	A	G	A	C
	0	1	2	3	4	5
A	1	1				
G	2	2				
A	3					
T	4					

Globales Alignment (Beispiel)

		C	A	G	A	C
	0	1	2	3	4	5
A	1	1				
G	2	2				
A	3	3				
T	4					

Globales Alignment (Beispiel)

		C	A	G	A	C
	0	1	2	3	4	5
A	1	1				
G	2	2				
A	3	3				
T	4	4				

Globales Alignment (Beispiel)

		C	A	G	A	C
	0	1	2	3	4	5
A	1	1	1			
G	2	2				
A	3	3				
T	4	4				

Globales Alignment (Beispiel)

		C	A	G	A	C
	0	1	2	3	4	5
A	1	1	1			
G	2	2	2			
A	3	3				
T	4	4				

Globales Alignment (Beispiel)

		C	A	G	A	C
	0	1	2	3	4	5
A	1	1	1			
G	2	2	2			
A	3	3	2			
T	4	4				

Globales Alignment (Beispiel)

		C	A	G	A	C
	0	1	2	3	4	5
A	1	1	1			
G	2	2	2			
A	3	3	2			
T	4	4	3			

Globales Alignment (Beispiel)

		C	A	G	A	C
	0	1	2	3	4	5
A	1	1	1	2		
G	2	2	2			
A	3	3	2			
T	4	4	3			

Globales Alignment (Beispiel)

		C	A	G	A	C
	0	1	2	3	4	5
A	1	1	1	2		
G	2	2	2	1		
A	3	3	2			
T	4	4	3			

Globales Alignment (Beispiel)

		C	A	G	A	C
	0	1	2	3	4	5
A	1	1	1	2		
G	2	2	2	1		
A	3	3	2	2		
T	4	4	3			

Globales Alignment (Beispiel)

		C	A	G	A	C
	0	1	2	3	4	5
A	1	1	1	2		
G	2	2	2	1		
A	3	3	2	2		
T	4	4	3	3		

Globales Alignment (Beispiel)

		C	A	G	A	C
	0	1	2	3	4	5
A	1	1	1	2	3	
G	2	2	2	1	2	
A	3	3	2	2	1	
T	4	4	3	3	2	

Globales Alignment (Beispiel)

		C	A	G	A	C
	0	1	2	3	4	5
A	1	1	1	2	3	4
G	2	2	2	1	2	3
A	3	3	2	2	1	2
T	4	4	3	3	2	2

Globales Alignment (Analyse)

Laufzeit und Speicherbedarf:

- Konstante Laufzeit pro Eintrag $\rightarrow \mathcal{O}(mn)$
- Spaltenweise Berechnung: $\mathcal{O}(m + n)$ Speicherbedarf

Globales Alignment (Analyse)

Laufzeit und Speicherbedarf:

- Konstante Laufzeit pro Eintrag $\rightarrow \mathcal{O}(mn)$
- Spaltenweise Berechnung: $\mathcal{O}(m + n)$ Speicherbedarf

Berechnung des tatsächlichen Alignments:

- Zusatzmatrix, um gewählten Rekursionszweig zu speichern
- Speicherbedarf steigt auf $\mathcal{O}(mn)$

Semiglobales Alignment

Gegeben Text T und Muster P

Gesucht „Optimales“ Alignment zwischen P und einem Substring von T

Semiglobales Alignment

Gegeben Text T und Muster P

Gesucht „Optimales“ Alignment zwischen P und einem Substring von T

Idee: Passe Algorithmus für globales Alignment an:

Semiglobales Alignment

Gegeben Text T und Muster P

Gesucht „Optimales“ Alignment zwischen P und einem Substring von T

Idee: Passe Algorithmus für globales Alignment an:

- Erlaube eine kostenlose, beliebig lange Deletion zu Beginn des Textes
- Initialisiere erste Zeile mit Nullen

Semiglobales Alignment

Gegeben Text T und Muster P

Gesucht „Optimales“ Alignment zwischen P und einem Substring von T

Idee: Passe Algorithmus für globales Alignment an:

- Erlaube eine kostenlose, beliebig lange Deletion zu Beginn des Textes
- Initialisiere erste Zeile mit Nullen

→ Komplexität bleibt gleich

Semiglobales Alignment

Zusätzliche Optimierung:

- Bei vorgegebener Fehlerschranke nicht komplette Tabelle berechnen
- Senkt die erwartete Laufzeit auf etwa $\mathcal{O}(nk)$

		A	M	O	A	M	A	M	A	O	M
M	0	●	0	0	0	0	●	0	0	0	0
A	1	1	●	1	1	0	1	●	1	1	0
O	2	1	●	1	1	1	0	1	●	1	1
A	3	2	2	●	2	2	1	1	1	●	1
M	4	3	3	2	●	2	2	2	1	●	1
	5	4	3	3	2	●	2	2	2	2	●

Allgemeine Distanzmaße

Needleman-Wunsch-Algorithmus:

- Verallgemeinerung des DP-Algorithmus auf allgemeinere Distanzmaße möglich
- Unterschiedliche Gewichtung von Mismatches und Gaps
- Positionsabhängige Gewichtung für Gaps möglich

Allgemeine Distanzmaße

Needleman-Wunsch-Algorithmus:

- Verallgemeinerung des DP-Algorithmus auf allgemeinere Distanzmaße möglich
- Unterschiedliche Gewichtung von Mismatches und Gaps
- Positionsabhängige Gewichtung für Gaps möglich

Konsequenzen:

- Rekursionsgleichung muss angepasst werden (bei positionsabhängigen Gaps muss ganze Spalte/Zeile pro Zelle berücksichtigt werden)
- Bei letzterem Fall: Verschlechterung der Laufzeit auf $\mathcal{O}(mn(m+n))$

→ Edit-Distanz für unsere Zwecke wahrscheinlich ausreichend!