Fortgeschrittene Themen der statistische maschinelle Übersetzung

(Advanced SMT)

Zur Wortalinierung

Miriam Kaeshammer
Heinrich-Heine-Universität Düsseldorf

Wintersemester 2014/15

Wortalinierung - Lexikalische Übersetzungsmodelle

In unserem einfachsten Übersetzungsmodell entspricht jedes englische Wort e_j genau einem deutschen Wort $f_{a(j)}$ (oder NULL). a ist die Alinierungsfunktion, $t(e|f)$ die Verteilung der Wort-zu-Wort-Übersetzungswahrscheinlichkeit.

Wir haben dann

$$P(\vec{e}|a,\vec{f}) = \prod_{j=1}^{|\vec{e}|} t(e_j|f_{a(j)})$$

Außerdem

$$P(\vec{e},a|\vec{f}) = P(a|\vec{f}) \cdot P(\vec{e}|a,\vec{f})$$

wobei $P(a|\vec{f})$ gleichverteilt ist und nur von der Länge von $\vec{f}$ (und $\vec{e}$) abhängt, nicht aber von $\vec{e}$ und $\vec{f}$ selbst.

Schätzung von Übersetzungswahrscheinlichkeiten (1)

Woher bekommen wir $t(e|f)$?

Wenn wir parallele Korpora hätten, die auf Wortebene aliniert sind, könnten wir die Übersetzungswahrscheinlichkeiten mit der Maximum-Likelihood-Methode schätzen:

$$t(e|f) = \frac{C(e|f)}{C(f)}$$

Schätzung von Übersetzungswahrscheinlichkeiten (2)

Wenn wir für jeden Satz nicht *die eine richtige* Alinierung gegeben hätten, sondern mehrere mögliche, und jede hätte eine Wahrscheinlichkeit ($P(a|\vec{e},\vec{f})$), könnten wir auch zählen, wie oft $e|f$ vorkommt, und jedes Vorkommen mit $P(a|\vec{e},\vec{f})$ gewichten.

$\Rightarrow$ anteilige Häufigkeiten (*fractional counts*)

$$C(e|f;\vec{e},\vec{f}) = \sum_{a \in A(|\vec{e}|,|\vec{f}|)} \left[P(a|\vec{e},\vec{f}) \sum_{j=1}^{|\vec{e}|} \delta(e,e_j)\delta(f,f_{a(j)}) \right]$$

Alinierungswahrscheinlichkeiten

Wenn wir Übersetzungwahrscheinlichkeiten $t(e|f)$ hätten, könnten wir Alinierungswahrscheinlichkeiten ausrechnen:

$$P(a|\vec{e},\vec{f}) = \frac{P(\vec{e},a|\vec{f})}{P(\vec{e}|\vec{f})} = \frac{P(\vec{e},a|\vec{f})}{\sum_{a' \in A(|\vec{e}|,|\vec{f}|)} P(\vec{e},a'|\vec{f})}$$

. . . wir drehen uns im Kreis . . .

Lösung: EM-Algorithmus

EM-Algorithmus für lexikalische Übersetzungswahrsch.

Parameter, die wir suchen: $t(e|f)$

Versteckte Variable: Wortalinierung $\Rightarrow P(a|\vec{e},\vec{f})$

1. Initialisiere die Parameterwerte (uniform).
2. Wende das Modell auf die Daten an $\rightarrow$ Berechne $P(a|\vec{e},\vec{f})$.
   Für jedes Satzpaar:
   a. Berechne $P(\vec{e},a|\vec{f})$ für jede mögliche Alinierung a.
   b. Normalisiere jedes $P(\vec{e},a|\vec{f})$ um $P(a|\vec{e},\vec{f})$ zu erhalten.
3. Lerne ein neues Modell (von den aktuellen $P(a|\vec{e},\vec{f})$).
   Für jedes Wortpaar (e,f):
   a. Zähle anteilige Häufigkeiten $C(e|f)$
   b. Normalisiere die anteiligen Häufigkeiten, um neue Parameterwerte zu erhalten: $t(e|f) = \frac{C(e|f)}{\sum_{e' \in V_E} C(e'|f)}$

Wiederhole Schritte 2. und 3.

EM für IBM-Modell 1

Problem bei der Anwendung von EM (wie gerade präsentiert):
Man muss in jeder Runde für jedes Satzpaar alle möglichen Alinierungen „ausbuchstabieren“.

Für IBM-Modell 1 gibt es einen Ausweg: Die anteiligen Häufigkeiten (Schritt 3a.) können direkt aus den Übersetzungswahrscheinlichkeiten $t(e|f)$ der vorherigen Runde bestimmt werden. Der Umweg über die Alinierungswahrscheinlichkeiten (Schritt 2) ist nicht mehr nötig.

Herleitung siehe Tafel bzw. Folien zum SMT-Buch

$$C(e|f;\vec{e},\vec{f}) = \frac{t(e|f)}{\sum_{i=0}^{|\vec{f}|} t(e|f_i)} \sum_{j=1}^{|\vec{e}|} \delta(e,e_j) \sum_{i=0}^{|\vec{f}|} \delta(f,f_i)$$

$$t(e|f) = \frac{\sum_{(\vec{e},\vec{f})\in\mathfrak{K}} C(e|f;\vec{e},\vec{f})}{\sum_{e'\in V_E} \sum_{(\vec{e},\vec{f})\in\mathfrak{K}} C(e'|f,\vec{e},\vec{f})}$$

EM für IBM-Modell 1: Pseudocode

$k = 0$, initialize $t_0(\bullet|\bullet)$
repeat
  $k = k + 1$
  Initialize all counts C to 0
  for all $(\vec{e}, \vec{f}) \in \mathfrak{K}$ **do**
    for all $j \in 1 \ldots |\vec{e}|$ **do**
      $Z = 0$
      for all $i \in 0 \ldots |\vec{f}|$ **do**
        $Z+ = t_{k-1}(e_j|d_i)$
      for all $i \in 0 \ldots |\vec{f}|$ **do**
        $c = t_{k-1}(e_j|f_i)/Z$
        $C(e_j|f_i)+ = c$
        $C(f_i)+ = c$
  for all $(e, f) \in \{V_E \times V_F\}$ **do**
    $t_k(e|f) = C(e|f)/C(f)$
until some criterion is met (e.g. fixed number of iterations)

EM für IBM-Modell 1: Beispiel (1)

Korpus:

1. (Hund bellte, dog barked)
2. (Hund, dog)

1. Initialisiere uniform

$t_0(e\|f)$	dog	barked
Hund	$\frac{1}{2}$	$\frac{1}{2}$
bellte	$\frac{1}{2}$	$\frac{1}{2}$
NULL	$\frac{1}{2}$	$\frac{1}{2}$

2a. Anteilige Häufigkeiten auf Satzebene

In Bezug auf Folie 7: Die schwarzen Brüche sind $t_0(e|f)$. Die Summe $\sum_{i=0}^{|\vec{f}|} P(e|f_i)$ wird in der letzten Zeile gebildet. In rot sehen Sie dann die eigentlichen anteiligen Häufigkeiten bzw. wie sie berechnet wurden.

$C_1(e\|f,\vec{e}_1,\vec{f}_1)$	dog	barked
Hund	$\frac{1}{2}\cdot\frac{2}{3}=\frac{1}{3}$	$\frac{1}{2}\cdot\frac{2}{3}=\frac{1}{3}$
bellte	$\frac{1}{2}\cdot\frac{2}{3}=\frac{1}{3}$	$\frac{1}{2}\cdot\frac{2}{3}=\frac{1}{3}$
NULL	$\frac{1}{2}\cdot\frac{2}{3}=\frac{1}{3}$	$\frac{1}{2}\cdot\frac{2}{3}=\frac{1}{3}$
$\sum \quad (\rightarrow Z)$	$\frac{3}{2}$	$\frac{3}{2}$

$C_1(e\|f,\vec{e}_2,\vec{f}_2)$	dog
Hund	$\frac{1}{2}\cdot 1=\frac{1}{2}$
NULL	$\frac{1}{2}\cdot 1=\frac{1}{2}$
$\sum \quad (\rightarrow Z)$	1

EM für IBM-Modell 1: Beispiel (2)

2b. Anteilige Häufigkeiten auf Korpusebene

$C(e\|f)$	dog	barked	$\sum$ $(\rightarrow C(f))$
Hund	$\frac{1}{3} + \frac{1}{2} = \frac{5}{6}$	$\frac{1}{3}$	$\frac{7}{6}$
bellte	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{2}{3}$
NULL	$\frac{1}{3} + \frac{1}{2} = \frac{5}{6}$	$\frac{1}{3}$	$\frac{7}{6}$

3. Neue Parameter

$t_1(e\|f)$	dog	barked
Hund	$\frac{5}{6} \cdot \frac{6}{7} = \frac{5}{7}$	$\frac{1}{3} \cdot \frac{6}{7} = \frac{2}{7}$
bellte	$\frac{1}{2}$	$\frac{1}{2}$
NULL	$\frac{5}{7}$	$\frac{2}{7}$

Iteriere 2a + 2b + 3

Bemerkung: Sowohl die Häufigkeiten als auch die Parameter $t_k(e|f)$ stimmen mit denen des allgemeinen Algorithmus (Folie 6) überein!

Die wahrscheinlichste Alinierung

Nachdem wir IBM-Modell 1 trainiert haben, wie alinieren wir die Satzpaare $(\vec{e}, \vec{f})$?

$\Rightarrow$ Finde die wahrscheinlichste Alinierung (Viterbi-Alinierung):

$$\hat{a} = \underset{a \in A(|\vec{e}|,|\vec{f}|)}{\operatorname{argmax}} P(a|\vec{e}, \vec{f}) = \underset{a \in A(|\vec{e}|,|\vec{f}|)}{\operatorname{argmax}} P(\vec{e}, a|\vec{f})$$

Für IBM-Model 1: $P(\vec{e}, a|\vec{f}) = \frac{1}{(|\vec{f}|+1)^{|\vec{e}|}} \prod_{j=1}^{|\vec{e}|} t(e_j|f_{a(j)})$

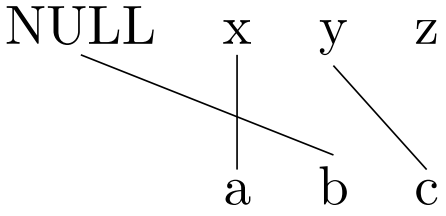
$$\hat{a}_j = \underset{i \in 0 \ldots |\vec{f}|}{\operatorname{argmax}} t(e_j|f_i)$$

Repräsentation von Alinierung (1)

Beispiel

ein Satzpaar `de:` x y z `en:` a b c

beste Alinierung, die das EM-IBM1-Training ergibt:

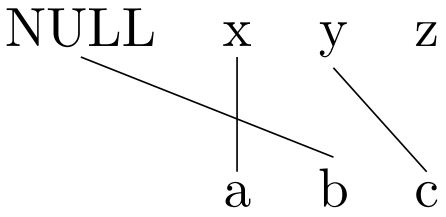


Repräsentation	
Alinierungsfunktion	$a(1) = 1, a(2) = 0, a(3) = 2$
(Integer-) Array	`[1,0,2]`
`i-j`-Format (oft 0-basiert)	`0-0 1-2`

Repräsentation von Alinierung (2)

`i-j`-Format

`0-0 1-2`



- `i`: Index deutscher Satz $\vec{f}$, `i`: Index englischer Satz $\vec{e}$
- Indizierung fängt bei 0 an
- Null-Alignierungen werden nicht explizit dargestellt
- Vorteil: Darstellung von *many-to-many-alignments* möglich