Schriftliche Wiederholungsprüfung zur Übung
Statistische Methoden in der maschinellen Sprachverarbeitung
WS 2015/16
Dozent: Helmut Schmid

Aufgabe

Sie sollen ein **interpoliertes Backoff-Modell** für bedingte Wahrscheinlichkeiten von Wortart-Tags gegeben ein Wortsuffix berechnen. Ein Beispiel ist die Wahrscheinlichkeit p(Nomen|keiten)

Bei einem interpolierten Backoff-Modell ist die bedingte Wahrscheinlichkeit $p(t|a_1^n)$ des Wortart-Tags t gegeben das Suffix $a_1...a_n$ rekursiv definiert durch:

$$p(t|a_1^k) = \pi(t|a_1^k) + \alpha(a_1^k)\, p(t|a_2^k) \quad \text{für } 1 \leq k \leq n$$

Die "Pseudo-Wahrscheinlichkeit" $\pi(t|a_1^k)$ ist 0 falls $f(a_1^k, t) = 0$ (also falls nicht beobachtet) und sonst

$$\pi(t|a_1^k) = \frac{f(a_1^k, t) - \delta_k}{\sum_{t'} f(a_1^k, t')}$$

$f(a_1^k, t)$ ist die **Häufigkeit** des Tags t in Verbindung mit dem Wortsuffix $a_1...a_k$.
δ_k ist der **Discount-Faktor** für k-Gramm-Tag-Paare, der wie folgt berechnet wird

$$\delta_k = \frac{N_1^k}{N_1^k + 2N_2^k}$$

wobei N_i^k die Zahl der k-Gramm-Tag-Paare mit Häufigkeit i ist:

$$N_i^k = |\{(a_1^k, t) | f(a_1^k, t) = i\}|$$

Die letzte Backoff-Wahrscheinlichkeit ist wie folgt definiert:

$$p(t) = \frac{f(t)}{\sum_{t'} f(t')}$$

Häufigkeiten von $(k-1)$-Gramm-Tag-Paaren können durch Summation von k-Gramm-Tag-Paar-Häufigkeiten berechnet werden:

$$f(a_2^k, t) = \sum_a f(aa_2^k, t)$$

Zur Berechnung des Backoff-Faktors $\alpha(a_1^k)$ werden die Pseudo-Wahrscheinlichkeiten $\pi(t|a_1^k)$ für alle Tags summiert und von 1 subtrahiert:

$$\alpha(a_1^k) = 1 - \sum_t \pi(t|a_1^k)$$

Sie können bei Ihrem Programm annehmen, dass folgende Daten bereits zur Verfügung stehen:

n Länge der Buchstaben-NGramme, z.B. 5 bei einem 5gramm-Modell

freq Hashtabelle, die Paare von Buchstabenfolgen der Länge n und Tags auf ihre Häufigkeit (die größer 0 ist) abbildet. Sie können das genaue Format selbst definieren. (Es könnte bspw. eine Hashtabelle von Hashtabellen sein: $freq{'keiten'}{'Nomen'} = 3.

Am besten schreiben Sie eine Funktion **computeprob(freq, n)**, welcher Sie die Tabelle **freq** und den Längenparameter **n** übergeben, und die folgende Schritte ausführt:

Schritte:

- Berechnung der Häufigkeiten N_1 und N_2 sowie des Discounts δ für die Häufigkeiten $f(a_1^n, t)$ in der Tabelle **freq**
- Berechnung der Kontext-Häufigkeiten $f(a_1^n)$ durch Summation der Häufigkeiten $f(a_1^n, t)$.
- Berechnung der Pseudo-Wahrscheinlichkeiten $\pi(t|a_1^n) = \frac{f(a_1^n,t)-\delta}{f(a_1^n)}$ für alle N-Gramme in der Tabelle **freq** und Speicherung in einer (globalen) Hashtabelle **prob**
- Berechnung der Backoff-Faktoren $\alpha(a_1^n) = 1 - sum_t\, \pi(t|a_1^n)$ und Speicherung in einer (globalen) Hashtabelle **backoff**
- Berechnung der $n-1$-Gramm-Häufigkeiten $f(a_2^n, t) = \sum_{a_1} f(a_1^n, t)$ und Speicherung in **newfreq**
- rekursiver Aufruf **computeprob(newfreq, n-1)** falls $n > 1$
  sonst Berechnung der Wahrscheinlichkeiten $p(t) = \frac{f(t)}{\sum_{t'} f(t')}$

30 Punkte

Sie können sich 5 Zusatzpunkte verdienen, indem Sie eine Funktion **prob(suffix, tag)** schreiben, welche für ein gegebenes Suffix und Tag die interpolierte Wahrscheinlichkeit aus den Pseudo-Wahrscheinlichkeiten und den Backoff-Wahrscheinlichkeiten berechnet:

$$p(t|a_1^n) = \pi(t|a_1^n) + \alpha(a_1^n)\; p(t|a_2^n)$$

Sie können annehmen, dass **prob** und **backoff** als globale Variablen verfügbar sind.

Einen weiteren Zusatzpunkt bekommen Sie, wenn Sie angeben, wie Ihr Programm modifiziert werden muss, um das Kneser-Ney-Verfahren zu implementieren.