Übung 4: Sprachidentifizierer

Rechenübungen

Gegeben seien die folgenden Häufigkeiten von Buchstabenfolgen:

	a	b	$\langle/s\rangle$
a	14	4	2
b	6	12	2
$\langle s\rangle$	1	1	0

Die Häufigkeit von "ab" ist hier 4.

- Schätzen Sie die (ungeglätteten) Wahrscheinlichkeiten eines Markowmodelles 1. Ordnung.
- Welche Wahrscheinlichkeit hat die Buchstabenfolge *ababb*?

Implementierungsaufgaben

Aufgabe 1: Training eines mit der Backoff-Methode geglätteten Markowmodelles

Schreiben Sie ein Programm, welches folgende Aufgaben löst:

1. Das Programm erhält drei Argumente auf der Kommandozeile: die n-Gramm-Größe n, den Namen der Datei mit den Trainingsdaten und den Namen der Ausgabedatei.
2. Es extrahiert die Häufigkeiten aller Buchstaben-n-Gramme aus den Trainingsdaten.
3. Es berechnet, wieviele n-Gramme genau einmal bzw. zweimal aufgetreten sind.
4. Es berechnet den Discount für die Parameterglättung nach der Formel

$$\delta = \frac{N_1}{N_1 + 2N_2}$$

wobei N_k die Zahl der n-Gramme mit Häufigkeit k ist.

(Für $N_1 \leq N_2$ sollte ein fester Discount von 0.5 verwendet werden, weil hier der mit der Formel berechnete Wert unzuverlässig ist.)

5. Es berechnet die Kontext-Häufigkeiten durch Summation der n-Gramm-Häufigkeiten
$$f(a_1 a_2 ... a_{n-1}) = \sum_a f(a_1 a_2 ... a_{n-1} a)$$

6. Es berechnet für alle n-Gramme relative Häufigkeiten (mit Discount) nach der Formel
$$p^*(a_n|a_1, ..., a_{n-1}) = \frac{f(a_1, ..., a_n) - \delta}{f(a_1, ..., a_{n-1})}$$
und speichert sie in einem Dictionary.

7. Es berechnet n-1-Gramm-Häufigkeiten durch Summation der n-Gramm-Häufigkeiten
$$f(a_2 ... a_n) = \sum_a f(a a_2 ... a_n)$$

8. weiter mit Schritt 3, wobei nun relative Häufigkeiten für (n-1)-Gramme berechnet werden. Bei Unigrammen wird dann gestoppt.

9. Zum Schluss werden die berechneten Parameter mit pickle in einer Datei gespeichert.

Aufgabe 2: Laden Sie nun die Datei `www.cis.uni-muenchen.de/~schmid/lehre/StatNLP/data/Language-Samples.tgz` herunter und packen Sie das Archiv mit dem Befehl "tar zxf ..." aus.

Berechnen Sie für jede der Dateien mit dem Programm aus Aufgabe 1 die relativen N-Gramm-Häufigkeiten und speichern Sie sie jeweils in einer Datei.

Schreiben Sie nun ein weiteres Programm, welches Folgendes macht:

- Es liest die erzeugten Dateien ein, berechnet die Backoff-Faktoren nach der Formel
$$\alpha(a_1, ..., a_i) = 1 - \sum_a p^*(a|a_1, ..., a_i)$$
Die Parameter der verschiedenen Sprachmodelle werden separat gespeichert.

- Es liest eine weitere Datei ein, in welcher der zu klassifizierende Text steht.

- Es berechnet für jede Sprache L eine logarithmierte Wahrscheinlichkeit gemäß der Formel:
$$lp_L(a_1, ..., a_n) = \sum_{i=1}^{n} log\, p_L(a_i|a_{i-k}, ..., a_{i-1})$$
wobei $p_L(a_i|a_{i-k}, ..., a_{i-1})$ jeweils rekursiv wie folgt berechnet wird:
$$\begin{aligned} p_L(a_i|a_1^{i-1}) &= p_L^*(a_i|a_1^{i-1}) + \alpha(a_1^{i-1}) p_L(a_i|a_2^{i-1}) \text{ für } 1 < i \leq n \\ p(a_1) &= p_L^*(a_1) + \alpha() \frac{1}{1000} \end{aligned}$$

Die Unigramm-Wahrscheinlichkeiten werden also mit einer uniformen Verteilung geglättet, wobei von 1000 unterschiedlichen möglichen Zeichen ausgegangen wird.

Statt Grenzsymbolen fügen wir n-1 Leerzeichen am Anfang und ein Leerzeichen am Ende hinzu. Das funktioniert in der Praxis recht gut.

- Es gibt die Sprache (= Name der Parameterdatei) aus, bei welcher die Wahrscheinlichkeit maximal war.

Testen Sie das Programm mit verschiedenen Eingaben.

Ein Beispiel für die Berechnung der Backoff-Glättungen finden Sie auf den Vorlesungs-Folien 88 und 89.