Sprachidentifizierung

Vorgehen:

- Man sammelt für jede relevante Sprache ein **Textkorpus**.
- Man trainiert für jede Sprache ein Markowmodell, d.h. **schätzt** seine Parameter aus dem Korpus.
- Man berechnet die **Wahrscheinlichkeit des Textes** für jedes Sprachmodell.
  (Falls bekannt, multipliziert man noch die Apriori-Wk. p(L) der Sprache.)
- Man gibt die Sprache aus, bei der die **Wahrscheinlichkeit** am größten war.

Model parameters that have to be trained/estimated before deploying the model.

Input text:
"The second question concerns the reversed burden of proof. It is perfectly correct that we need to apply this in certain cases. "

Language identifier

$$\hat{L} = \arg\max_L p(L|T)$$

The model output the most probable language

Language candidates, e.g. en, de, fr

Apply several tranformations

Trick: Anwendung des Bayes'sche Theorems:

$$\arg\max_L p(L|T) = \arg\max_L \frac{p(T|L)p(L)}{p(T)}$$

Die Textwahrscheinlichkeit $p(T)$ ist eine Konstante, die keinen Einfluss auf das Ergebnis der arg-max-Operation hat und daher weggelassen werden kann:

$$\arg\max_L p(L|T) = \arg\max_L p(T|L)p(L)$$

Falls keine Information über die Apriori-Wahrscheinlichkeiten $p(L)$ der Sprachen verfügbar ist, können wir sie als gleichverteilt annehmen und ebenfalls ignorieren:

$$\arg\max_L p(L|T) = \arg\max_L p(T|L)$$

Angenommen der Text T besteht aus der Zeichenfolge $a_1, a_2, ..., a_n =: a_1^n$. Wir zerlegen $p(T|L)$ in ein Produkt von bedingten Wahrscheinlichkeiten:

$$\begin{aligned} p(T|L) &= p_L(a_1^n) = p_L(a_1, ..., a_n) \\ &= p_L(a_1)p_L(a_2|a_1)p_L(a_3|a_1, a_2)...p_L(a_n|a_1, ..., a_{n-1}) \\ &= \prod_{i=1}^{n} p_L(a_i|a_1, ..., a_{i-1}) \end{aligned}$$

$a_{n+1} = \langle s \rangle$:

$$p_L(a_1^n) = \prod_{i=1}^{n+1} p_L(a_i|a_1, ..., a_{i-1})$$

Add a word ending marker <s> to handle variable length texts

$$p_L(T) = \prod_{i=1}^{n+1} p_L(a_i|a_{i-k}...a_{i-1})$$

Apply **Markov-assumption** that the character a_i depends on only k previous characters (the sequence of context characters start at index i-k instead of 1)

If a Markov model has order k=2, it means the each conditional probability has 2 context characters. If we talk about n-gram markov model, n is equal k+1.
It means, a 3-gram markov model is a model with order k=2.
a 5-gram markov model will have k= 4.

Für $k = 2$ bekommen wir:
$p(\text{Er ölt}) =$
$p(E|\underline{\langle s\rangle,\langle s\rangle})\ p(r|\langle s\rangle,E)\ p(\_|E,r)\ p(ö|r,\_)\ p(l|\_,ö)\ p(t|ö,l)\ p(\underline{\langle s\rangle}|l,t)$

$\Rightarrow$ Es werden k Grenzsymbole $\langle s \rangle$ am Anfang und eines am Ende hinzugefügt:
$a_{-1} = a_0 = a_{n+1} = \langle s \rangle$

This is how we calculate the probability p of an input text "Er ölt" using a markov model with k=2 (3-gram-markov-model)

To define the prob of the first character, we use k special symbols <s> as its context.
* we only have **one** ending marker regardless of the number of k.

These are the parameters of the model. Now we want to find a way to estimate them.

Parameterschätzung

Die Parameter werden mit relativen Häufigkeiten aus Trainingsdaten geschätzt:

$$p(c|Spra) = \frac{f(Sprac)}{f(Spra)}$$

Falls das 5-Gramm *Sprac* nicht in den Trainingsdaten auftaucht, wird seine Wahrscheinlichkeit mit 0 geschätzt!

- ⇒ Die Wk. der gesamten Buchstabenfolge wird 0, egal wie gut die anderen Textteile modelliert werden.
- ⇒ Sparse-Data Problem
- ⇒ Notwendigkeit der Parameterglättung (später behandelt)

Beispiel: Wort n-Gramm-Wahrscheinlichkeiten

$$p(w_n|w_1...w_{n-1}) = \frac{f(w_1...w_n)}{f(w_1...w_{n-1})} = \frac{f(w_1...w_n)}{\sum_w f(w_1...w_{n-1}w)}$$

For example:
After we count the freq of each ngram from the corpus we have f(Sprac):10, f(Spras): 2, f(Sprao):11, f(Spral): 3, f(hause): 12, f(hunde):15, …

p(c| Spra) = f(Sprac) / f(Sprac)+f(Spras)+ f(Sprao)+f(Spral)

= 10 /10+ 2+11+3
= 10/26

Nullwahrscheinlichkeiten müssen vermieden werden, außer das entsprechende n-Gramm ist unmöglich.

Prinzip der Parameterglättung: Man nimmt den beobachteten n-Grammen etwas Wahrscheinlichkeit weg und verteilt sie an die nicht beobachteten n-Gramme.

Example

Katz'sche Backoff-Glättung

Backoff-Glättung wird für bedingte Wahrscheinlichkeiten verwendet.

Statt die Wahrscheinlichkeitsmasse gleichmäßig über alle unbeobachteten Wörter zu verteilen, werden Sie gemäß einer Backoff-Verteilung mit kleinerem Kontext verteilt.

$$p(w_i|w_{i-k}^{i-1}) = \begin{cases} \frac{f(w_{i-k}^i)-\delta_k}{f(w_{i-k}^{i-1})} & \text{falls } f(w_{i-k}^i) > 0 \\ \alpha(w_{i-k}^{i-1})\, p(w_i|w_{i-k+1}^{i-1}) & \text{sonst} \end{cases}$$

Dabei gilt: $f(w_{i-k}^{i-1}) = \sum_w f(w_{i-k}^{i-1}w)$

Der Backoff-Faktor α stellt sicher, dass sich eine Wk.-Verteilung ergibt.

Die Backoff-Verteilung $p(w_i|w_{i-k+1}^{i-1})$ wird rekursiv auf dieselbe Weise geglättet.

Die Unigramm-Wahrscheinlichkeiten $p(w_i)$ werden entweder mit relativen Häufigkeiten $f(w_i)/N$ geschätzt oder rekursiv mit einer uniformen Verteilung $p_{\text{uniform}}(w_i) = 1/B$ geglättet.

Für jede n-Gramm-Größe k wird ein eigener Discount δ_k berechnet.

We use this smoothing method in Übung4

Backoff-Glättung mit Interpolation

freq of "ngram"

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

discount value of k-gram

Hier wird jeweils die mit α gewichtete Backoff-Wahrscheinlichkeit hinzuaddiert.

we call this part "context"

$$p(w_i|w_{i-k}^{i-1}) = \frac{\max(0, f(w_{i-k}^i) - \delta_k)}{f(w_{i-k}^{i-1})} + \alpha(w_{i-k}^{i-1})\, p(w_i|w_{i-k+\underline{1}}^{i-1})$$

the reduced context

we call this part "word"

freq of "context"

Auch hier gilt: $f(w_{i-k}^{i-1}) = \sum_w f(w_{i-k}^{i-1}w)$

p_backoff

Die max-Operation verhindert negative Häufigkeiten.

backoff factor of the context C $\alpha(C) = 1 - \sum_{w:f(C,w)>0} \frac{f(C,w)-\delta}{f(C)}$

Example for a 3 gram model, k= 2

$$P(\text{Buch}|\text{ das, rote}) = \frac{\max(0, f(\text{das, rote, Buch}))-\text{discount\_k}}{f(\text{das rote})} + \text{backoff\_factor}_{(\text{das, rote})} * P(\text{Buch}|\ \textbf{rote})$$

Example

Let's call this part **p*(word | context)**

$$p(w_i|w_{i-k}^{i-1}) = \frac{\max(0, f(w_{i-k}^{i}) - \delta_k)}{f(w_{i-k}^{i-1})} + \alpha(w_{i-k}^{i-1})\, p(w_i|w_{i-k+1}^{i-1})$$

Assuming a 3 gram markov model, we have to estimate p(Buch|das, rote)

P_backoff (Buch | rote)

p(Buch | das, rote) = p* (Buch | das, rote) + α(das, rote) · (p* (Buch | rote) + α(rote) · p* (Buch))

$p^*(\text{Buch} \mid \text{das, rote}) = (\text{f(das, rote, Buch)} - \delta_3) \;/\; \text{f(das, rote)}$

$p^*(\text{Buch} \mid \text{rote}) = (\text{f(rote, Buch)} - \delta_2) \;/\; \text{f(rote)}$

$p^*(\text{Buch}) = \text{f(Buch)} \;/\; \text{N}$

Note:
p_backoff can be calculated using a standard ngram frequency counting method or the Kneser-Ney ngram frequency counting method

While **p*(Buch| das, rote)** can use only the normal method for counting ngram frequency

How to calculate each p*(word | context)?

Given the following 3-gram frequencies
f(das, rote, Buch) = 5
f(dieses, rote, Buch) = 2
f(gute, rote, Buch) = 4
f(das, gelbe, Buch) = 1
f(das, rote, Kleid) = 2
f(dieses, rote, Kleid) = 2
f(das, rote, Haus) = 8

Kneser-Ney Backoff-Verteilung

Bei der bisherigen Berechnung der n-1-Gramm-Häufigkeiten zur Schätzung der Backoff-Verteilung summieren wir die Häufigkeiten über alle möglichen Vorgängerwörter w':

$$f(C, w) = \sum_{w'} f(w', C, w)$$

C ist eine (eventuell leere) Folge von Wörtern.

Bei Kneser-Ney zählen wir, wieviele **unterschiedliche** Wörter vor dem Wort-n-Gramm aufgetreten sind:

$$f^*(C, w) = \sum_{w'} \mathbf{1}_{f(w', C, w) > 0}$$

$\mathbf{1}_{test}$ ist 1, falls *test* wahr ist und sonst 0.

Die Kneser-Ney-Methode zählt n-Gramm-Types (statt -Tokens).

Aus den so ermittelten Häufigkeiten, werden dann die Parameter der Backoff-Wahrscheinlichkeits-Verteilungen geschätzt.

$$p_{backoff}(w|C) = \frac{f^*(C, w)}{\sum_{w'} f^*(C, w')}$$

p*(das,rote,Buch) = ?

Standard-Method
p*(das,rote,Buch) = f(das, rote, Buch) - discount(3-gram) / f(das, rote)
N1 = 1, N2 = 3
discount(3-gram) = N1 / N1+ (2 * N2) = 1 / 1+ (2*3) = 1 / 5 = 0.14
f(das, rote, Buch) = 5
f(das, rote) = f(das, rote, Buch) + f(das, rote, kleid) + f(das, rote, Haus)
= 5 + 2+ 8 = 15

p*(das,rote,Buch) = 5 - (0.14) / 15
....
p*(Buch | rote) = ?

Standard-Method
We need to the frequencies of bigram, which we derive from the existing 3-gram frequencies (for each 3-gram, we remove the first words, then we get the bigrams)
f(rote, Buch) = f(das,rote,Buch) + f(dieses,rote,Buch) + f(gute, rote, Buch) = 5 + 2+ 4 = 11
f(gelbe, Buch) = f(das, gelbe, Buch) = 1
f(rote, Kleid) = f(das, rote, Kleid) + f(dieses, rote, Kleid) = 2 + 2 = 4
f(rote, Haus) = f(das, rote, Haus) = 8

p*(Buch | rote) = f(rote, Buch) - Discount(2-gram) / f(rote)
N1 = 1, N2 = 0
discount(2-gram) = = N1 / N1+ (2 * N2) = 1 / 1+ (2*0) = 1
f(rote) = f(rote, Buch) + f(rote,Kleid) + f(rote, Haus) = 11 + 4+ 8 = 23
p*(Buch | rote) = 11 - 1 / 23

Kneser-Key method
f(rote, Buch) = 1(das,rote,Buch) + 1(dieses,rote,Buch) +1(gute, rote, Buch) = 1+1+1 = 3
f(gelbe, Buch) = 1(das, gelbe, Buch) = 1
f(rote, Kleid) = 1(das, rote, Kleid) + 1(dieses, rote, Kleid) = 2
f(rote, Haus) = 1(das, rote, Haus) = 1
N1 = 2, N2 =1
discount(2-gram) = = N1 / N1+ (2 * N2) = 2 / 2+ (2*1) = 2/4 = 0.5
f(rote) = f(rote, Buch) + f(rote,Kleid) + f(rote, Haus) = 3 + 2 + 1 = 6
p*(Buch | rote) = f(rote, Buch) - Discount(2-gram) / f(rote) = 3 - 0.5 / 6

....

p*(Buch) = ?
p*(Buch) = f(Buch) / N
standard
f(buch) = f(rote, Buch) + f(gelbe, Buch) = 11+ 1 = 12
f(Kleid) = f(rote, Kleid) =4
f(Haus) = f(rote, Haus) = 8
N = f(empty context) = f(buch) + f(Kleid) + f(Haus) = 12 + 4 + 8 =24
p*(Buch) = 12 / 24

Kneser-Key
f(buch) = 1(rote, Buch) + 1(gelbe, Buch) = 1+ 1 = 2
f(Kleid) = 1(rote, Kleid) =1
f(Haus) = 1(rote, Haus) = 1
N = f(empty context) = f(buch) + f(Kleid) + f(Haus) = 2 + 1+ 1 = 4
p*(Buch) = 2 /4

....

How to compute a backoff factor?

$$\alpha(C) = 1 - \sum_{w:f(C,w)>0} \frac{f(C,w) - \delta}{f(C)}$$

This part is p*(context, word) that we have calculated previously

Let “das, rote” be the context of “das, rote, buch”

To compute a backoff factor of a context, we have to first compute all **p*(context,word)** of all possible word.
It means,

backoff(das,rote) = 1 - [p*(das, rote, Buch) + p*(das, rote, Kleid), + p*(das, rote, Haus)]

backoff(rote) = 1- [p*(rote, Buch) + p*(rote, Kleid) + p*(rote, Haus)]

....
Important note

Schmid just edited some explanations about the discount value (Slide 80, 84)

Katz'sche Backoff-Glättung

Backoff-Glättung wird für bedingte Wahrscheinlichkeiten verwendet.

Statt die Wahrscheinlichkeitsmasse gleichmäßig über alle unbeobachteten Wörter zu verteilen, werden Sie gemäß einer Backoff-Verteilung mit kleinerem Kontext verteilt.

$$p(w_i|w_{i-k}^{i-1}) = \begin{cases} \frac{f(w_{i-k}^{i})-\delta_k}{f(w_{i-k}^{i-1})} & \text{falls } f(w_{i-k}^{i}) > 0 \\ \alpha(w_{i-k}^{i-1})\ p(w_i|w_{i-k+1}^{i-1}) & \text{sonst} \end{cases}$$

Dabei gilt: $f(w_{i-k}^{i-1}) = \sum_w f(w_{i-k}^{i-1}w)$

Der Backoff-Faktor α stellt sicher, dass sich eine Wk.-Verteilung ergibt.

Die Backoff-Verteilung $p(w_i|w_{i-k+1}^{i-1})$ wird rekursiv auf dieselbe Weise geglättet.

Die Unigramm-Wahrscheinlichkeiten $p(w_i)$ werden entweder mit relativen Häufigkeiten $f(w_i)/N$ geschätzt oder rekursiv mit einer uniformen Verteilung $p_{\text{uniform}}(w_i) = 1/B$ geglättet.

Für jede n-Gramm-Größe $k+1$ wird ein eigener Discount δ_k berechnet.

Previously Delta_k stood for Discount of k-gram.
For example, Delta_3 is calculated using the 3-gram-freq.

Now since we use k normally when refer to the lenght of the context of an n-gram, this notation of discount would be misleading.

There for we define Delta_k now as Delta of (k+1 gram).
For example, if you see Delta_2, it means it is the Delta of the 3-gram, and you will calculate it from the 3-gram-freq

Interpolierte Backoff-Glättung

Beispiel:

$$p(\text{Buch} \mid \text{das, rote}) = p^*(\text{Buch} \mid \text{das, rote}) + \alpha(\text{das, rote}) \cdot (p^*(\text{Buch} \mid \text{rote}) + \alpha(\text{rote}) \cdot p^*(\text{Buch}))$$

$p^*(\text{Buch} \mid \text{das, rote}) = (f(\text{das, rote, Buch}) - \delta_2) \,/\, f(\text{das, rote})$

$p^*(\text{Buch} \mid \text{rote}) = (f(\text{rote, Buch}) - \delta_1) \,/\, f(\text{rote})$

$p^*(\text{Buch}) = f(\text{Buch}) \,/\, N$