Linguistische Informatik

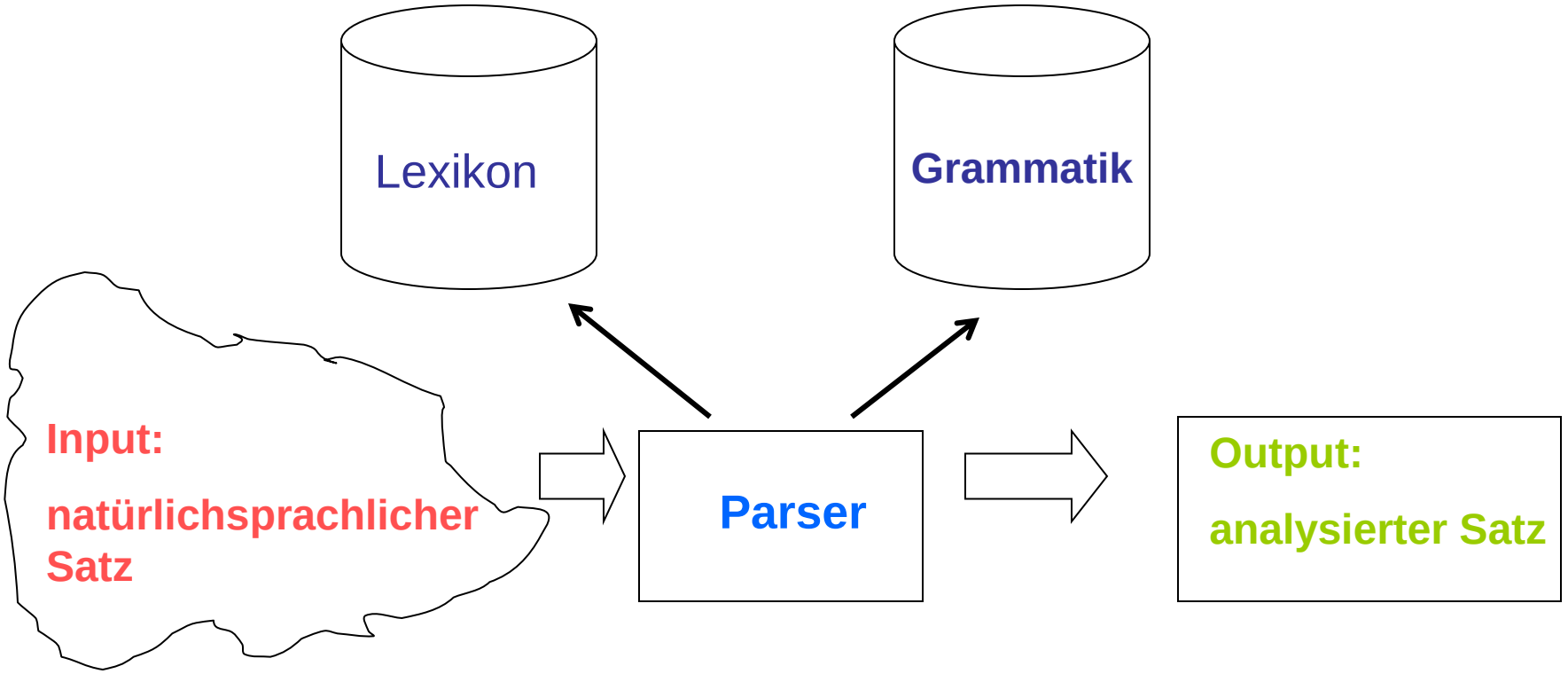
Gerhard Heyer
Universität Leipzig
heyer@informatik.uni-leipzig.de

UNIVERSITÄT LEIPZIG
Institut für Informatik

ASV
Automatische Sprachverarbeitung

Automatische Sprachverarbeitung

Das klassische Verarbeitungsmodell



Automatische Sprachverarbeitung

Statistisches Sprachmodell

Fragestellung:

Ist ein Satz S, gegeben durch eine Folge von Wortformen, ein korrekter Satz einer Sprache L?

Wege der Beantwortung (durch den Computer)

a) Auflistung aller Sätze der Sprache L und Suche nach Satz S in dieser Liste

→ für nat. Sprache nicht möglich

b) Konstruktionsvorschrift (Grammatik) für L erstellen und Satz S prüfen

→ für nat. Sprache sehr komplex (Einbeziehung von Mundarten, fachsprachl. Besonderheiten)

→ Prüfung von S aufwändige Operation

→ semantische Korrektheit??

c) probabilistisches Sprachmodell

→ Berechnung der Wahrscheinlichkeit, daß S ein korrekter Satz von L ist. (eigentlich: Wahrscheinlichkeit, daß S in einem Text der Sprache L vorkommt.)

Automatische Sprachverarbeitung

Grundlage

Basis der Berechnung: Verteilung der Wortformen in Texten der Sprache L

- → **Auszählen des Auftretens der Wortformkombinationen**
- → **Bestimmung der Wahrscheinlichkeiten für das Aufeinanderfolgen von Wortformen**
- → **Berechnung der Wahrscheinlichkeit des Satzes S aus den einzelnen Wahrscheinlichkeiten der Wortformenabfolge von S**

In Texten der Sprache L treten „sinnvolle“ Wortformkombinationen auf

- → **Berücksichtigung sowohl syntaktischer als auch semantischer Aspekte**

Notation

Sei X eine Zufallsvariable mit einer endlichen Menge V(X) von m Ereignissen.

| X = x | sei die Anzahl von Ereignissen bei denen X den Wert x hat (d. h. $x \in V(X)$).

Die Wahrscheinlichkeit des Auftretens von x_i (Abkürzung $P(x_i)$ ist:

$$P(X = x_i) = \frac{|x_i|}{\sum_{j=1}^{m} |x_j|}$$

Automatische Sprachverarbeitung

Beispiel

Sei W das Auftreten einer bestimmten Wortform w_i aus der Menge der m Wortformen eines Textes.
Die Wahrscheinlichkeit des Auftretens der i-ten Wortform w_i ist dann:

$$P(W = w_i) = \frac{|w_i|}{\sum_{j=1}^{m} |w_j|}$$

Bedingte Wahrscheinlichkeit

Bedingte Wahrscheinlichkeit

Die Wahrscheinlichkeit für das Eintreten eines Ereignisses **X** unter der Voraussetzung , dass das Ereignis **y** schon eingetreten ist, heißt **bedingte Wahrscheinlichkeit P (x | y).**

$$P(x \mid y) = \frac{P(x, y)}{P(y)}$$

Sind x und y voneinander unabhängig, so gilt:

$$P(x, y) = P(x) * P(y)$$

Die bedingte Wahrscheinlichkeit unabhängiger Ereignisse ist:

$$P(x \mid y) = P(x)$$

Automatische Sprachverarbeitung

Beispiel

Die bedingte Wahrscheinlichkeit des Aufeinanderfolgens zweier Wortformen ist:

$$P(W_2 = w_j \mid W_1 = w_i) = \frac{|W_1 = w_i, W_2 = w_j|}{|W_1 = w_i|}$$

Bayessches Gesetz

Wenn die Ereignisse **x ∈ V (x)** einander **paarweise ausschließen** und die Menge der **m Elementarereignisse** ausschöpfen**, *so gilt für die bedingte Wahrscheinlichkeit*:**

$$P(x \mid y) = \frac{P(x) * P(y \mid x)}{P(y)}$$

Verallgemeinerungen

$$P(w, x \mid y, z) = \frac{P(w, x) * P(y, z \mid w, x)}{P(y, z)}$$

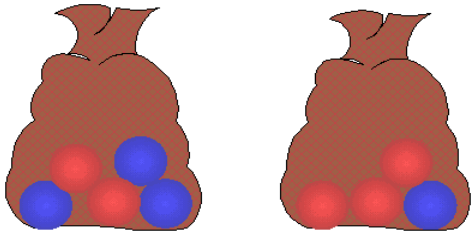
$$P(w_1, w_2, w_3, \ldots, w_n) = P(w_1) * P(w_2 \mid w_1) * P(w_3 \mid w_1, w_2) \ldots * P(w_n \mid w_1, \ldots, w_{n-1})$$

Bayessches Theorem: Beispiel

Zwei Säcke enthalten rote und blaue Kugeln.

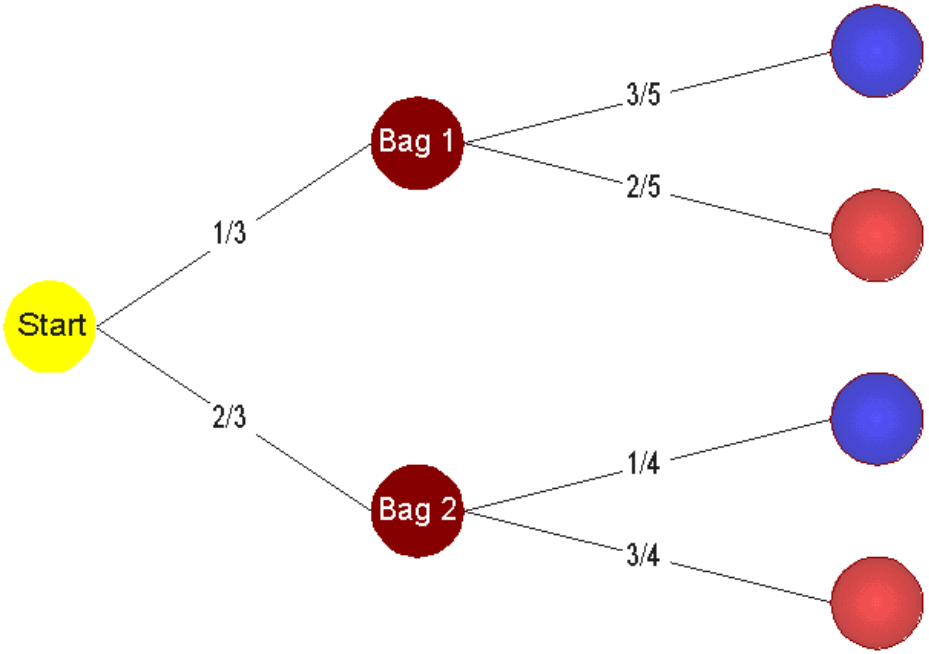
Im ersten Schritt wird gewürfelt, um einen Sack auszuwählen. Wird eine „1“ oder „2“ gewürfelt, wählen wir Sack Nr. 1, sonst Sack Nr. 2

Danach wird eine Kugel zufällig aus diesem Sack entnommen.



Wir haben eine blaue Kugel gezogen. Wie groß ist die Wahrscheinlichkeit, dass sie aus Sack Nr. 1 stammt?

Bayessche Theorem: Beispiel ausgerechnet



$$P(Sack_1 \mid Blau) = \frac{\begin{array}{c}\text{Produkt der Wahrscheinlichkeiten}\\ \text{zu einer blauen Kugel aus Sack Nr.1}\end{array}}{\begin{array}{c}\text{Summe aller solcher Produkte,}\\ \text{die zu blauen Kugeln führen}\end{array}} = \frac{\frac{1}{3}\cdot\frac{3}{5}}{\frac{1}{3}\cdot\frac{3}{5}+\frac{2}{3}\cdot\frac{1}{4}} = \frac{6}{11}$$

Bayessche Formel allgemein

P*(*h*|*D*) = *a posteriori* Wahrscheinlichkeit von *h

P*(*h*) = *a priori* Wahrscheinlichkeit von *h

***P*(*D*|*h*) = Wahrscheinlichkeit des Ereignisses D unter der Hypothese h**

***P*(*D*) = Wahrscheinlichkeit des Ereignisses D unabhängig von einer Hypothese**

Im Beispiel: D = blaue Kugel gezogen

h = vorher Sack 1 ausgewählt.

Die Maximum a posteriori Hypothese

$$h_{MAP} = \arg\max_{h \in H} P(h \mid D)$$

$$= \arg\max_{h \in H} \frac{P(D \mid h) P(h)}{P(D)}$$

$$= \arg\max_{h \in H} P(D \mid h) P(h)$$

Maximum likelihood Hypothese: Falls alle Hypothesen die gleiche a priori Wahrscheinlichkeit haben, sprechen wir von der Maximum likelihood Hypothese:

$$h_{ML} = \arg\max_{h \in H} P(D \mid h)$$

Anwendung: Statistisches Modell der Deutschen Sprache

Weise allen Folgen von Wortformen der Länge n eine Wahrscheinlichkeit zu, d. h.

$$P (W_{1,n} = w_{1,n}) \qquad \text{für alle Folgen } w_{1,n} .$$

$W_{1,n}$ ist eine Folge von n Zufallsvariablen $w_1, w_2, \ldots , w_n$, die jeweils irgendeine Wortform des Deutschen als Wert nehmen können, und $w_{1,n}$ ist eine konkrete Folge von deutschen Wortformen.

Diese Folge kann auf der Grundlage der verallgemeinerten ***Bayesschen Regel*** berechnet werden.

$$P (w_{1,n}) = P (w_1) * P (w_2 \mid w_1) * P (w_3 \mid w_{1,2}) * \ldots * P (w_n \mid w_{1,n-1})$$

Why is this so?

A sequence of wordforms can be considered a conditional probability.

For two wordforms we have

$$(1) \qquad P(w_i, w_j) = P(w_i) \cdot P(w_j \mid w_i)$$

For three wordforms we have

$$(2) \qquad P(w_i, w_j, w_k) = P((w_i, w_j), w_k) = P(w_i, w_j) \cdot P(w_k \mid w_i, w_j)$$
$$= P(w_i) \cdot P(w_j \mid w_i) \cdot P(w_k \mid w_i, w_j)$$

Probability of a sentence

Generalising (2), the probability of any sentence of length n can be stated as:

$$(3) \qquad P(w_1, w_2, w_3, \ldots, w_n) = P(w_1) \cdot P(w_2 \mid w_1) \cdot P(w_3 \mid w_1, w_2) \cdot \ldots \cdot P(w_n \mid w_1, w_2, w_3, \ldots, w_{n-1})$$

Using the *maximum likelihood estimate* we can compute this probability by counting the occurrences of wordforms:

$$(4) \qquad P(w_1, w_2, w_3, \ldots, w_n) = \frac{|w_1|}{\sum_{k=1}^{m} |w_k|} \cdot \frac{|w_1, w_2|}{|w_1|} \cdot \frac{|w_1, w_2, w_3|}{|w_1, w_2|} \cdot \ldots \cdot \frac{|w_1, w_2, w_3, \ldots, w_{n-1}, w_n|}{|w_1, w_2, w_3, \ldots, w_{n-1}|} = \frac{|w_1, w_2, w_3, \ldots, w_{n-1}, w_n|}{\sum_{k=1}^{m} |w_k|}$$

Beispiel

Text der Sprache L = Trainingskorpus

(zur Berechnung der Wahrscheinlichkeiten):

Herr Schuhmann liest ein Buch. Herr Schuhmann isst ein Brot. Frau Müller isst ein Hähnchen.

Gehören folgende Sätze zur Sprache L?

a) Herr Schuhmann isst ein Brot.

b) Herr Schuhmann isst ein Hähnchen.

c) Herr Schuhmann liest ein Brot.

d) Frau Müller liest ein Buch.

Bedingte Wahrscheinlichkeiten

Für bedingte Wahrscheinlichkeit der Kombination von zwei Wortformen **gilt:**

$$(5) \qquad P(w_j \mid w_i) == \frac{|w_i, w_j|}{|w_i|}$$

Bsp.: P(Schuhmann | Herr) = |Herr, Schuhmann| / |Herr| =
P(liest | Schuhmann) = |Schuhmann, liest| / |Schuhmann| =
P(ein | liest) = |liest, ein| / |liest| =
P(liest | Müller) = |Müller, liest| / |Müller| =

Bedingte Wahrscheinlichkeiten

Für das Aufeinanderfolgen dreier Wortformen ergibt sich:

$$(6) \qquad P(w_i, w_j, w_k) = P((w_i, w_j), w_k) = P(w_i, w_j) \cdot P(w_k \mid w_i, w_j)$$
$$= P(w_i) \cdot P(w_j \mid w_i) \cdot P(w_k \mid w_i, w_j)$$

**P(Herr, Schuhmann, liest) = P(Herr)*P(Schuhmann | Herr) *
P(liest | Herr, Schuhmann)**

Wahrscheinlichkeit eines Satzes

Für den gesamten Satz erhalten wir nach dem *maximum likelihood estimate:*

$$(7) \qquad P(w_1, w_2, w_3, \ldots, w_n)$$

$$= \frac{|w_1|}{\sum_{k=1}^{m} |w_k|} \cdot \frac{|w_1, w_2|}{|w_1|} \cdot \frac{|w_1, w_2, w_3|}{|w_1, w_2|} \cdot \ldots \cdot \frac{|w_1, w_2, w_3, \ldots, w_{n-1}, w_n|}{|w_1, w_2, w_3, \ldots, w_{n-1}|} = \frac{|w_1, w_2, w_3, \ldots, w_{n-1}, w_n|}{\sum_{k=1}^{m} |w_k|}$$

P(Herr, Schuhmann, isst, ein, Brot)
= |Herr, Schuhmann, isst, ein, Brot| / Anzahl aller Wortformen
=
P(Herr, Schuhmann, liest, ein, Brot)
= |Herr, Schuhmann, liest, ein, Brot| / Anzahl aller Wortformen
=
P(Frau, Müller, liest, ein, Buch)
= |Frau, Müller, liest, ein, Buch| / Anzahl aller Wortformen
=

Automatische Sprachverarbeitung

First summary

Approach up till now not really convincing:

- **To compute the probability of any sentence we would have to compute the occurrence of any combination of n wordforms of the total vocabulary N of some language ($n \in N$)**
  - huge list
  - only possible for some specific corpus, not for language L in general
- **Insufficient treatment of sentences that do not appear in the trainings corpus**
  - sentences not contained in the trainings corpus are being assigned probability 0
  - this happens, even if we only change wordorder
  - however, any trainings corpus can only contain a finite number of sentences
  - we do not adequately compute the probability of sentences that do not occur in the trainings corpus (in fact, this is the majority of sentences)

Wahrscheinlichkeit eines Satzes
Bi- und Trigramme

Beobachtung:

Beziehungen zwischen den Wortformen eines Satzes stark lokal geprägt. Wortformen sind zu Phrasen gruppiert.

→ **Wahrscheinlichkeit des Auftretens von Wortform w_i stark von restlichen Wortformen der gleichen Phrase, weniger stark von Wortformen anderer Phrasen beeinflußt. Wird aber von n-Gramm-Modell nicht ausgenutzt.**

→ **Es genügt, die Wahrscheinlichkeit des Auftretens von Wortformen zu approximieren. Nur wenige Vorgänger sind zu berücksichtigen.**

- ausreichend: Verwendung von lediglich 2 vorausgehenden Wortformen
- mehr Vorgänger bringen kaum mehr Genauigkeit, erhöhen jedoch enorm den Rechenaufwand

Berechnungsgrundlage *n-gram Modell*

Annahme, dass nur die vorangehenden n-1 Wortformen von Einfluss auf die Wahrscheinlichkeit der nächsten Wortform sind, wobei ***n = 3 (daher tri-gram)***

(8) $P(w_n \mid w_1, \ldots , w_{n-1}) = P(w_n \mid w_{n-2}, w_{n-1})$

$$P(w_{1,n}) = P(w_1) * P(w_2 \mid w_1) * P(w_3 \mid w_{1,2}) * \ldots * P(w_n \mid w_{n-2},w_{n-1})$$

$$= P(w_1) * P(w_2 \mid w_1) * \prod_{i=3}^{n} P(w_i \mid w_{i-2,i-1})$$

$$= \prod_{i=1}^{n} P(w_i \mid w_{i-2},w_{i-1})$$

Abschätzung

Wahrscheinlichkeit der n-ten Wortform wird abgeschätzt zu

$$(9) \qquad P(w_n \mid w_1, w_2, \ldots, w_{n-1}) = \frac{|w_1, w_2, w_3, \ldots, w_{n-1}, w_n|}{|w_1, w_2, w_3, \ldots, w_{n-1}|} \approx \frac{|w_{n-2}, w_{n-1}, w_n|}{|w_{n-2}, w_{n-1}|}$$

→ Auftreten der n-ten Wortform hängt nur noch von den beiden vorangehenden Wortformen ab

für Wahrscheinlichkeit des ganzen Satzes gilt dann:

$$(10) \qquad P(w_1, w_2, w_3, \ldots, w_n)$$

$$= P(w_1) \cdot P(w_2 \mid w_1) \cdot P(w_3 \mid w_1, w_2) \cdot \ldots \cdot P(w_n \mid w_1, w_2, w_3, \ldots, w_{n-1})$$

$$= \frac{|w_1|}{\sum_{k=1}^{m} |w_k|} \cdot \frac{|w_1, w_2|}{|w_1|} \cdot \frac{|w_1, w_2, w_3|}{|w_1, w_2|} \cdot \frac{|w_2, w_3, w_4|}{|w_2, w_3|} \cdot \ldots \cdot \frac{|w_{n-2}, w_{n-1}, w_n|}{|w_{n-2}, w_{n-1}|}$$

Beispiel (1)

P(Herr, Schuhmann, isst, ein, Brot)

= P(Herr) * P(Schuhmann | Herr) * P(isst | Herr,Schuhmann) * P(ein | Schuhmann,isst) * P(Brot | isst,ein)

= |Herr|/|“alle Wortformen“| * |Herr,Schuhmann|/|Herr| * |Herr,Schuhmann,isst|/|Herr,Schuhmann| * |Schuhmann,isst,ein|/|Schuhmann,isst| * |isst,ein,Brot|/|isst,ein|

= |Herr,Schuhmann,isst|*|Schuhmann,isst,ein|*|isst,ein,Brot| / |“alle Wortformen“|*|Schuhmann,isst|*|isst,ein|

=

Würdigung

- **höchstens Dreierkombinationen von Wortformen zu betrachten ermöglicht starke Reduktion der Komplexität (bzgl. Kombinationsmöglichkeiten)**

- **Wahrscheinlichkeit von Satz S nicht sofort 0, wenn S nicht im Trainingskorpus vorkommt. Hauptsache alle Kombinationen von 2 und 3 aufeinanderfolgenden Wortformen in S treten im Trainingskorpus auf.**

- **Trotzdem Schwierigkeit bei extrem seltenen Wortformen: Tritt eine solche Wortform in S auf, ist Wahrscheinlichkeit groß, dass eines der Trigramme nicht im Korpus vorkommt, daher P(S) = 0**

Glättung / Smoothing

Behandeln Problem der extrem seltenen bzw. der nicht im Trainingskorpus auftretenden Wortformen.

zwei grundsätzliche Strategien

- **Backing Off Smoothing**

  Zurückgreifen auf n-Gramme geringerer Stufe bei fehlender Information

- **Discounting**

  Abzug eines kleinen Betrages von den Wahrscheinlichkeiten, die mittels Trainingskorpus berechnet wurden und Verteilung auf nicht aufgetretene Wortformkombinationen

Beispiel
Add-One-Smoothing

Add-One-Smoothing

- **zur tatsächlichen Frequenz im Trainingskorpus wird „sicherheitshalber“ stets 1 addiert**
- **damit alle Wahrscheinlichkeiten größer als 0**
- **schlecht für niederfrequente Wörter: In ihrer Nachbarschaft ist praktisch alles erlaubt. (Kaum Unterschied zwischen tatsächlich beobachteter Nachbarschaft und nicht beobachteter Nachbarschaft)**

Beispiel

Add-One-Smoothing – Beispiel

P(Herr, Schuhmann, isst, ein, Brot)

= P(Herr) * P(Schuhmann | Herr) * P(isst | Herr,Schuhmann) * P(ein | Schuhmann,isst) * P(Brot | isst,ein)

=

P(Frau, Müller, liest, ein, Buch)

=

P(Herr, Schuhmann, liest, ein, Brot)

=

Beispiel
lineare Interpolation

Einfache lineare Interpolation

- **Einbeziehung von Bi- und Unigrammen**
- **Idee:**
  - **Dreierwortformkombination des Satzes tritt nicht im Trainingskorpus auf**
    → Ausnutzen der Information über Zweierwortformkombinationen
  - **Zweierwortformkombination des Satzes tritt nicht im Trainingskorpus auf**
    → Ausnutzen der Information über die einzelne Wortform
- **Umsetzung:**
  - **Ausnutzung aller drei Informationsquellen (Tri-, Bi- und Unigramme)**
  - **aber unterschiedliche Gewichtung der Quellen, da Trigramm aussagekräftiger als Bigramm usw.**

Automatische Sprachverarbeitung

Beispiel
lineare Interpolation – Formel

Statt

$$(9) \qquad P(w_n \mid w_1, w_2, \ldots, w_{n-1}) \approx \frac{|w_{n-2}, w_{n-1}, w_n|}{|w_{n-2}, w_{n-1}|}$$

wird

$$(11) \qquad P(w_n \mid w_1, w_2, \ldots, w_{n-1}) \approx \lambda_1 \cdot \frac{|w_{n-2}, w_{n-1}, w_n|}{|w_{n-2}, w_{n-1}|} + \lambda_2 \cdot \frac{|w_{n-1}, w_n|}{|w_{n-1}|} + \lambda_3 \cdot \frac{|w_n|}{\sum_{k=1}^{m} |w_k|}$$

angesetzt.

Für die Gewichtungsfaktoren λ_i gilt: $0 \leq \lambda_i \leq 1$; $\Sigma \lambda_i = 1$. Sie sind entsprechend dem Trainingskorpus festzulegen (z.B. durch Hidden-Markov-Modelle).

Automatische Sprachverarbeitung

Beispiel

lineare Interpolation – Beispiel

Sei |“alle Wortformen“| = n

P(Herr, Schuhmann, isst, ein, Brot)

$$
\begin{aligned}
&= P(Herr) * P(Schuhmann \mid Herr) * P(isst \mid Herr,Schuhmann) * \\
&\quad P(ein \mid Schuhmann,isst) * P(Brot \mid isst,ein) \\
&= (\lambda_3 * (|Herr|/n)) * \\
&\quad (\lambda_2 * (|Herr,Schuhmann|/|Herr|) + \lambda_3 * (|Schuhmann|/n)) * \\
&\quad (\lambda_1 * (|Herr,Schuhmann,isst|/|Herr,Schuhmann|) + \\
&\quad\ \lambda_2 * (|Schuhmann,isst|/|Schuhmann|) + \lambda_3 * (|isst|/n)) * \\
&\quad (\lambda_1 * (|Schuhmann,isst,ein|/|Schuhmann,isst|) + \\
&\quad\ \lambda_2 * (|isst,ein|/|isst|) + \lambda_3 * (|ein|/n)) * \\
&\quad (\lambda_1 * (|isst,ein,Brot|/|isst,ein|) + \lambda_2 * (|ein,Brot|/|ein|) + \lambda_3 * (|Brot|/n))
\end{aligned}
$$

Anwendung: Automatische Klassifikation

Ein Klassifikator ist eine möglichst gute Annäherung an die unbekannte Zielfunktion φ

$$\varphi: DxC \rightarrow \{T , F\}$$

mit den Dokumenten D und den Kategorien C, welche jedem Paar

$$<d j , ci> \in DxC$$

einen Wahrheitswert zuweist.

Zielfunktion kann durch Trainingsdaten angenähert werden.

Erläuterung der Definition

	d_1	...	...	d_j	...	...	d_n
c_1	a_{11}	...	...	a_{1j}	...	...	a_{1n}
...	...	...	...	...	...	...	...
c_i	a_{i1}	...	...	a_{ij}	...	...	a_{in}
...	...	...	...	...	...	...	...
c_m	a_{m1}	...	...	a_{mj}	...	...	a_{mn}

Dokument Klassifikation: Berechnung eines Wertes {0,1} zu jedem Eintrag in der Dokument-Kategorie-Matrix.

C = {c_1,....., c_m} ist eine Menge vordefinierter Kategorien.

D = {d_1,..... d_n} ist eine Menge von Dokumenten.

1 für a_{ij}: Zuordnung von d_j zu c_i

0 für a_{ij}: keine Zuordnung von d_j zu c_i

Naive Bayes

Das *Wahrscheinlichkeitsmodell* für einen Klassifikator ist die bedingte Wahrscheinlichkeit

$$p(C|F_1,\ldots,F_n)$$

mit einer Klassenvariablen *C*

und Feature Variablen F_1 bis F_n.

Mit Bayes Theorem erhalten wir die Umformung

$$p(C|F_1,\ldots,F_n) = \frac{p(C)\ p(F_1,\ldots,F_n|C)}{p(F_1,\ldots,F_n)}.$$

Naive Bayes (2)

Der Zähler beschreibt dabei die Wahrscheinlichkeit, dass in einem Text mit der Klasse *C* die Features F_1 bis F_n gemeinsam auftreten:

$$p(C, F_1, \ldots, F_n)$$

Dieses Modell der gemeinsamen Wahrscheinlichkeit kann als bedingte Wahrscheinlichkeit umformuliert werden:

$$\begin{aligned} &= p(C)\ p(F_1, \ldots, F_n|C) \\ &= p(C)\ p(F_1|C)\ p(F_2, \ldots, F_n|C, F_1) \\ &= p(C)\ p(F_1|C)\ p(F_2|C, F_1)\ p(F_3, \ldots, F_n|C, F_1, F_2) \end{aligned}$$

Naive Bayes (3)

Die „naive“ Annahme besagt nun, dass jedes Feature F_i unabhängig von jedem anderen Feature F_j ist,

$$p(F_i|C, F_j) = p(F_i|C)$$

Das Modell der gemeinsamen Wahrscheinlichkeit kann damit vereinfacht werden zu

$$p(C, F_1, \ldots, F_n) = p(C)\ p(F_1|C)\ p(F_2|C)\ p(F_3|C)\ \cdots$$

$$= p(C) \prod_{i=1}^{n} p(F_i|C).$$

Naive Bayes (4) - Klassifikator

Der „naive“ Bayes‘sche Klassifikator verbindet das naive Wahrscheinlichkeitsmodell mit der *maximum a posteriori (MAP)* Entscheidungsregel:

Wähle diejenige Klasse, die am wahrscheinlichsten ist.

$$\text{classify}(f_1, \ldots, f_n) = \text{argmax}_c \; p(C = c) \prod_{i=1}^{n} p(F_i = f_i | C = c)$$

Anwendung: POS Tagging

Ausgangsmaterial: Satz

Ziel: Zuordnung: Wort – part-of-speech (Wortart)

Bsp. (aus Manning/Schütze: Foundations of statistical natural language processing)

The representative put chairs on the table.

AT NN VBD NNS IN AT NN

bzw.

AT JJ NN VBZ IN AT NN

Automatische Sprachverarbeitung

Tag set (nach Manning/Schütze)

tag	part of speech
AT	aticle
BEZ	the word *is*
IN	preposition
JJ	adjective
JJR	comperative adjective
MD	modal
NN	singular or mass noun
NNP	singular proper noun
NNS	plural noun
PERIOD	. : ? !
PN	personal pronoun
RB	adverb

tag	part of speech
RBR	comparative adverb
TO	the word *to*
VB	verb, base form
VBD	verb, past tense
VBG	verb, present paticiple, gerund
VBN	verb, past participle
VBP	verb, non-3rd person singular present
VBZ	verb, 3rd singular present
WDT	wh-determiner (what, which)

Tagging versus parsing

Parsing

- **Ermittlung der syntaktischen Struktur eines Satzes.**
- **→ Abhängigkeiten**
- **→ Phrasen**

Tagging

- **Bestimmung der syntaktischen Kategorien der Wörter eines Satzes.**
- **→ wesentlich einfacher**

Automatische Sprachverarbeitung

Nutzen

als Vorstufe für weitere Verarbeitung der Texte

→ Auflösung von syntaktischen Mehrdeutigkeiten im Text

z.B. bei:

- **Information Extraction**
  Suche nach Werten für Lücken in Mustern/Templates
- **Question Answering**
  - Bestimmung der erfragten Kategorie und anschließendem
  - Vergleich mit vorhandenem Material

Präzisierung der Aufgabe

Für eine gegebene Folge von Wortformen *(die beobachteten Daten)* $w_1, ..., w_n$ wollen wir die wahrscheinlichste POS-Folge von tags *)* $t_1, ..., t_n$ bestimmen.

$$t_1, ..., t_n = \underset{t_1,...,t_n}{\operatorname{argmax}} P(t_1, ..., t_n | w_1, ..., w_n)$$

Die Anwendung des Bayes'schen Gestzes ergibt

$$t_1, ..., t_n = \underset{t_1,...,t_n}{\operatorname{argmax}} \frac{P(w_1, ..., w_n | t_1, ..., t_n) P(t_1, ..., t_n)}{P(w_1, ..., w_n)}$$

Da alle Wortfolgen identisch sein sollen, können wir den Nenner ignorieren:

$$t_1, ..., t_n = \underset{t_1,...,t_n}{\operatorname{argmax}} P(w_1, ..., w_n | t_1, ..., t_n) P(t_1, ..., t_n)$$

Zwei wichtige Annahmen

1. **Die Wahrscheinlichkeit einer Wortform hängt nur von ihrem eigenen POS-tag ab, nicht von dem POS-tag anderer Wortformen**

$$P(w_1, ..., w_n|t_1, ..., t_n) \approx P(w_1|t_1)P(w_2|t_2)...P(w_n|t_n)$$

2. **Die Wahrscheinlichkeit eines POS-tags hängt nur von dem unmittelbar vorangehenden POS-tag ab *(Markov-Eigenschaft erster Ordnung)***

$$P(t_1, ..., t_n) \approx P(t_1|t_0)P(t_2|t_1)...P(t_n|t_{n-1})$$

Vereinfachte Formel

Unter den genannten Annahmen erhalten wir die vereinfachte Formel:

$$t_1, \ldots, t_n = \underset{t_1,\ldots,t_n}{\operatorname{argmax}} \prod_{i=1}^{i=n} P(w_i|t_i) P(t_i|t_{i-1})$$

Für die Berechnung sind jeweils die bedingten Wahrscheinlichkeiten $P(w_i|t_i)$ *(wie wahrscheinlich ist eine bestimmte Wortform, gegeben ein POS-tag)* und $P(t_i|t_{i-1})$ *(wie wahrscheinlich ist ein POS-tag, gegeben ein vorangehendes POS-tag)* auszurechnen und das Produkt zu maximieren.

Beide Aufgaben können mit einem Hidden Markov-Modell (HMM) und dem sog. Viterbi-Algorithmus gelöst werden.

Automatische Sprachverarbeitung

Links und Vertiefungshinweise

Stuttgat-Tübingen_Tagset (STTS):

http://www.sfs.nphil.uni-tuebingen.de/Elwis/stts/stts.html

http://www.coli.uni-sb.de/sfb378/negra-corpus/stts.asc

TnT-Tagger (Trigrams‘n‘Tags) von Thorsten Brants

http://www.coli.uni-sb.de/~thorsten/tnt/

Beispiele für getaggten Text:

http://www.coli.uni-sb.de/sfb378/negra-corpus/negra-corpus.html

Ergänzende Literatur

E.Charniak, Statistical Language Learning, MIT Press: Cambridge (Mass.) 1993

C. Manning und H.Schütze, Foundations of Statistical Natural Language Processing, MIT Press: Cambridge (Mass.) 1999 (^{3}2000)

D.Juravsky, J.Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition, Prentice Hall: San Francisco 2000