

Programmierkurs Python II

Stefan Thater & Michaela Regneri
FR 4.7 Allgemeine Linguistik (Computerlinguistik)
Universität des Saarlandes

Sommersemester 2010



Übersicht

- Einführung zu maschinellem Lernen
- Sprachmodelle
- Textklassifizierung
- Word Sense Disambiguation, Teil 1
- Ein paar praktische Hinweise

Mit dankbaren Credits an
Dietrich Klakow, für die
Folien von SNLP 2007.

Maschinelles Lernen

- Prinzip: lerne aus gesehenen Mustern von linguistischen Informationen -- irgendwas --
- Unterschiedliche Korpora für unterschiedliche Lernziele:
 - Sprachmodelle
 - Parsebäume
 - Übersetzungs-Modelle
 - Textkategorien
 - ...
- Grundsätzlich: je zahlreicher und je informativer die Daten, desto besser

3

Korpusanalyse

- Einfache Beispiele für Informationen (Konkordanzen)

Burschenschaft »Arminia« auch die Karlsruher Händel-Festspiele. Als Girokonto für alle Azubis, Schüler, und richtet sich an Seminar richtet sich vorzugsweise an 2000 (Programme for International

Studenten
Student
Studenten
Studenten
Student

der Technischen Hochschule in in Halle und Göttingen von den usw. für 0 DM Gebühren. Außerdem in den ersten Semestern. des Grundstudiums im 1. oder 3. Assessment) ergeben, dass die

der deutsch spricht und die ausgeschaltet, die und Mineralstoffen kommen. hieß es. « Die Entwicklung hinzu. Die Schulalter. Top Symptome

Kinder antworten meistens
Kinder verschwinden meistens
Kinder reagieren meistens
Kinder gehen meistens
Kinder wachsen meistens
Kinder fallen meistens

auf schwedisch - aus in die Häuser hinein, es noch instinktiv auf nur einmal die Woche in in einem armen soziale durch ein verändertes

4

Formales

- Wahrscheinlichkeit eines Wortes: $p(w = \text{wort})$
= die Wahrscheinlichkeit das Wortes *wort* anzutreffen
- Wahrscheinlichkeit einer Wortfolge:
 - Wahrscheinlichkeit von „der Hund bellt“ =
 $p(w_1 = \text{der}, w_2 = \text{Hund}, w_3 = \text{bellt})$
 - kürzer: $p(\text{der}, \text{Hund}, \text{bellt})$
- Bedingte Wahrscheinlichkeit eines Wortes:
 $p(w_2|w_1)$ = die Wahrscheinlichkeit für w_2 , wenn davor w_1 gesehen wurde

5

Eigenschaften von Wahrscheinlichkeiten

- $p(w) \geq 0$
- $\sum_{i=1}^N p(w_i) = 1$, wobei N die Anzahl aller Worte ist
- $p(w_2|w_1) = \frac{p(w_1, w_2)}{p(w_1)}$ (Definition von bedingter Wahrscheinlichkeit)

6

Einfache Korpus-Analyse

- ist eine „Spaghetti Napoli“ eine Kollokation?
- Prinzip: überprüfe, ob *Spaghetti* und *Napoli* öfter zusammen auftreten, als per Zufall zu erwarten wäre
- Ein numerisches Maß, das Korrelationen misst, ist z.B. „Pointwise Mutual Information“
- auch eingesetzt für Pattern-Suche etc.

0 für unabhängige X und Y
> 0 wenn X und Y öfter
zusammen auftreten als
zufällig

$$PMI = \log \frac{p(X, Y)}{p(X) \times p(Y)}$$

7

Einfache Korpus-Analyse - Beispiel

DeWac-Korpus:	1.4 x 10 ⁹	Wörter
Spaghetti:	2448	Vorkommen
Napoli:	310	Vorkommen
Spaghetti Napoli:	12	Vorkommen

$$PMI = \log \frac{p(X, Y)}{p(X) \times p(Y)}$$

$$p(\text{„Napoli“}) \approx \frac{310}{1.4 \times 10^9}$$

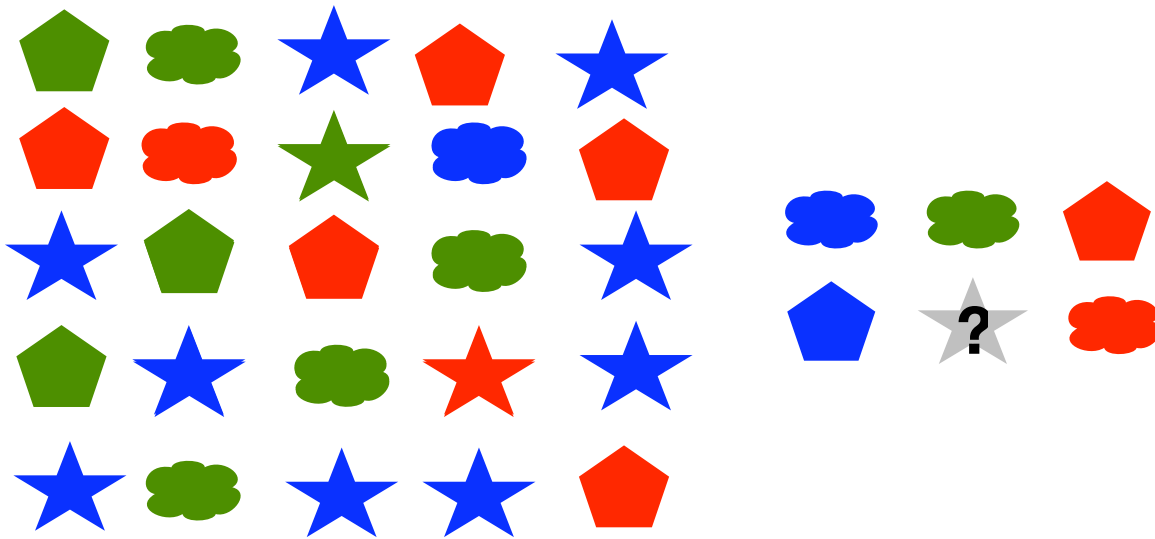
$$p(\text{„Spaghetti“}) \approx \frac{2448}{1.4 \times 10^9}$$

$$p(\text{„Spaghetti Napoli“}) \approx \frac{12}{1.4 \times 10^9}$$

$$PMI = \log \frac{p(\text{„Spaghetti Napoli“})}{p(\text{Spaghetti}) \times p(\text{Napoli})} = 4.3451$$

8

Datenmangel (Sparse Data), schematisch



9

Begriffe zu Sprachmodellen

- **Sprache** (Deutsch, Pascal,...)
- **Wörter**
 - z.B. deutsche Wörter
 - könnten auch Buchstaben, Silben etc. sein
- Die Sprache ist hier die Zusammenfassung gültiger *Wortfolgen*
 - vgl. Automaten: elementare Einheiten = Alphabet
 - hier: elementare Einheiten = Wörter

10

Der Satz von Bayes

$$p(w_2|w_1) = \frac{p(w_1, w_2)}{p(w_1)}$$

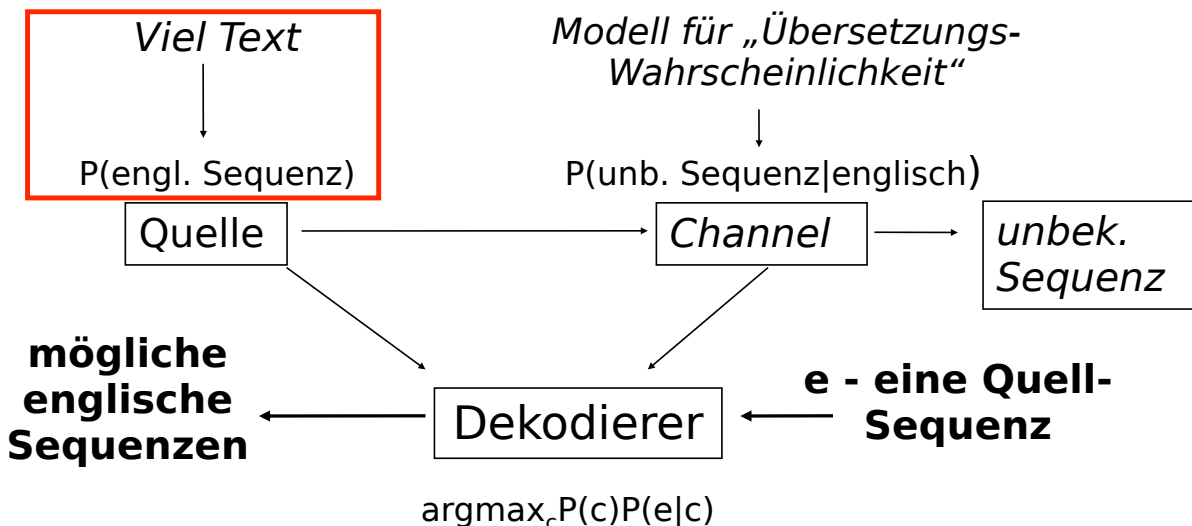
- Theorem: $p(w_2|w_1) * p(w_1) = p(w_1|w_2) * p(w_2)$
- Beweis über Formel-Umstellung:

$$\begin{aligned}
 p(w_2|w_1) * p(w_1) &= \frac{p(w_1, w_2)}{p(w_1)} * p(w_1) \\
 &= p(w_1, w_2) \\
 &= \frac{p(w_1, w_2)}{p(w_2)} * p(w_2) \\
 &= p(w_1|w_2) * p(w_2)
 \end{aligned}$$

- Das wird später noch sehr praktisch...

Das „Noisy Channel Model“

Ziel: ein Englischer Satz aus einer unbekanntem Sequenz (deutsch, Sprachsignal,...)



Sprachmodelle

- Problem: woher bekommt man $p(w = \text{wort})$ oder $p(w_1, w_2, w_3, ..?)$
- Sprachmodelle geben Auskunft darüber, mit welcher Wahrscheinlichkeit eine Wort-Sequenz in einer Sprache als gültige Sequenz auftaucht
 - damit kann man z.B. auch das wahrscheinlichste „nächste Wort“ bestimmen
 - klassische Anwendung:

Der Schimmel galoppiert.

The white horse gallops.
The mold gallops.

13

Sprachmodelle

- Wahrscheinlichkeit für $p(w_1, w_2, w_3, \dots, w_N)$
 $= p(w_N | w_1, w_2, \dots, w_{N-1}) * p(w_1, w_2, \dots, w_{N-1})$
 $= p(w_N | w_1, w_2, \dots, w_{N-1}) * p(w_{N-1} | w_1, w_2, \dots, w_{N-2}) * p(w_1, w_2, \dots, w_{N-2})$

$$\prod_{i=1}^N p(w_i | w_1, w_2, \dots, w_{i-1})$$

- In Prosa: Die Wahrscheinlichkeit einer Sequenz ist das Produkt aus allen Wahrscheinlichkeiten der einzelnen Worte im Kontext ihrer Vorgänger

14

Sprachmodelle

$$\prod_{i=1}^N p(w_i | w_1, w_2, \dots, w_{i-1})$$

- Problem: Dieses Produkt wird fast immer 0
- Tentative Lösung: Die Markov-Annahme
$$p(w_i | w_1, w_2, \dots, w_{i-1}) = p(w_i | w_{i-M+1}, \dots, w_{i-1})$$
- Unterschiedliche Sprachmodelle = unterschiedliches M:
 - Unigram: $p(w_i | w_1, w_2, \dots, w_{i-1}) = p(w_i)$
 - Bigram: $p(w_i | w_1, w_2, \dots, w_{i-1}) = p(w_i | w_{i-1})$
 - Trigram: $p(w_i | w_1, w_2, \dots, w_{i-1}) = p(w_i | w_{i-1}, w_{i-2})$
 - ...
- Das impliziert eine *Unabhängigkeitsannahme*

15

Unabhängigkeitsannahme

- Für viele Zwecke eingeführte Annahme / Definition:
 - ein Wort ist *unabhängig* von seinem Vorgänger, bzw. nur von seinen n-1 Vorgängern (n-Gramme)
 - dann: $p(w_1 | w_2) = p(w_1)$
- Direkte Folge:
 - $p(w_2 | w_1) = \frac{p(w_1, w_2)}{p(w_1)}$
 - $p(w_1, w_2) = p(w_1) * p(w_2)$
- Das kann man direkt auf n-Gramme übertragen (Wahrscheinlichkeit einer Sequenz = Produkt aller n-Gramm- Wahrscheinlichkeiten)

16

Maximum Likelihood Estimation (MLE)

- ein Sprachmodell basiert immer auf einem Korpus, das die Sprache repräsentiert
- ein Korpus ist immer nur ein Ausschnitt einer Sprache
- eine Möglichkeit, Wahrscheinlichkeiten anzunähern, ist MLE
- die Wahrscheinlichkeit eines Wortes (n-Gramms) ist seine relative Häufigkeit im Korpus

$$p(w) = \frac{\text{count}(w)}{\text{all words}}$$

- Für N-Gramme: $p(w_1, w_2, \dots, w_n) = \frac{\text{count}(w_1, w_1, w_2, \dots, w_n)}{\text{count}(w_1, w_1, w_2, \dots, w_{n-1})}$

17

Sprachmodell-Szenario

- Gegeben sei eine Liste von Bigrammen aus einem Korpus (Wahrscheinlichkeit für jedes Wortpaar)
- Die Wahrscheinlichkeit $p(w_1, w_2, w_3, \dots, w_N)$ ist dann

$$\prod_{i=1}^N p(w_i | w_{i-1}) = \prod_{i=1}^N \frac{p(w_{i-1}, w_i)}{p(w_{i-1})}$$

$$= \prod_{i=1}^N \frac{\text{count}(w_{i-1}, w_i)}{\text{count}(w_{i-1})}$$

18

Die wahrscheinlichste Fortsetzung

- Gegeben ein Sprachmodell und eine Wortsequenz (w_1, \dots, w_i) generiere die wahrscheinlichste Fortsetzung

$$w_{i+1} = \max_w [p(w|w_1, \dots, w_i)]$$

Wahrscheinlichkeit der gegebenen Sequenz * neues n-Gramm mit Wort aus der Sprache

- Für ein n-Gramm-Modell:

$$w_{i+1} = \max_w [p(w|w_{i+1-n}, \dots, w_i) * \prod_{j=1}^i p(w_j|w_{j+1-n}, \dots, w_{j-1})]$$

verändert sich mit der n-Gramm-Länge

- Das ist trivial äquivalent mit

Das wahrscheinlichste n-Gramm beginnend mit den n-1 letzten Worten der Sequenz

$$w_{i+1} = \max_w [p(w|w_{i+1-n}, \dots, w_i)]$$

19

MLE und Datenmangel

- Trotz Markov-Annahme wird man immer ungesehene n-Gramme antreffen
- wenn ein n-Gramm in der Kette Wahrscheinlichkeit 0 hat, wird die Wahrscheinlichkeit der Sequenz 0
- das ist das grundsätzliche Problem bei maschinellem Lernen: Datenmangel (*Sparse Data Problem*)
 - Einordnungen passieren nur aufgrund gesehener Muster
 - ungesehene Muster kann man nicht adäquat einordnen

20

Smoothing

- Jede Wortfolge bekommt eine Wahrscheinlichkeit > 0
- Verschiedene grundsätzliche Taktiken:
 - Frequenzen manipulieren: Addiere eine kleine Zahl zu den Wortanzahlen (und verändere das Sprachmodell)
 - Wahrscheinlichkeiten manipulieren: schätze eine Wahrscheinlichkeit für unbekanntes, normalisiere die übrigen Wahrscheinlichkeiten (Summe =1)
 - Backing-Off: Ermittle n-Gramm-Wahrscheinlichkeit aus n-1-Grammen, ..., Unigrammen und „Zerogrammen“ (alle Worte haben die gleiche Wahrscheinlichkeit)

21

Smoothing

- Einfaches Beispiel (n-Gramme, $n=i$): Häufigkeit + x

$$p(w_1, \dots, w_i) = \frac{\text{count}(w_1, \dots, w_i) + \lambda}{N + \beta * \lambda}$$

- λ = kleine positive Zahl (z.B. 1 oder 0,5)
 - β = Anzahl der Bigramme im Test-Korpus
- Nachteil: Alle ungesehenen Sequenzen haben die gleiche Wahrscheinlichkeit

22

Text-Klassifizierung

- Gegeben:
 - eine Menge von Text-Kategorien (Spam/Nicht-Spam, Nachrichten-Themen, ...)
 - ein Text (oder mehrere)
- Aufgabe: Ordne den Text einer Kategorie zu
- Prinzip:
 - wähle verschiedene Kriterien (*features*), die bei der Texteinordnung helfen
 - trainiere ein Modell mit annotierten Texten (Features und Kategorie)
 - bestimme die wahrscheinlichste Kategorie

23

Text-Klassifizierung mit Naïve Bayes

$$c = \max_{c_i} [p(\chi|c_i) * p(c_i)]$$

- c : Kategorie, $p(c_i) =$ „A-Priori-Wahrscheinlichkeit“
- χ : sämtliche (!) features
- Auch hier Unabhängigkeitsannahme, dieses Mal zwischen den Features ($N =$ Anzahl der Features)

$$p(\chi_1, \dots, \chi_N | c) = \prod_{i=1}^N p(\chi_i | c)$$

- Features könnten auch Wörter sein
 - Bsp.: Sprachmodelle für Spam oder Nicht-Spam
 - Klasse = die Kategorie, für die der Text wahrscheinlicher ist

24

Word Sense Disambiguation

- Aufgabe: entscheide für ein mehrdeutiges Wort, welche Bedeutung es im Kontext hat
- *Bedeutung* kann unterschiedliches bedeuten
 - Wörterbuch-Eintrag
 - WordNet-Nummer
 - Übersetzung
 - ...
- Wichtig für maschinelle Übersetzung, aber ggf. auch Suchmaschinen uvm.
- zu einem gewissen Grad in Sprachmodellen implizit gegeben, aber funktioniert oft nicht...

25

(Eine) Anwendung für WSD



The screenshot shows the Google Translate web interface. At the top, the Google Translate logo is on the left, and navigation links for 'Home', 'Text und Web', 'Übersetzte Suche', and 'Tools' are on the right. Below the navigation is a header 'Text oder Internetseite übersetzen'. The main area contains the instruction 'Geben Sie Text oder eine Internetseiten-URL ein.' and a text input field containing 'Der Schimmel galoppiert.'. To the right of the input field, the translation is shown: 'Übersetzung: Deutsch » Englisch' and 'The mold galloping.'. At the bottom, there are two dropdown menus for language selection, currently set to 'Deutsch' and 'Englisch', with a right-pointing arrow between them.

26

„Schimmel“



Schimmel
Moderater SafeSearch ist aktiviert
Bilder-Suche Erweiterte Bildsuche Einstellungen

Bilder Angezeigt: Beliebige Größe Beliebiger Typ Alle Farben

Ergebnisse 1 - 21 v

Verwandte Suchvorgänge: [schimmel pferde](#) [pferd](#)



... gesundheitsgefährdender Schimmel ...
406 x 613 - 42k - jpg
www.pavatex.ch



Schimmel mit vier Scheiben ?
1600 x 1200 - 402k - jpg
www.wallstreet-online.de



Schimmel in einer Wohnung in Lehre ...
500 x 400 - 68k - jpg
www.hydro-stop.de
[Mehr von www.hydro-stop.de]



Der Schimmel zur ...
495 x 600 - 73k
www.steinan-an-der-strasse.de



Schimmel verborgen unter einer ...
500 x 350 - 72k - jpg
www.hydro-stop.de



Der Schimmel entsteht durch Essens- ...
600 x 476 - 136k - jpg
www.zeichenmacher.net



... zahmen Schimmel als Haustier.
438 x 538 - 42k - jpg
www.nasen-von-nauru.de



ver-schwommener Schimmel
400 x 323 - 20k - jpg
www.airbrush-atelier-kaiser.de



... - galicia.de/images/schimmel.jpg
640 x 480 - 59k - jpg
www.travel-tours-galicia.de



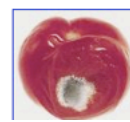
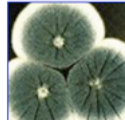
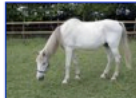
Schimmelanierung: Schimmel ...
600 x 387 - 181k - jpg
www.heimwerker-tipps.net



Problemfall Schimmel: Nach der ...
881 x 610 - 50k - jpg
www.pressebox.de



Schimmel??? Immer wieder komme neue ...
640 x 480 - 65k - jpg
www.bausachverstaendiger-neumann.de



27

„Schimmel Pferd“



Schimmel Pferd
Moderater SafeSearch ist aktiviert
Bilder-Suche Erweiterte Bildsuche Einstellungen

Bilder Angezeigt: Beliebige Größe Beliebiger Typ Alle Farben



Obwohl ich das größte Pferd in der ...
2560 x 1920 - 2414k - jpg
www.reiten-in-ruhe.de



... Horse Schimmel Pferd Pferde ...
240 x 240 - 9k - jpg
www.amazon.de



Schimmel
255 x 217 - 10k
www.wissen.de
[Mehr von www.wissen.de]



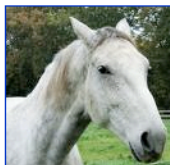
Pferdedieb owned by Schimmel Pferd
495 x 600 - 73k
www.nfs-center.de



Ein Pferd ohne Reiter bleibt ein ...
472 x 678 - 66k - jpg
www.pferdegraphik.de



PFERDE
730 x 475 - 43k - jpg
cyberhexe1.piranho.de



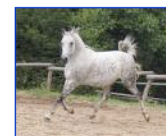
Aber sind Pferde nur das zierende ...
500 x 483 - 72k - jpg
www.buecher-wiki.de



Diese weißen Pferde werden oft mit ...
400 x 300 - 32k - jpg
www.minigestuet-westerstein.de



... sieht man den Schimmel und seine ...
120 x 171 - 4k - jpg
www.reisen-themenguide.de



... bis das Pferd vollkommen weiß ...
400 x 329 - 50k
www.hallo-pferd.de

28

„Schimmel galoppiert“



Schimmel galoppiert
Moderater SafeSearch ist aktiviert

Bilder-Suche

[Erweiterte Bildsuche](#)
[Einstellungen](#)

Bilder Angezeigt:



... einem **Schimmel galoppiert** durch ...
367 x 550 - 94k - jpg
www.zoonar.de
[[Mehr von www.zoonar.de](#)]



... **galoppiert** auf einem **Schimmel** an ...
550 x 367 - 89k - jpg
www.zoonar.de



Sisters
800 x 533 - 340k
photocompetition.hispeed.ch



Schimmel
2404 x 1775 - 2649k - jpg
fotowettbewerb.hispeed.ch



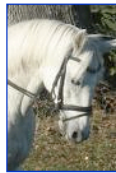
Zwei prächtige **Schimmel galoppieren** ...
170 x 246 - 14k - jpg
www.weltbild.de
[[Mehr von www.weltbild.de](#)]



Einmal in der Woche **galoppiert** er ...
357 x 671 - 30k - jpg
www.tv-abenteuer.net



... **Schimmel galoppieren**...
mehr
100 x 145 - 9k - jpg
www.weltbild.de



Fellfarbe: **Schimmel**
333 x 500 - 42k - jpg
www.joelle.de



schimmel_galopp.jpg
720 x 698 - 255k - jpg
www.lusitanos-bruckdorf.de



Frau **galoppiert** auf einem **Schimmel** ...
550 x 367 - 84k - jpg
www.zoonar.de

„Schimmel Modell“



schimmel modell
Moderater SafeSearch ist aktiviert

Bilder-Suche

[Erweiterte Bildsuche](#)
[Einstellungen](#)

Bilder Angezeigt:



Schimmel. Modell. Fortissimo 8
255 x 246 - 22k - gif
www.piano-manufaktur.de



Schimmel Modell C-120 E mit
Silent- ...
400 x 351 - 146k - jpg
www.musikhug.ch
[[Mehr von www.musikhug.ch](#)]



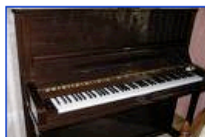
Schimmel Klavier Modell 120 J
600 x 452 - 84k - jpg
www.piano-gaebler.de



Flügel **SCHIMMEL Modell K-169** Kla
T
400 x 474 - 109k - jpg
www.musikhug.ch



SCHIMMEL, Modell Konzert 169
T
480 x 175 - 6k - jpg



Schimmel Piano Modell 128/T
aus ...
500 x 222 - 24k - jpg



... Marke **Schimmel, Model 112**
in ...
555 x 446 - 26k - jpg



Klavier **Schimmel Modell Nr**
320 x 240 - 10k - jpg
www.dkd94.com

Word Sense Disambiguation (WSD)

- Annahme für alle Algorithmen:
 - Ein Wort hat eine endliche Anzahl von klar abgegrenzten Bedeutungen
 - anhand des Kontextes ist die Bedeutung entscheidbar
- Das ist in der Realität nur bedingt so
 - Menschen haben für schwierige Fälle (viele Bedeutungen, die sich überlappen) auch nur ca. 65% Präzision
 - Die Aufgabe ist artifiziell; Menschen können Texte verstehen, ohne Worten Nummern zuzuordnen

31

Word Sense Disambiguation

- Annahmen von Yarowsky (1992,1993):
 - *One sense per discourse*
Mehrere Vorkommen des gleichen Wortes in einem Text haben die gleiche Bedeutung
 - *One sense per collocation*
Der Sinn eines Wortes kann durch syntaktische Konstellationen mit den benachbarten Wörtern bestimmt werden (\approx im wesentlichen distributionelle Hypothese)

32

WSD als Klassifizierung

- Angenommene Trainingsdaten: Ein Korpus, in dem das Wort in Frage mit seinem korrekten Sinn annotiert ist
- WSD kann so als Klassifizierungs-Aufgabe betrachtet werden
 - gegeben eine Menge von Wort-Auftreten
 - ordne jedem Wort-Auftreten den richtigen Wortsinn zu
- Dafür kann man z.B. auch den Naïve Bayes-Algorithmus verwenden

33

Naïve Bayes

- Die Features könnten im einfachsten Fall Worte im Kontext sein (Variable: Kontextgröße)
- Die einzelnen Worte im Kontext würden dann als unabhängig voneinander betrachtet (*bag of words*)
- Aber auch beliebige andere Features sind denkbar, je nach vorhandenen Trainingsdaten (POS in der Umgebung, Syntax, Wordsinne in der Umgebung...)

34

Naïve Bayes: Vorgehen

D1

Klavier... Pianist...
Flügel... spielen...
Konzert... Kinder...

D2

Lebensmittel...
Pilz... weiß...
Kühlschrank...
Brot... schwarz...

D3

galoppiert... Hof...
Pferde... spielen...
Kinder... weiß...

Wort	Pferd	Pilz	Flügel	Σ
Klavier	0	0	10	10
spielen	1	0	3	4
Pilz	0	10	0	10
weiß	4	7	4	15
Pferde	10	0	0	10
Flügel	2	0	9	11
Brot	4	9	0	13
Hof	9	0	1	10
schwarz	1	8	4	13

Naïve Bayes: Vorgehen

$$p(\chi_1, \dots, \chi_N | c) = \prod_{i=1}^N p(\chi_i | c)$$

Wort	Pferd	Pilz	Flügel	Σ
Klavier	0	0	10	10
spielen	1	0	3	4
Pilz	0	10	0	10
weiß	4	7	4	15
Pferde	10	0	0	10
Flügel	2	0	9	11
Brot	4	9	0	13
Hof	9	0	1	10
schwarz	1	8	4	13
Σ	31	34	31	96

$$p(\text{weiß} | Pf) = \frac{p(\text{weiß} \wedge Pf)}{p(Pf)}$$

$$p(\text{weiß} | Pf) = \frac{4/96}{\text{apriori}(Pf)}$$

apriori(x) ändert sich mit der Häufigkeit der Wortbedeutung

Bigramme in Python, Variante 1

```
import re

def makeBigrams(file):
    store = dict()
    with open(file) as f:
        r = [w for w in re.split('[\W_]', f.read()) if w != '']
        for bigram in [r[i] + ' ' + r[i+1] for i in range(len(r)-1)]:
            clean = [word for word in re.split('\W', bigram) if word != '']
            bi = clean[0] + ' ' + clean[1]
            if bi not in store:
                store[bi] = 1
            else:
                store[bi] = store[bi] + 1
    return store
```

37

Bigramme in Python, Variante 2

Für größere Sprachmodelle wird der Speicher überlaufen -> shelve-Objekte! (Siehe Kurs I, VL 8)

```
def makeBigrams2(file):
    store = dict()
    with open(file) as f:
        r = [w for w in re.split('[\W_]', f.read()) if w != '']
        for bigram in [r[i] + ' ' + r[i+1] for i in range(len(r)-1)]:
            clean = [word for word in re.split('[\W_]', bigram) if word != '']
            first, second = clean[0], clean[1]
            if first not in store:
                store[first] = {second:1}
            elif second in store[first]:
                store[first][second] = store[first][second] + 1
            else:
                store[first][second] = 1
    return store
```

38

Bigramme ohne Python

```
bsh$ tr -sc 'a-zA-Z' '\012' < dieTextdatei > tokens1
bsh$ tail +2 tokens1 > tokens2
bsh$ paste tokens1 tokens2 | sort | uniq -c | sort
```

- **tr** ersetzt Zeichen
 - c = „complement“
 - s = „squeeze“
- **tail** zeigt das Endstück einer Datei (ab Position x)
- **paste** positioniert 2 Texte als Spalten nebeneinander
- **uniq** löscht aufeinander folgende Dopplungen
 - c fügt deren Anzahl hinzu
- **sort** sortiert (für Optionen siehe *man sort*)

39

Project Gutenberg

(Copyright etc.)
Language: English

Character set encoding: ASCII

*** START OF THE PROJECT GUTENBERG EBOOK NOTHING TO EAT ***

Produced by Charles Aldarondo, Charles Franks
and the Online Distributed Proofreading Team

[Illustration: "PROTESTING, EXCUSING, AND SWEARING A VOW,
SHE'D NOTHING WORTH EATING TO GIVE US FOR DINNER."]

NOTHING TO EAT.
Illustrated.
NOT
By the Author of "Nothing to Wear"

"I'll nibble a little at what I have got."

40



Zusammenfassung

- Kurze Einführung zu maschinellem Lernen
- Sprachmodelle
- einfacher Naive-Bayes-Klassifizierer
- Praktisches (zur Übung und sonst)