Sprachsynthese: Part-of-Speech-Tagging

Uwe Reichel (Änderungen von F. Schiel 2016)
Institut für Phonetik und Sprachverarbeitung
Ludwig-Maximilians-Universität München
reichelu|schiel@phonetik.uni-muenchen.de

2. November 2016

Inhalt

- Motivation
- Wortarten
- POS-Probleme
- Tagger
  - Markov-Tagger
  - Transformationsbasiertes Tagging
- Evaluierung

Motivation

Was bedeutet Part-of-Speech (POS) Tagging?

- Jedes Token wird einer (syntaktischen) Wortart zugeordnet
- Reduktion der Kombinatorik (Mio Wörter $\rightarrow$ 52 STTS tags)
- Syntaktische Struktur bleibt (einigermaßen) erhalten
- Keine 'tiefe' Analyse, z.B. Numerus-Korrespondez Subject-Verb: *sie ... suchten*
- Beispiel:

Am	blauen	Himmel	ziehen	die	Wolken	.
APPRART	ADJA	NN	VVFIN	ART	NN	$.

Motivation

Beispiel mit G2P: AmBlauenHimmel.txt

Motivation

- Grundlage/Ersatz für syntaktische und semantische Analysen
- Robuster als 'tiefe' Analyse; immer eine Lösung
- **Grundlage für prosodische Modellierung**
  - POS-Sequenzen und Phrasengrenzen
  - Akzentuierbarkeit von Inhalts- vs. Funktionswörtern
- **Grundlage für morpho-syntaktische Zerlegung**

Wortarten

Z.B. Deutsch: STTS (Stuttgart-Tübingen Tagset)

Label	*Wortart*	*Beispiel*
Nomen		
NN	Substantiv	**Tisch**
NE	Eigennamen	**Hans, Hamburg**
TRUNC	Kompositions-Erstglied	**An-** *und Abreise*
Verben		
VVFIN	Finites Verb, voll	*du* **gehst**
VVIMP	Imperativ, voll	**komm***!*
VVINF	Infinitiv, voll	**gehen**
VVIZU	Infinitiv mit zu, voll	**anzukommen**
VVPP	Partizip Perfekt, voll	**gegangen**
VAFIN	Finites Verb, aux	*wir* **werden**
VAIMP	Imperativ, aux	**sei** *ruhig!*
VAINF	Infinitiv, aux	**sein, werden**
VAPP	Partizip Perfekt, aux	**gewesen**
VMFIN	Finites Verb, modal	**wollte**
VMINF	Infinitiv, modal	**wollen**
VMPP	Partizip Perfekt, modal	er hat **gekonnt**

Wortarten

Adjektive		
ADJA	Attributives Adjektiv	*das* **große** *Haus*
ADJD	Adverbiales oder prädikatives Adjektiv	*er fährt/ist* **schnell**
Pronomen		
PDS	Substituierendes Demonstrativpronomen	**dieser, jener**
PDAT	Attribuierendes Demonstrativpronomen	**dieser** *Mensch*
PIS	Substituierendes Indefinitpronomen	**keine, viele**
PIAT	Attribuierendes Indefinitpronomen	**irgendein Glas**
PIDAT	Attrib. Indef.pron. + Determiner	*ein* **wenig** *Wasser*
PPER	Irreflexives Personalpronomen	**er, dich, ihr**
PPOSS	Substituierendes Possessivpronomen	**deiner**
PPOSAT	Attribuierendes Possessivpronomen	**mein** *Buch*
PRELS	Substituierendes Relativpronomen	*der Hund,* **der**
PRELAT	Attribuierendes Relativpronomen	*der Mann,* **dessen** *Hund*
PRF	Reflexives Personalpronomen	**sich, einander**
PWS	Substituierendes Interrogativpronomen	**wer, was**
PWAT	Attribuierendes Interrogativpronomen	**welcher** *Hut*
PWAV	Adverbiales Interrog./Relativ.pron.	**warum, wo**
Adpositionen		
APPR	Präposition; Zirkumposition links	**in** *der Stadt*
APPRART	Präposition mit Artikel	**im** *Haus*
APPO	Postposition	*ihm* **zufolge**
APZR	Zirkumposition rechts	*von jetzt* **an**

Wortarten

Adverbien		
ADV	Adverb	**heute** *nicht*
PAV	Pronominaladverb	**dafür, deswegen**
Konjunktionen		
KOUI	Unterord. Konj. + 'zu' + Infinitiv	**um** *zu leben*
KOUS	Unterordnende Konjunktion mit Satz	**weil, daß**
KON	Nebenordnende Konjunktion	**und, oder**
Partikeln		
PTKZU	*zu* vor Infinitiv	**zu** *gehen*
PTKNEG	Negationspartikel	**nicht**
PTKVZ	Abgetrennter Verbzusatz	er kommt **an**
PTKANT	Antwortpartikel	**ja, danke**
PTKA	Partikel bei Adjektiv oder Adverb	**am** *schönsten*
KOKOM	Vergleichspartikel, ohne Satz	**als, wie**
Sonstige		
ART	Bestimmter oder unbestimmter Artikel	**der, eine**
CARD	Kardinalzahl	**zwei, 1984**
FM	Fremdsprachliches Material	**big**
ITJ	Interjektion	**ach**
XY	Nichtwort, Sonderzeichen	**D2XW3**
$,	Komma	**,**
$.	Satzbeendende Interpunktion	**.?!;:**
$(	Sonstige Satzzeichen; satzintern	**(-**

Wortarten

Englische Tagsets

- **Penn Tagset** (Marcus et al., 1993): 45 tags
- **C7 Tagset** (Leech et al., 1994): 146 tags

Probleme beim reinen Lexikon-Lookup

- **Out of Vocabulary-Fälle (OOV)**: POS für Wörter, die nicht im Lexikon stehen, unbekannt
- **Ambiguitäten**: Wort kann verschiedene POS-Labels tragen
  - *Sucht*: NN vs. VVFIN
  - *ab*: ADP vs. PTKVZ
  - *als*: KOKOM vs. KOUS
  - *am*: APPRART vs. PTKA
  - *Anfangs*: NN vs. ADV

Lösungen

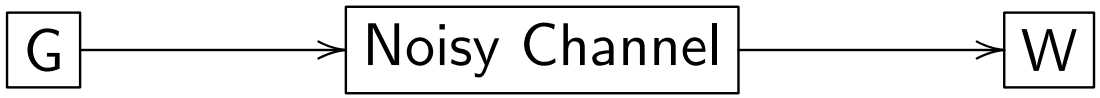
- Einbeziehen des **Wortkontexts**
- Einbeziehen POS-relevanter **Worteigenschaften**

Tagger-Überblick

- Regelbasierte Verfahren: ENGTWOL (Voutilainen, 1995)
- **Statistische Verfahren**: Jelinek (1985), Tree Tagger (Schmidt, 1995)
- **Transformationsbasierte Verfahren**: Brill (1995)

Statistische Verfahren: Noisy-Channel-Modell

Noisy-Channel-Modell



- G: unbekannter Input in einen verrauschten Kanal
- W: zugehöriger Output, der am Kanalausgang beobachtet werden kann
- **Aufgabe:** G-Rekonstruktion anhand von W
- **Bezug zum POS-Tagging:**
  - W: beobachtete Wortfolge
  - G: zugrundeliegende unbekannte POS-Sequenz
  - **Aufgabe:** Rekonstruktion der unbekannten POS-Sequenz anhand der beobachteten Wortfolge

Statistische Verfahren: Grundmodell

Mathematische Formulierung

- Schätzung der wahrscheinlichsten Tag-Sequenz $\hat{G}$, gegeben die beobachtete Wortfolge W

$$\hat{G} = \arg\max_G \Big[P(G|W) \Big]$$

- **Zur Berechnung**: Umformung unter Zuhilfename des **Satzes von Bayes**

$$\hat{G} = \arg\max_G \left[\frac{P(G)P(W|G)}{P(W)} \right]$$

- Da der Nenner konstant ist, trägt er nichts zur Maximierung bei und kann deshalb weggelassen werden:

$$\hat{G} = \arg\max_G \Big[P(G)P(W|G) \Big]$$

Transitionswahrscheinlichkeiten

$P(G)$: Transitionswahrscheinlichkeiten

- Wahrscheinlichkeit der POS-Sequenz $G = g_1 \ldots g_n$
- gemäß **Kettenregel**

$$\begin{aligned} P(g_1 \ldots g_n) &= P(g_1)P(g_2|g_1)P(g_3|g_1g_2)\ldots P(g_n|g_1 \ldots g_{n-1}) \\ &= P(g_1)\prod_{k=2}^{n} P(g_k|g_1 \ldots g_{k-1}) \end{aligned}$$

- **vereinfachende Markov-Annahme:** Vorgeschichte ist begrenzt auf Länge m (z.B. **Bigramm-Modell**, m=1)

$$\begin{aligned} P(g_1 \ldots g_n) &= P(g_1)P(g_2|g_1)P(g_3|g_2)\ldots P(g_n|g_{n-1}) \\ &= P(g_1)\prod_{k=2}^{n} P(g_k|g_{k-1}) \end{aligned}$$

Transitionswahrscheinlichkeiten

Maximum-Likelihood-Schätzung (MLE) der bedingten Wahrscheinlichkeiten

$$P(g_k|g_{k-1}) = \frac{\#(g_{k-1}, g_k)}{\#(g_{k-1})}$$

- die Häufigkeiten $\#(*)$ werden anhand eines POS-gelabelten **Trainingskorpus** ermittelt
- komplette Wahrscheinlichkeitsmasse wird für **beobachtete** Ereignisse verwendet

$\longrightarrow$ Wahrscheinlichkeit **0** für **ungesehene Ereignisse (Out-of-Vocabulary-Fälle, OOV)**

Transitionswahrscheinlichkeiten

Smoothing

- **MLE nicht adäquat**, da Trainingsdaten immer **begrenzt**
- Wahrscheinlichkeitsmasse für *OOVs* durch Verringerung der beobachteten Häufigkeiten $\#(g_{k-1}, g_k)$ um einen kleinen Betrag
- entspricht einer **Glättung** (*Smoothing*) der Wahrscheinlichkeitsfunktion
- **Verfahren:** *Absolutes Discounting*, *Good-Turing*, *Witten-Bell*, *Kneser-Ney*, ...

Emissionswahrscheinlichkeiten

$P(W|G)$: Emissionswahrscheinlichkeiten

- $P(W|G) = P(w_1 \ldots w_n|g_1 \ldots g_n)$
- **vereinfachende Annahme:** Wahrscheinlichkeit des Worts w_i hängt nur von POS-Label g_i ab: $P(W|G) \approx \prod_i P(w_i|g_i)$.
- Schätzung der $P(w_i|g_i)$ mit **MLE**:

$$P(w_i|g_i) \quad = \quad \frac{\#(g_i, w_i)}{\#(g_i)}$$

- **Smoothing**, s.o.

Statistische Verfahren: zurück zum Grundmodell

eingesetzt in $P(G)$ und $P(W|G)$:

$$\begin{aligned}\hat{G} &= \arg\max_G \Big[P(G)P(W|G)\Big] \\ &= \arg\max_G \Big[\prod_{i=1}^{n} P(g_i|g_{i-1})P(w_i|g_i)\Big]\end{aligned}$$

Wie findet man $\hat{G}$?

- Formalisierung der Transitions- und Emissionswahrscheinlichkeiten als **Hidden-Markov-Modell** zur Generierung von $\hat{G}$

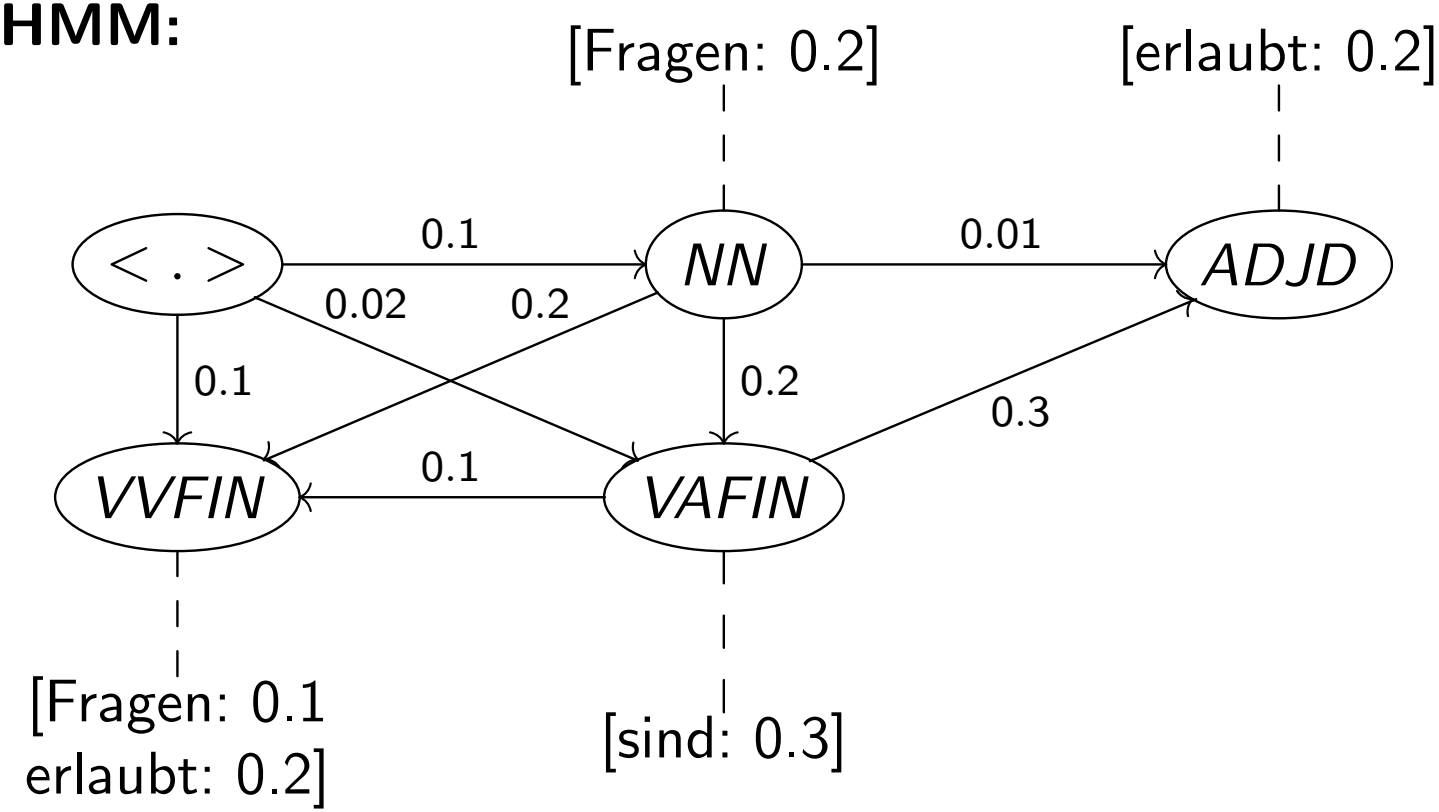
Hidden-Markov-Modelle

Hidden-Markov-Modell (HMM)

- vorstellbar als **probabilistischer Automat** (vgl. determ. endlicher Automat in Folien Textnormalisierung)
- **Zustände**
  - einer je POS-Label g_y (y: Index über Label-Inventar)
  - beinhalten **Emissionswahrscheinlichkeiten** $P(w_x|g_y)$
- **Übergänge**
  - mit **Transitionswahrscheinlichkeiten** $P(g_x|g_y)$ gewichtet
- Modell zur **Generierung** einer beobachteten Wortfolge
- durch Durchlaufen eines unbekannten (*hidden*) Pfads (einer Sequenz von Zuständen)

Hidden-Markov-Modelle

- **HMM:**



- **Transitionswahrscheinlichkeiten:** $P(ADJD|NN) = 0.01, \ldots$
- **Emissionswahrscheinlichkeiten:** $P(Fragen|NN) = 0.2$

Viterbi

Suche nach der wahrscheinlichsten POS-Sequenz $\hat{G}$

- **Brute Force:**
  - Berechnung von $P(G)P(W|G)$ für alle möglichen Sequenzen G
  - **nicht praktikabel**. z.B. Textlänge $n = 1000$, POS-Inventar: 50 Tags $\longrightarrow$ Anzahl möglicher POS-Sequenzen: 50^{1000}
- Lösung: **Viterbi-Verfahren**

Viterbi

Viterbi-Verfahren:

- **Beobachtete Wortfolge** W: *Fragen, sind, erlaubt*
- Aufbau einer **Trellis**: **Zustand-Beobachtungssequenz-Gitter**

$\downarrow$ **Zustand** g_j	**Beobachtung** $w_t \rightarrow$		
	Fragen	sind	erlaubt
ADJD			
NN			
VAFIN			
VVFIN			

- **Ziel:** Suche nach dem **wahrscheinlichsten Pfad** durch die Trellis

Viterbi

Initialisierung

$$\delta_j(1) \quad = \quad P(g_j| < . >) \cdot P(w_1|g_j)$$

$\delta_j(t)$: Trellis-Eintrag für Zustand (POS) g_j und Beobachtung w_t
i, j: Index über Zustände, t: Index über Zeit

↓ **Zustand** g_j	**Beobachtung** w_t →		
	Fragen	sind	erlaubt
ADJD	$P(\text{ADJD}\|< . >)P(\text{Fragen}\|\text{ADJD}) = 0$		
NN	$P(\text{NN}\|< . >)P(\text{Fragen}\|\text{NN}) = 0.02$		
VAFIN	0		
VVFIN	0.01		

Viterbi

Induktion

- Fülle die Trellis spaltenweise folgendermaßen auf:

$$\delta_j(t) \quad = \quad \max_i \left[\delta_i(t-1)P(g_j|g_i)P(w_t|g_j)\right]$$

- Zu jedem Knoten wird derjenige Vorgängerzustand g_i gewählt, der $\delta_j(t)$ maximiert

$\longrightarrow$ Gewinnung des **global wahrscheinlichsten Pfads** durch **lokale Maximierung** der akkumulierten Wahrscheinlichkeiten.

Viterbi

↓ **Zustand** g_j	**Beobachtung** $w_t \rightarrow$		
	Fragen	sind	erlaubt
ADJD	0		
NN	**0.02**		
VAFIN	0	$\delta_2(1)P(\mathbf{VAFIN}\|\mathbf{NN})P(\mathbf{sind}\|\mathbf{VAFIN})$=**0.012**	
VVFIN	0.01		

Viterbi

Backtracing

- Rückverfolgung des δ-maximierenden Pfads ausgehend vom maximalen δ-Eintrag in der letzten Trellis-Spalte

↓ **Zustand** g_j	**Beobachtung** w_t →		
	Fragen	sind	erlaubt
ADJD	0	0	**0.012** $P(\mathrm{ADJD}\|\mathrm{VAFIN})P(\mathbf{erlaubt}\|\mathrm{ADJD})$=**0.00072**
NN	**0.02**	0	0
VAFIN	0	**0.012**	0
VVFIN	0.01	0	0.012 $P(\mathrm{VAFIN}\|\mathrm{VVFIN})P(\mathrm{erlaubt}\|\mathrm{VVFIN})$=0.00024

Viterbi

- die dabei durchlaufenen Zustände bilden die gesuchte POS-Sequenz $\hat{G}$

$\longrightarrow$ $\hat{G}$: **NN, VAFIN, ADJD**

Lösung des "0-Problems"

- Aufmultiplizieren von Wahrscheinlichkeiten ≤ 1 führt schon bei kurzen Wortsequenzen zu Werten ≈ 0
- **Proportionalität:** $(x \cdot y) \sim (\log x + \log y)$
- Ersetzung von $\prod_i \Big[P(g_i|g_{i-1})P(w_i|g_i)\Big]$ durch $\sum_i \Big[\log P(g_i|g_{i-1}) + \log P(w_i|g_i)\Big]$

Transformationsbasiertes Tagging

Brill Tagger (Brill, 1995)

- Tagging-Regeln werden automatisch aus den Trainingsdaten gewonnen
- **Lernalgorithmus:**
  1. weise jedem Wort das wahrscheinlichste POS-Label zu
  2. **iterate until** Verbesserung $<$ Schwelle
     - wähle aus einer Menge von Transformationsregeln diejenige aus, die zum besten Tagging-Ergebnis führt
     - füge diese Regel hinten an die Liste bisher ausgewählter Regeln *trl* an
     - tagge das Corpus unter Anwendung dieser Regel neu

Transformationsbasiertes Tagging

- **Beispiel einer Transformationsregel:**
  NN $\rightarrow$ VB **if** (*vorangehendes POS-Label gleich TO*)
  expected to/TO race/NN $\longrightarrow$ *expected to/TO race/VB*
- Transformationsregeln ergeben sich durch Einsetzen aller möglichen POS-Label in **Templates** der Form

  $$POS\ \mathbf{a} \longrightarrow POS\ \mathbf{b},\ \mathbf{if}\ <Condition>$$

Transformationsbasiertes Tagging: Templates

POS **a** $\longrightarrow$ *POS* **b**, **if**

- *vorangehendes (folgendes) Wort mit POS* **z** *gelabelt ist.*
- *das zweite vorangehende (folgende) Wort mit POS* **z** *gelabelt ist.*
- *eines der zwei (drei) vorangehenden (folgenden) Wörter mit POS* **z** *gelabelt ist.*
- *vorangehendes Wort mit POS* **z** *gelabelt ist, und das folgende mit POS* **w**.
- *das vorangehende (folgende) Wort mit POS* **z** *gelabelt ist, und das zweite vorangehende (folgende) Wort mit POS* **w**.

Transformationsbasiertes Tagging

- Lernalgorithmus sehr zeitaufwendig, wenn für jedes Template zur Belegung von **a, b, z, w** alle POS-Kombinationen zugelassen werden
- **Tagging:**
  1. weise jedem Wort das wahrscheinlichste POS-Label zu
  2. **foreach** Regel *tr* **in** *trl*: tagge den Text unter Anwendung von *tr* neu

Evaluierung von POS-Taggern

- anhand eines Testkorpus
- **Gold-Standard:** Vergleich des Tagger-Outputs mit manuell gesetzten Label; Relativierung durch menschliches Inter-Tagger-Agreement < 100 %
- **Baseline:** Vergleich des Outputs mit Output eines Baseline-Taggers, beispielsweise einem Unigramm-Tagger (ohne POS-History)

Evaluierung

Kappa-Statistik

- Vergleichbarkeit von Taggern bei beliebiger Tagset-Größe
- Bei einem Tagset der Größe 1 ist die Performanz des Taggers, der nur dieses eine Label vorhersagt, 100 %.
- **Ermittlung:**

$$\kappa \quad = \quad \frac{P_{\mathrm{t}}(C) - P_{\mathrm{z}}(C)}{1 - P_{\mathrm{z}}(C)}$$

- C: Wort korrekt getaggt, $P_{\mathrm{t}}(C)$: Anteil der vom Tagger korrekt klassifizierten Wörter, $P_{\mathrm{z}}(C)$: zu erwartender Anteil zufällig korrekt klassifizierter Wörter $= \frac{1}{|\mathrm{Tagset}|}$