

# **MAUS**

Münchener AUtomatische Segmentierung

Anna Iwanow

24.06.2008

# Was? Wie? Wo?

- Problemstellung
- MAUS
  - Architektur
  - Funktionen
  - Anwendung
- Quellen

# Problemstellung

- große phonetische Variabilität der lautsprachlichen Äußerungen
- hochwertige Segmentierung kann nur von Daten kleinen Umfangs erfolgen
- in der Sprachverarbeitung wird segmentiertes Sprachsignal in großen Ausmaß benötigt
  - für TTS-Systeme
  - (innerhalb gewisser Grenzen) für ASR-Systeme
  - zur Analyse phonologischer Prozesse
  - für Lautdaueranalysen

# MAUS

- segmentiert große Mengen an Sprachmaterial automatisch
- berücksichtigt deutsche Aussprachevarianten
- auf gelesene Sprache und Spontansprache anwendbar
- ordnet lautsprachliche Kategorien den entsprechenden Abschnitten im Sprachsignal zu

# MAUS Architektur

- hybrider Ansatz aus statistischen und regelbasierten Komponenten
- Hauptkomponenten:
  - Parametrisierung des Sprachsignals
  - Graphem-Phonem-Konversion
  - Viterbi-Dekoder
  - HMM

# Datenbasis

Sprachsignal    Orthographie

Lexikon - lookup

Aussprachelexikon  
(Vollformen)

*Vorschlagstranskription*

Anwendung der Regeln

Ausspracheregeln

*Aussprachevariantengraph*

Viterbi - Alignment

42 HMM  
(phonetische Symbole)

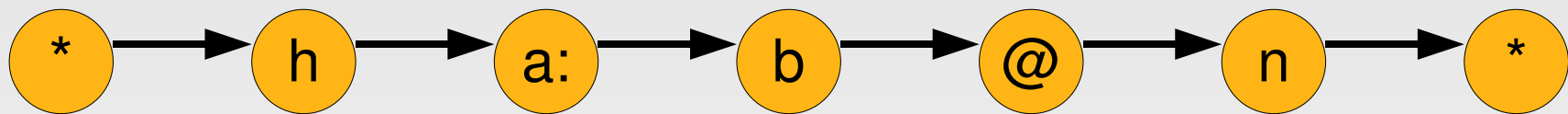
*Grobsegmentierung*

Refinement

**AUTOMATISCHE SEGMENTIERUNG**

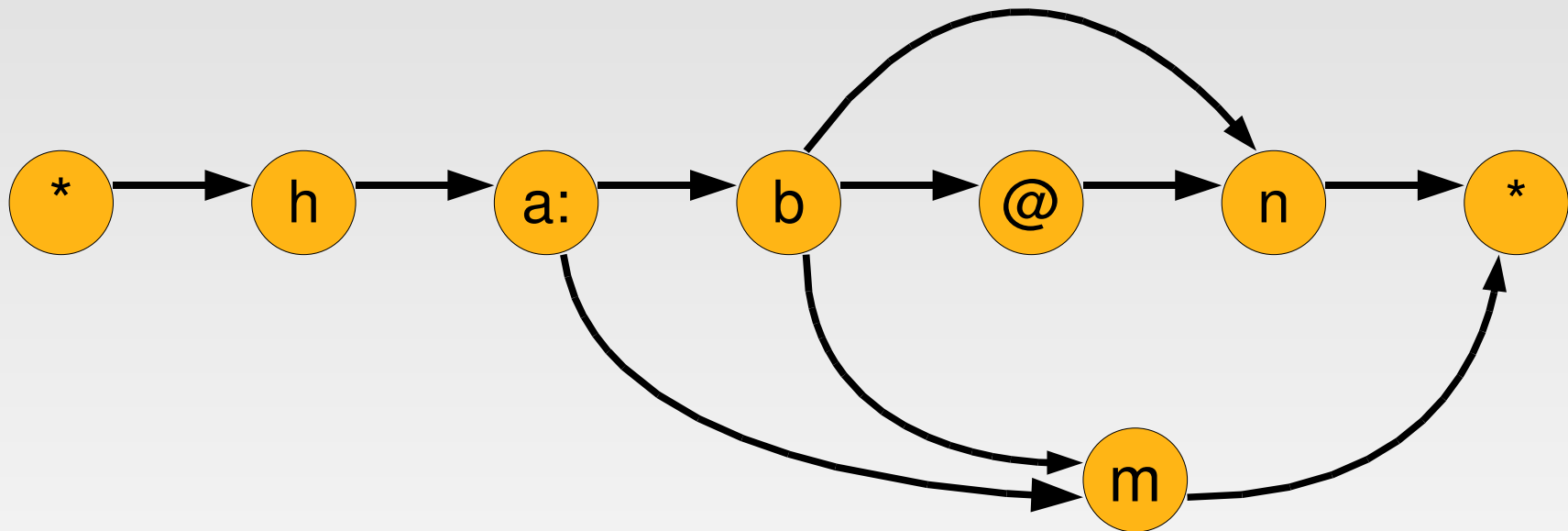
Architektur von MAUS [1]

# Varietengenerator



einfacher Graph in kanonischer Form [3]

# Variantengenerator



Graph mit Aussprachevarianten [3]

# Evaluierung

- Auswahl der entsprechenden Referenzsegmentierung
- Übereinstimmung der erzeugten Labelsegmente
- Übereinstimmung der segmentalen Grenzen

	labels	manual transcriptions	automatic transcription
Stops	p	93.8	76.4
	b	97.8	82.5
	t	92.5	80.2
	d	79.6	75.1
	k	92.1	89.2
	g	85.9	72.1
	Q	86.6	78.3
	<b>all stops</b>	<b>89.9</b>	<b>80.2</b>
Fricatives	f	99.2	99.6
	v	96.5	88.2
	s	98.5	95.1
	z	92.9	98.6
	S	99.2	94.0
	C	98.3	94.3
	j	96.4	97.5
	x	99.4	92.9
	h	92.3	71.5
	<b>all fricatives</b>	<b>98.0</b>	<b>93.6</b>
Nasals	m	98.2	97.0
	n	97.9	94.9
	N	93.4	83.5
	<b>all nasals</b>	<b>97.5</b>	<b>94.4</b>
	l	98.0	64.1
	r	96.0	99.0
<b>all consonants</b>	<b>94.8</b>	<b>88.4</b>	

Tab.1: identische Bezeichnungen der Labels bei menschlichen und automatischen Transkriptoren (%) [4]

segment boundary	manual segmentations	automatic segmentations
N – N	16	43
N – Fvd	11	34
V – L	14	36
V – Fvd	9	31
Fvl – Fvd	12	28
Fvl – Pvl	5	21
Fvl – Pvd	7	19
V – N	9	19
N – V	8	18
L – V	8	17
V -Pvd	12	19
Pvd – V	6	12
V – V	15	20
N – Pvd	11	15
Fvl – Fvl	11	13
Fvl – N	13	14
V – Fvl	7	8
N – Fvl	6	7
Fvd – V	12	12
N – Pvl	10	10
Pvd – N	9	9
V – Pvl	12	11
Pvl – Fvl	11	10
Fvl – V	7	6
Pvl – N	12	7

Pvl :: stimmloser Plosiv  
 Pvd :: stimmhafter Plosiv  
 Fvl :: stimmloser Frikativ  
 Fvd :: stimmhafter Frikativ  
 L :: Lateral  
 N :: Nasal  
 V :: Vokal

PHONDAT II: 87,9%  
 VERBMOBIL: 78,5%

Tab.2: durchschnittliche  
 Abweichung der Segmentgrenzen  
 beim manuellen und automatischen  
 Segmentieren (ms)[4]

# Maus Funktionen

- traditionelles MAUS *maus*
- iteratives MAUS *maus.iter*
- *maus.corpus*

# iteratives MAUS

- Kombinieren des traditionellen MAUS und einem Lernalgorithmus, der die Regelmenge iterativ verarbeitet

## **Initialize rule set to general phonological rules**

**for**  $k = 1 \dots 10$

- MAUS segmentation yields transcript
- computation of a-posteriori probabilities for each observed rule in the transcript
- update of the general phonological rules due to the computed probabilities
- Evaluation of the MAUS segmentations compared to reference transcript

[2]

# Maus Funktionen

- **maus** *SIGNAL* = file.wav|nis *BPF* = file.par \  
*OUT* = name.TextGrid *OUTFORMAT* = TextGrid  
  
→ *PARAM*  
→ *file.wav, file.par*

# Anwendung

- `cp /home/iwanow/unizeugs/psp/MAUS .`
- `emacs maus` → Pfad anpassen

# Maus Funktionen

- **maus** ***SIGNAL*** = file.wav|nis  
***BPF*** = file.par|***KANTSTR*** = "a: b: t s e:"  
[***OUT*** = name.TextGrid|mau]  
[***OUTFORMAT*** = mau|TextGrid]  
[***CLEAN*** = 1]  
[***CANONLY*** = no]  
[*resample*=no]  
[***STARTWORD*** = 0] [***ENDWORD*** = 999999]

# Übungen

1. Startet maus.
2. Startet maus mit den **g009acn1\_011\_ABA** - Daten aus dem Beispielordner und lasst euch das Ergebnis einmal als TextGrid und einmal als mau-file ausgeben. (Benutze statt dem wav-file \*.nis)
3. Startet maus mit den Optionen *STARTWORD* = 4 und *ENDWORD* = 8. Was passiert?
4. Was passiert, wenn ihr *KANSTR* = "a: b e: t s e:" bzw. "a: # b e: # t s e:" statt der BPF-Option anwendet?

# Quellen

- [1] <http://www.phonetik.uni-muenchen.de>
- [2] Beringer, N. und Schiel, F. 1999. Independent Automatic Segmentation of Speech by Pronunciation Modelling. Proc. of the ICPHS 1999. San Francisco. August 1999. pp. 1653-1656.
- [3] Fach, M. 2000. Automatische Segmentierung, Verwaltung und Abfrage von Korpora gesprochener Sprache. Dissertation. Universität Stuttgart.
- [4] Wesenick, M.-B. und Kipp, A. 1996. Estimating the Quality of Phonetic Transcriptions and Segmentations of Speech Signals. *In: ICSLP-1996*, 129-132.
- [5] Schiel, F. 2004. MAUS goes iterative. Proc. of the IV. International Conference on Language Resources and Evaluation, Lisbon, Portugal, pp. 1015-1018.
- [6] Wesenick, M.-B. 1996. Automatic generation of German pronunciation variants. Proceedings of the ICSLP, Philadelphia, USA, 125-128.