
Tools für Computerlinguisten

Dialogsysteme

Theresa Schmidt

Ein paar Beispiele



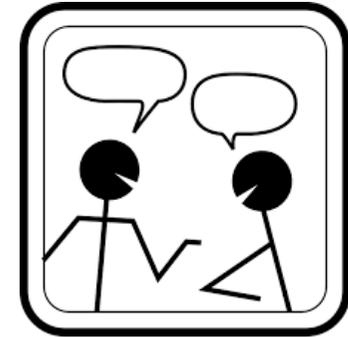
Anwendung



- Ausführend z.B. Buchungssysteme, Steuerung von Geräten, fahrender Aufzug
- Beratung / Auskunft z. B. Fahrpläne
- Interaktives Lernen z.B. Sprachen, Fluglotsentraining, Museen
- Unterhaltung z.B. alte Menschen, Spielzeug, (klinische) Psychologie



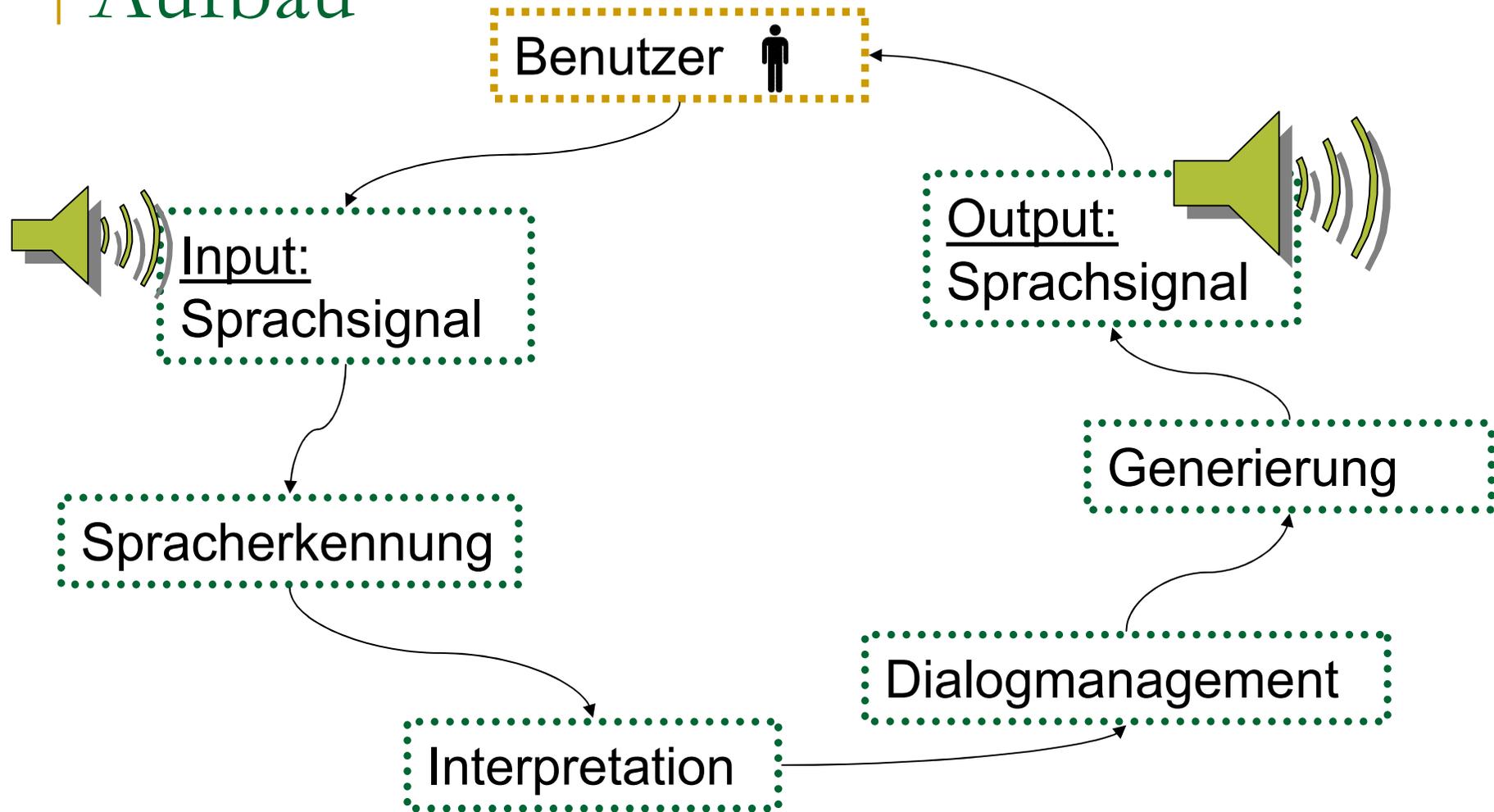
Warum?



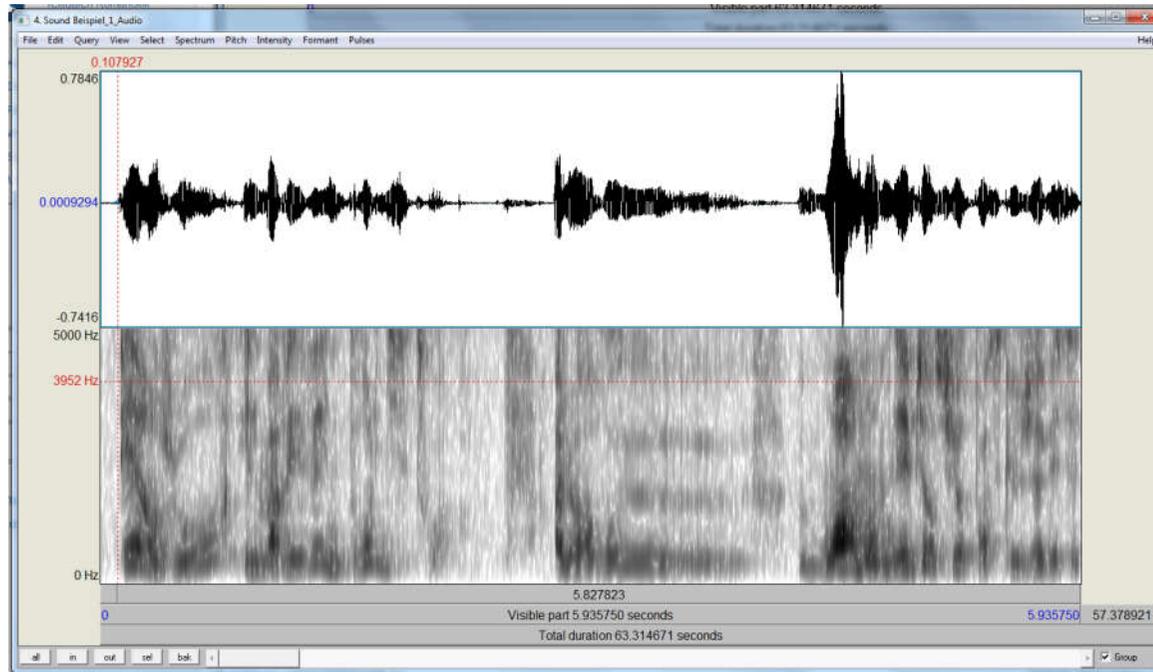
- Attraktiv: Sprache ist natürlich(st)es Kommunikationsmittel des Menschen.
- Bequem & intuitiv: Man muss „nichts“ tun oder denken.
- Persönlich: Es können Gegenfragen gestellt werden.
- Hilfreich: bei situativer oder dauerhafter Behinderung.



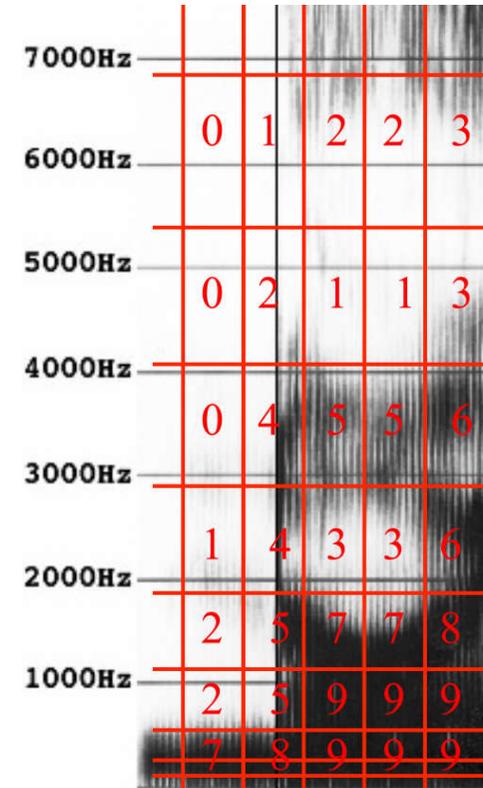
Aufbau



Automatic Speech Recognition (ASR)



Oszillogramm (oben), Spektrogramm (unten)



Merkmalsmuster

Spracherkennung

- Gesucht: die wahrscheinlichste Wortfolge W für die Beobachtungen O

$$\max_W P(W|O) = P(w_1 w_2 \dots w_n | o_1 o_2 \dots o_m)$$

-> Aussprache \neq Orthographie

- Mit Bayes:

$$\max_W P(W | O) = \max_W P(O | W) \cdot P(W)$$

-> abhängig von akustischem und Sprachmodell

Diskursinterpretation

The image displays three sequential screenshots of a mobile weather application interface, illustrating a dialogue sequence. Each screenshot shows the time, battery level, and signal strength at the top. The first screenshot (13:10, 76% battery) shows a user query: „Regnet es in Saarbrücken“ with a prompt to tap to edit. The response is: „Nein, ich denke nicht, dass es im Moment in Saarbrücken, Deutschland regnet.“ Below this is a weather card for Saarbrücken, showing „Meist bewölkt“ and a temperature of 7°. The second screenshot (13:09, 77% battery) shows a user query: „Und was ist morgen“ with a prompt to tap to edit. The response is: „Ja, morgen wird es höchstwahrscheinlich in Saarbrücken, Deutschland regnen:“ Below this is a weekly forecast card for Saarbrücken. The third screenshot (13:09, 76% battery) shows a user query: „Morgen auch“ with a prompt to tap to edit. The response is: „Du hast morgen keine Ereignisse im Kalender.“

Tag	Wetter	Max	Min
Dienstag	Wolke mit Regen	7	1
Mittwoch	Schnee	4	2
Donnerstag	Wolke mit Regen	8	2
Freitag	Wolke mit Regen	5	1
Samstag	Sonne/Wolke	3	-2

Dialogmanagement

- Das System muss sich merken, was bereits gesagt wurde.
- Der Nutzer soll möglichst frei sprechen können.

Dialogmanagement

- Das System muss sich merken, was bereits gesagt wurde.
- Der Nutzer soll möglichst frei sprechen können.

1.)

B: Ich möchte am Dienstag ab 14 Uhr mit dem ICE von München Hauptbahnhof nach Frankfurt am Main Hauptbahnhof fahren.

2.)

B: Ich möchte am Dienstag nach Frankfurt fahren.

S: Und von wo aus fahren Sie?

B: Von München Hauptbahnhof.

S: Wann genau möchten Sie nach Frankfurt (Main) Hbf fahren?

B: Ich brauche einen ICE, der nach 14 Uhr abfährt.

S: Es gibt eine Verbindung mit dem ICE um 14:05 Uhr und eine um 14:35 Uhr.

B: Die um 14:35, bitte.

S: Sie möchten die Verbindung ICE 126 von München Hbf nach Frankfurt (Main) Hbf am Dienstag, 23. Januar, um 14:35 Uhr nehmen, richtig?
Soll ich ein Ticket buchen?

Grounding

- Prinzip der Gegenseitigkeit: common ground / shared listening (das, was von beiden Parteien gleich verstanden wird)
- Grounding (Rückmeldungen & Nachfragen)
-> explizit oder implizit
- Backchanneling (kurze Äußerung, um Verständnis der vorhergehenden Äußerung anzuzeigen)

Frames und Grounding

Frame: Zug

B: Ich möchte am Dienstag mit dem Zug
nach Frankfurt fahren.
Datum, Aufgabe, Ziel

S: Und von wo aus fahren Sie?
Backchanneling

B: Von München Hauptbahnhof.
Start

S: Wann genau möchten Sie nach Frankfurt
(Main) Hbf fahren?
indirektes Grounding

B: Ich brauche einen ICE, der nach 14 Uhr
abfährt.

S: Es gibt eine Verbindung mit dem ICE um
indirektes Grounding
14:05 Uhr und eine um 14:35 Uhr.
Welche bevorzugen Sie?

B: Die um 14:35, bitte.

S: Sie möchten die Verbindung ICE 126 von
München Hbf nach Frankfurt (Main) Hbf
am Dienstag, 23. Januar, um 14:35 Uhr
nehmen, richtig? Soll ich ein Ticket
buchen?
direktes Grounding

B: Ja bitte.

Start:
München Hauptbahnhof
Ziel: Frankfurt (Main) Hbf
(nach Frankfurt -> wichtigstes Frankfurt:
Frankfurt am Main -> Bahnhof nicht näher
spezifiziert: Hauptbahnhof)
Datum: 23.01.2018
(am Dienstag -> der nächste Dienstag im
Kalender -> aktuelles Datum: 19.01.2018)
Uhrzeit:
14:35
T>14 Uhr (nach 14 Uhr) -> Datenbankabfrage
-> Rückfrage
Verkehrsmittel:
ICE
Buchung:
true -> Buchung starten, sobald alle Slots
gefüllt sind

-> Agenda: Buchung ausführen,
Rückmeldung geben

Sprecherwechsel (Turn-taking)

-> Überschneidungen

- B: Wie viele Kilometer sind es...
(Unterberechnung)
S: Die nächste rechts!
B: Wie viele Kilometer sind es noch?
S: Die Entfernung zum Ziel beträgt 15 km.



- B: Wie ist das Wetter in Minden?
S: In München scheint heute...
(Barge-in)
B: In Minden! Wie ist das Wetter in Minden?
S: Heute ist es in Minden sonnig.



Sprecherwechsel (Turn-taking)

-> Pausen (end of turn detection)

- B: Füge zur Einkaufsliste Brot, Milch ... (Pause)
(Cut-in) S: Ich habe die Einkaufsliste ergänzt.
B: ...und Eier hinzu.

- B: Spiele das Lied „Phantom of the Opera“
(Latency)
B: Hallo? Spiele „Phantom of the Opera“.
S: Soll ich „Phantom of the Opera – Titellied“ abspielen?
B: Ja.



Sprecherwechsel (Turn-taking)

-> Schnelligkeit



- Sprecherwechsel im Mensch-Mensch-Dialog: ab 200ms
- Warum? Menschen hören und sprechen inkrementell.
- Inkrementelle Sprachverarbeitung:
 - > Natürlichkeit des Dialogs wahren und möglichst keine Pausen entstehen lassen
 - > guter Umgang mit Teilsätzen erforderlich

Verarbeitungszeit verdecken

B: Ich möchte am Dienstag nach Frankfurt fahren.

S: Und von wo aus fahren Sie?

B: Von München Hauptbahnhof.

S: Wann genau möchten Sie nach Frankfurt (Main) Hbf fahren?

B: Ich brauche einen ICE, der nach 14 Uhr abfährt.

S: Es gibt eine Verbindung mit dem ICE um 14:05 Uhr und eine um 14:35 Uhr.

B: Die um 14:35, bitte.

S: Sie möchten die Verbindung ICE 126 von München Hbf nach Frankfurt (Main) Hbf am Dienstag, 23. Januar, um 14:35 Uhr nehmen, richtig? Soll ich ein Ticket buchen?

DialogOS starten

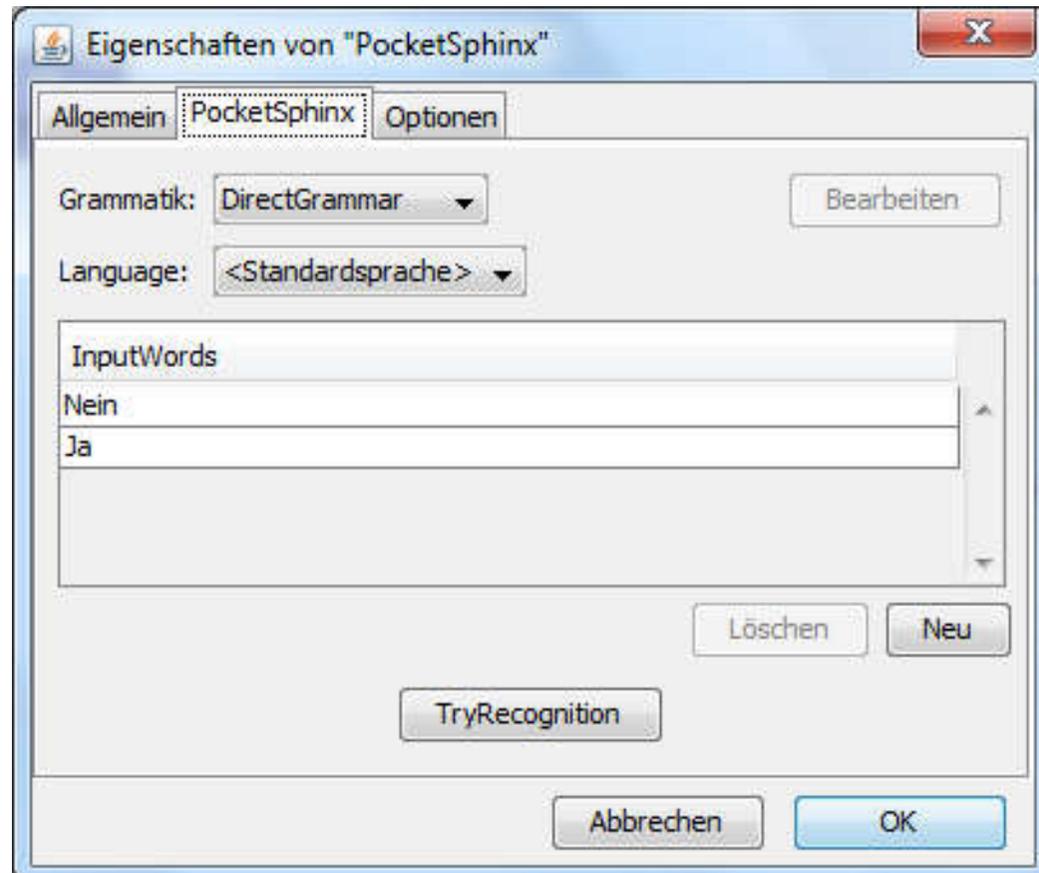
Im Speicherordner

Auf Windowsrechnern zunächst: `chcp65001` , um auf UTF-8 umzuschalten

Dann mit: `java -jar dialogos-all.jar`



Spracheingabe – Direkte Grammatik



Muster

| Alternativen
() Gruppierung
[] Option

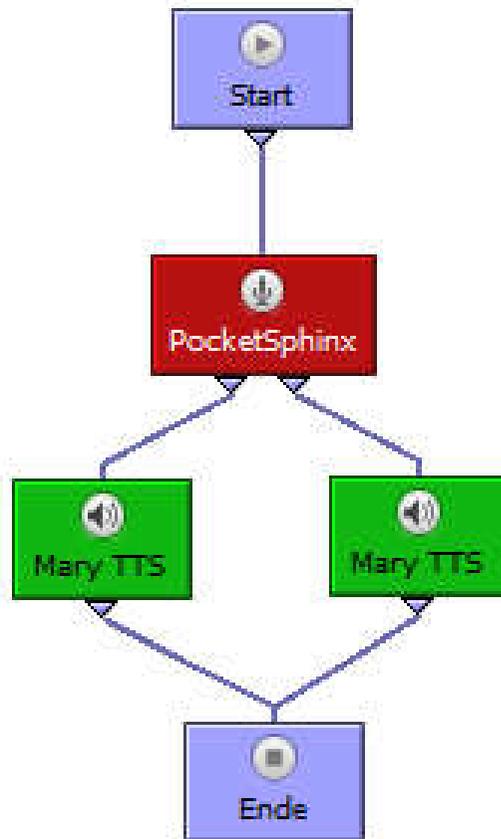
_ Platzhalter
:: und [] Listen

Regex

. Beliebiges Zeichen
+ Wiederholung
(mindestens einmal)
* Beliebige Wiederholung

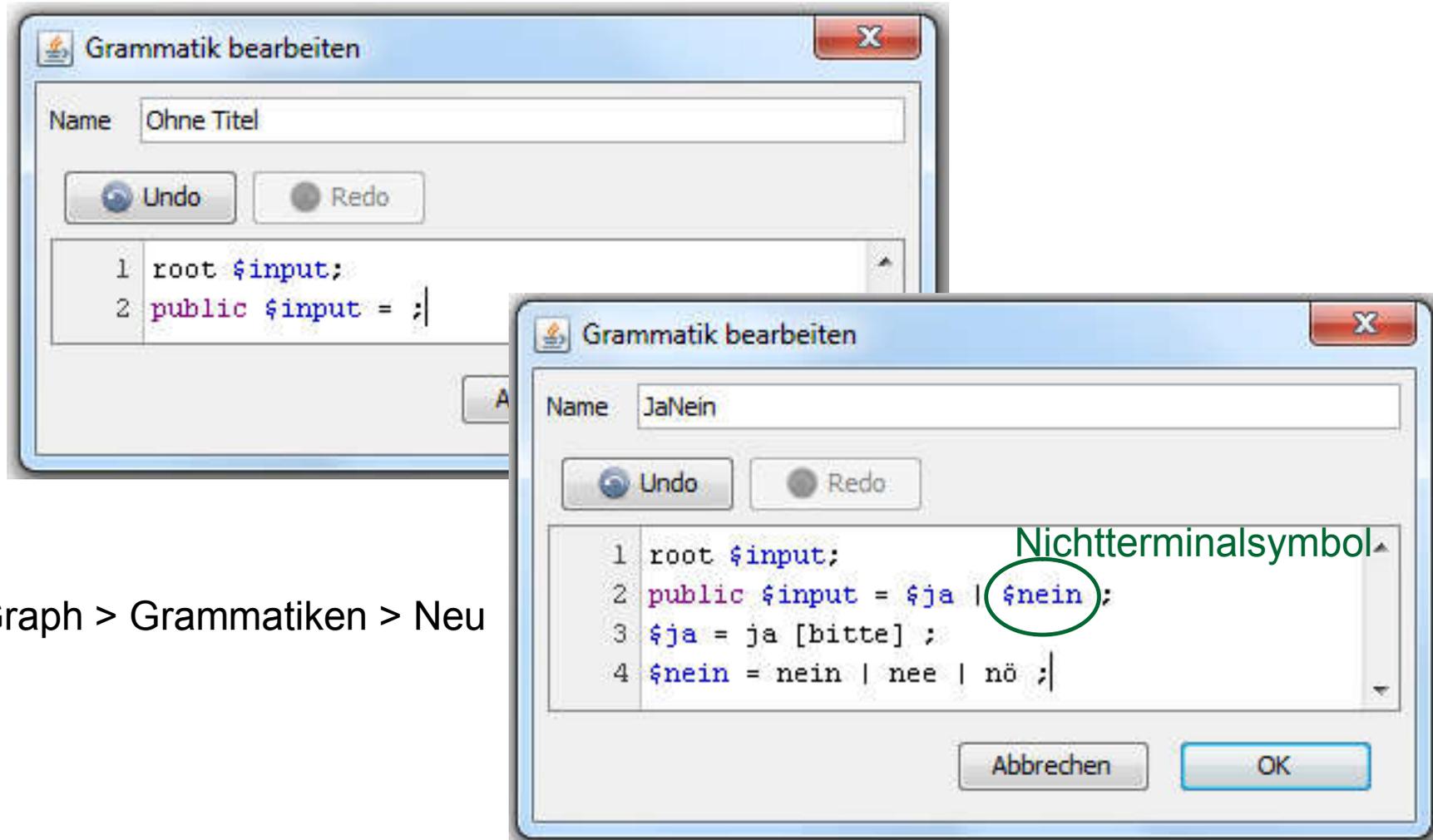
/regex/ = (Variable)

Aufgabe 1

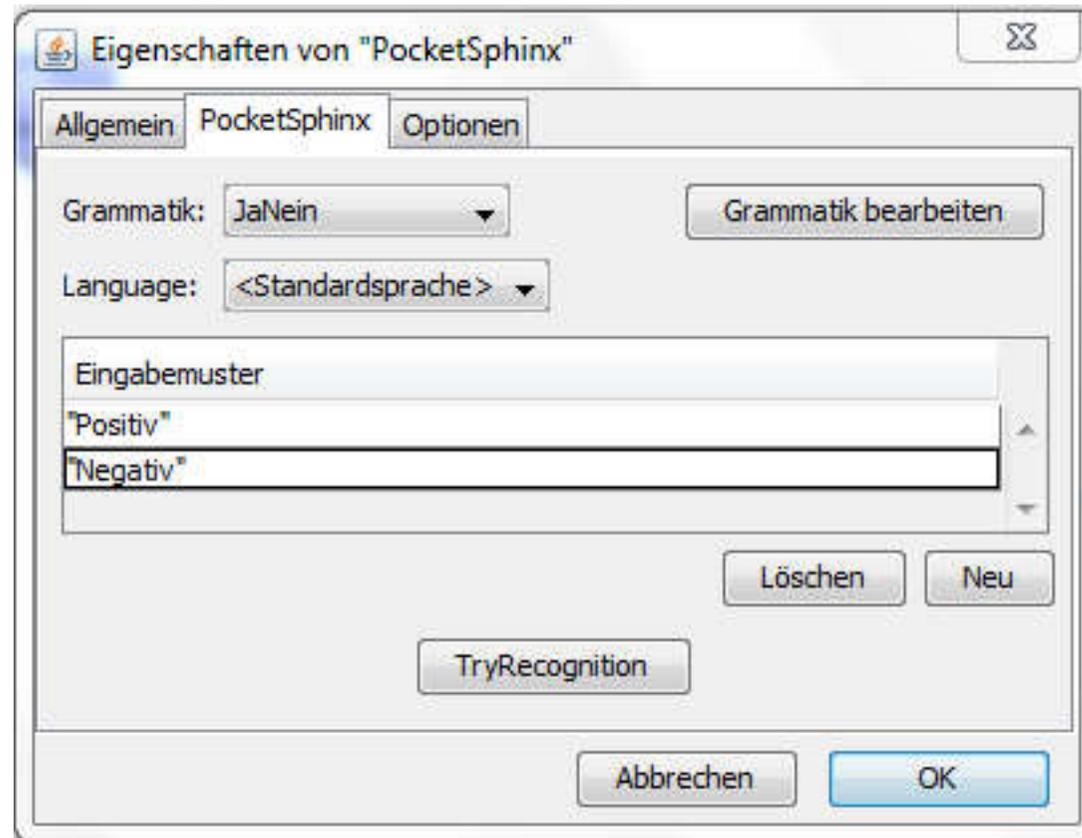


- Baut einen einfachen Automaten, der ungefähr so aussieht.
- Was kann PocketSphinx?

Grammatik – Eigene Grammatik



Spracheingabe – Eigene Grammatik



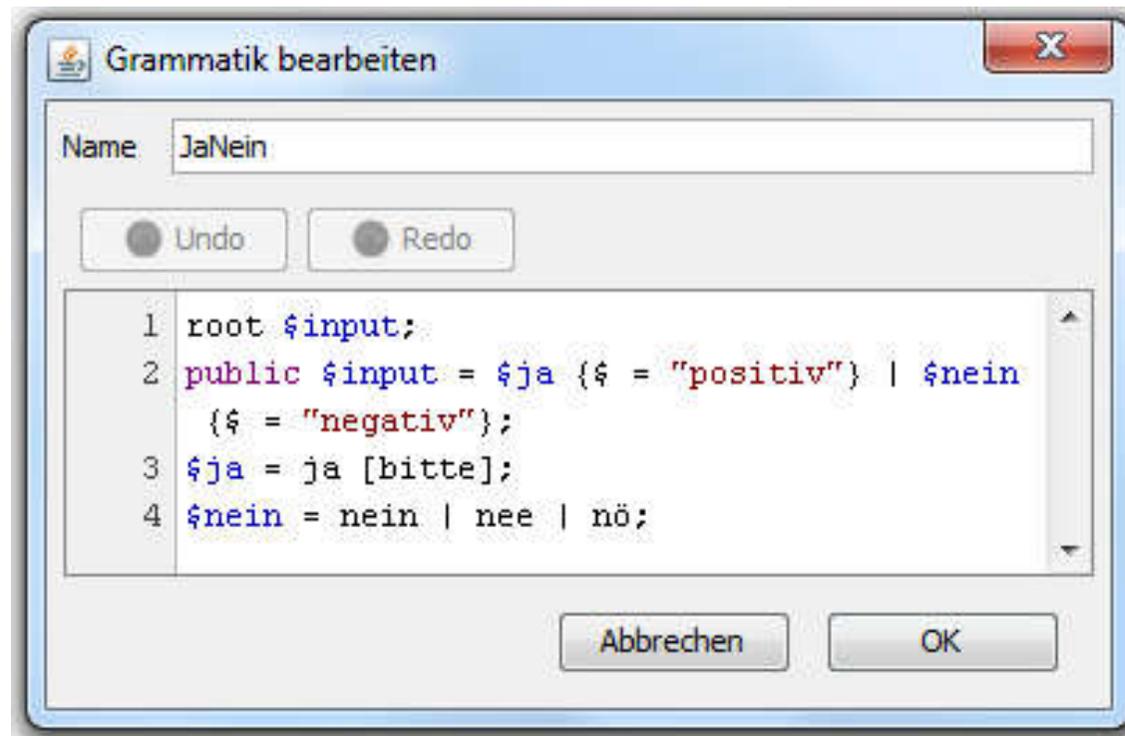
Verwendet man vordefinierte Grammatiken, müssen auch einzelne Wörter im Muster in „ “ gesetzt werden.

Tags

Tags ($\{\$ = \$xy\}$) berechnen einen Wert, der anstelle der erkannten Wortfolge zurückgegeben wird

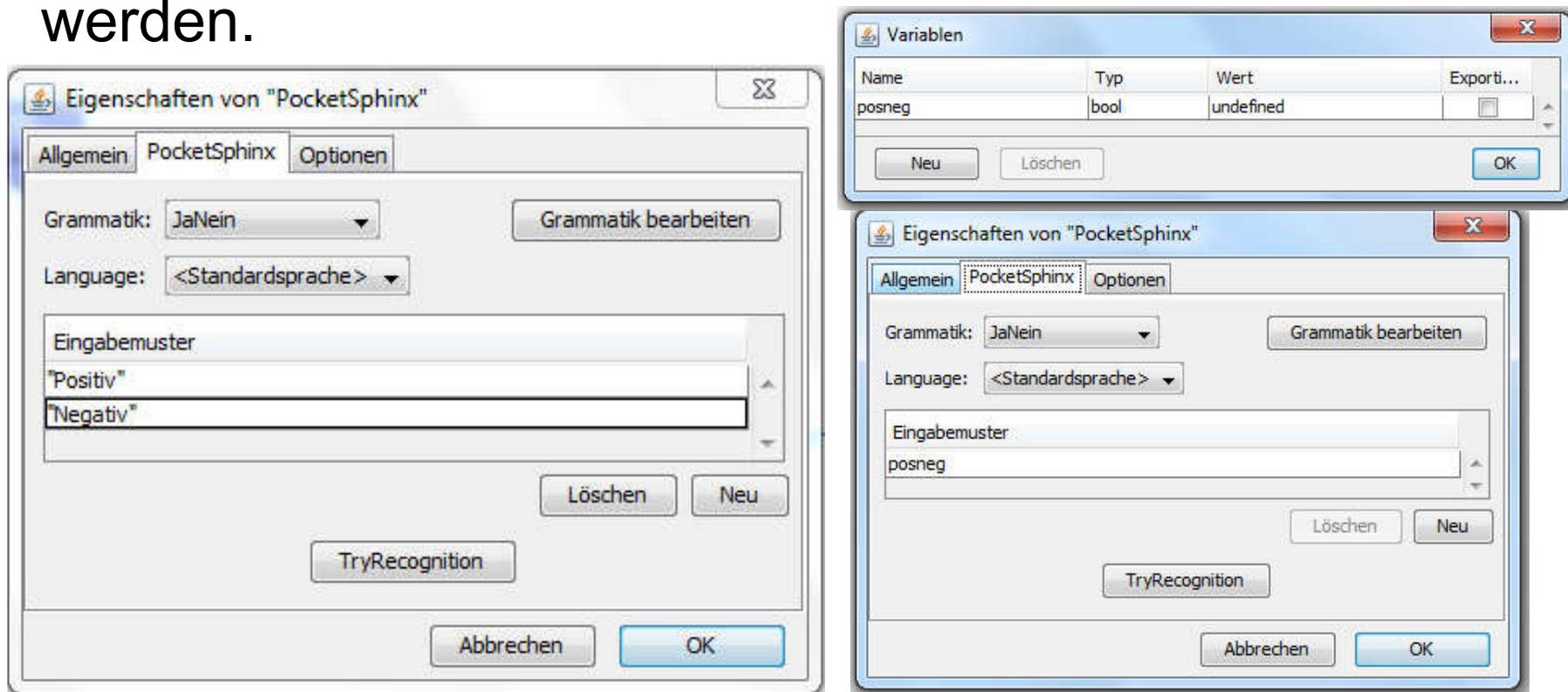
-> eingeschlossen durch geschweifte Klammern

-> aktueller Wert: \$



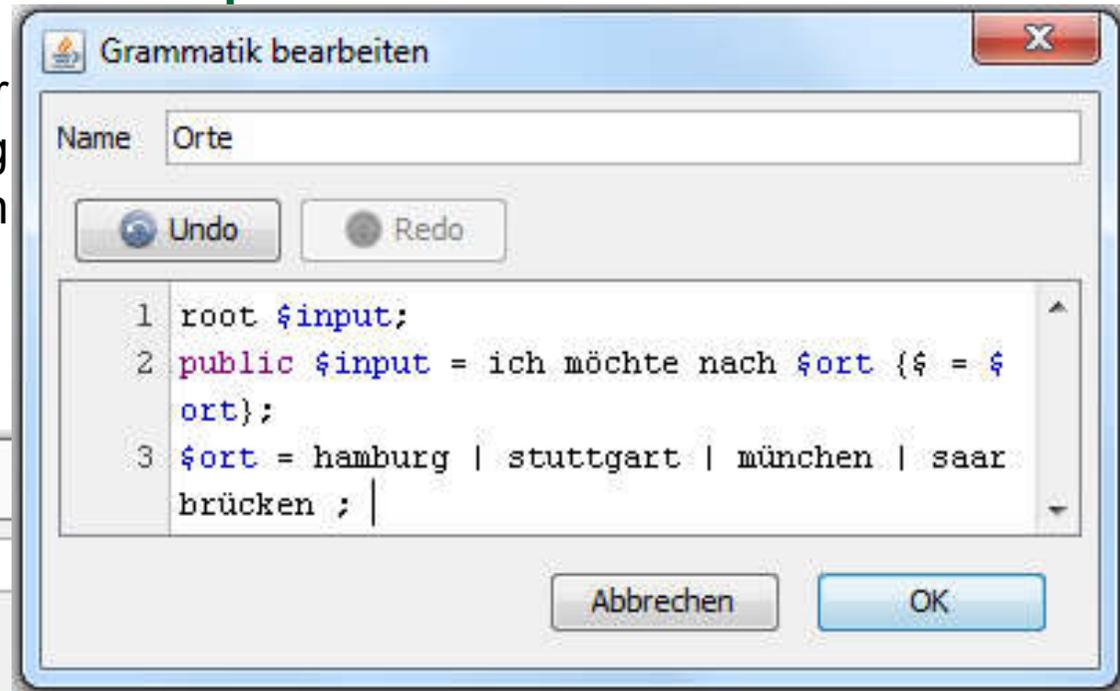
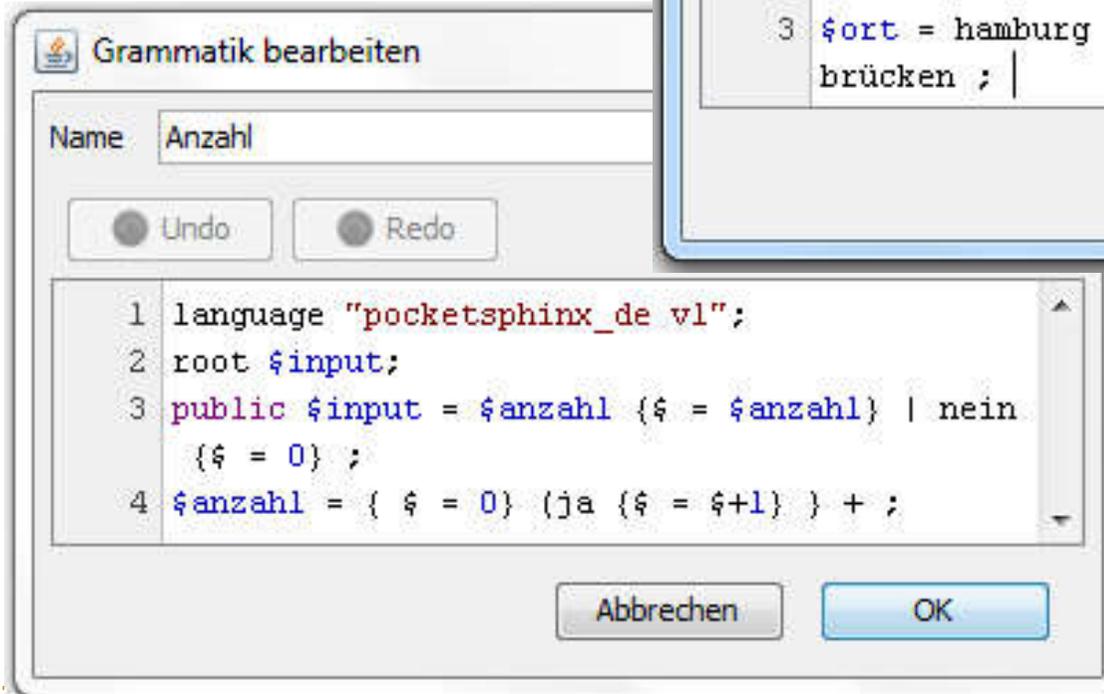
Tags

Im Erkennungsmuster des Spracherkennerknotens kann der Wert des Tags direkt verwendet oder in eine Variable (die man vorher anlegen muss) gespeichert werden.



Tags – weitere Beispiele

Relevanten Teil einer komplexen Äußerung herausfiltern

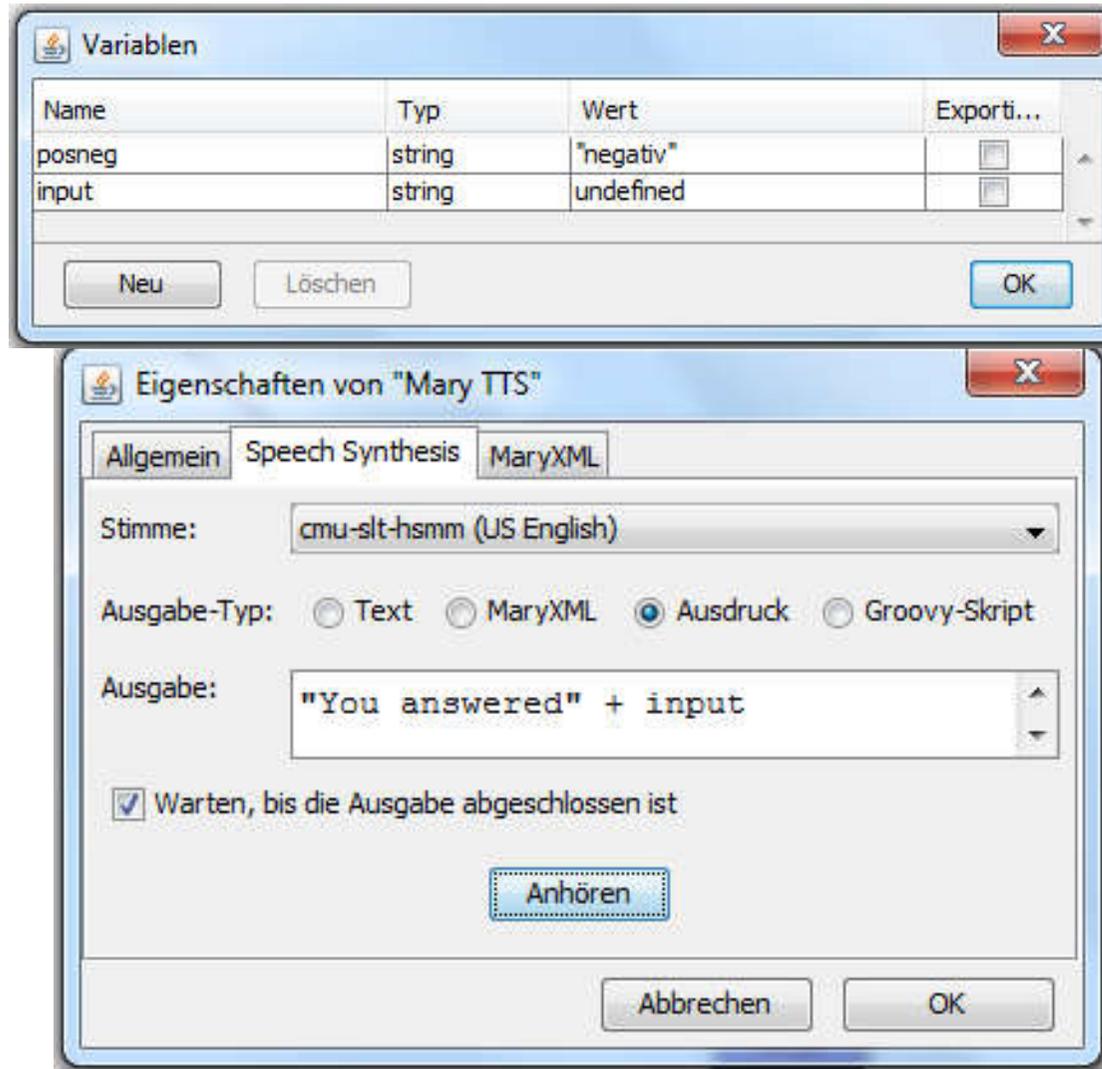


Wörter zählen

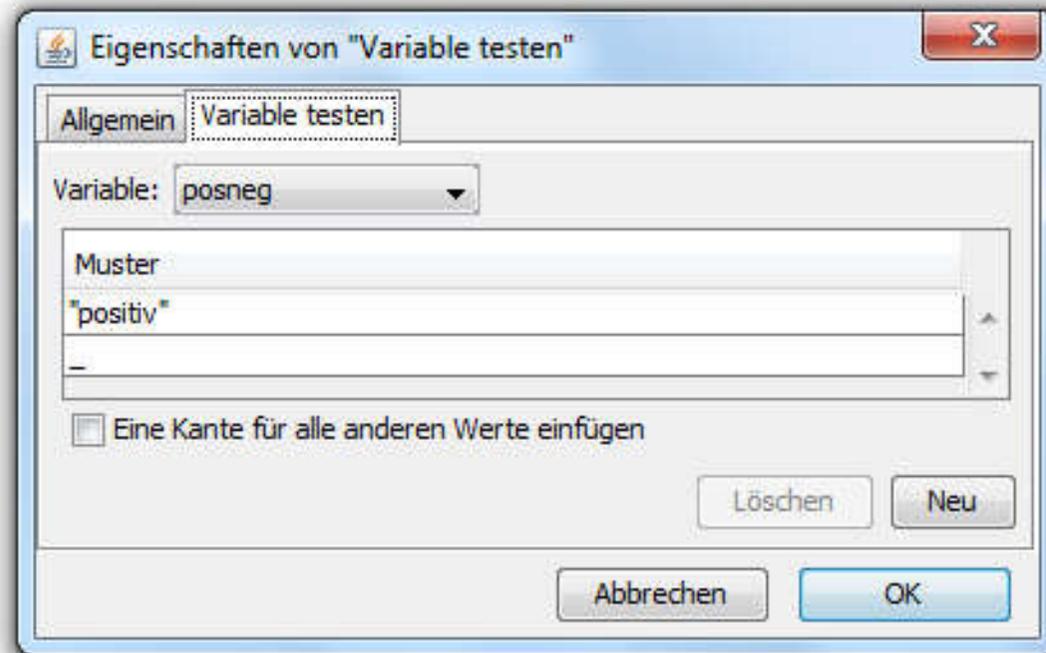
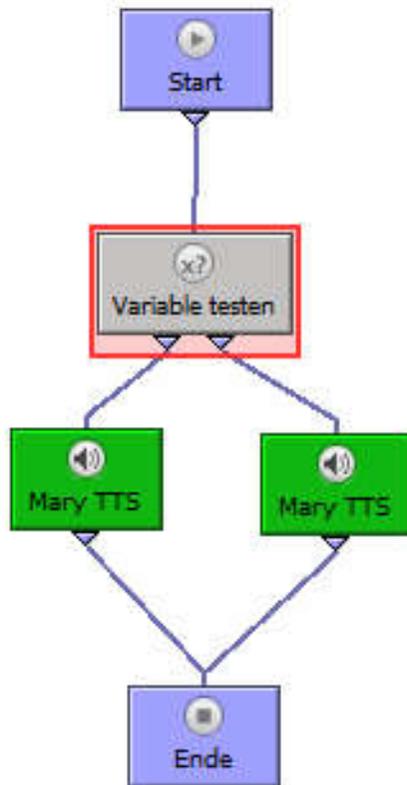
Spracheingabe - Grammatiken

- Prüft, ob eine Eingabe zulässig ist
- Liefert einen Wert (erkannte Wortfolge oder Wert eines Tags)

Graph > Variablen



Variablen testen



Externe Geräte

- Glühbirne „bauen“ (in der Kommandozeile): `java -jar demo-dialogos-client-all.jar`
- Client anlegen: Dialog > Geräte > Neu als Connector „CLT Connector (Fixed Server)“, als Server Name „localhost“ und als Port „8888“
- In einem Ausgabeknoten „on“ oder „off“ als Wert angeben



Aufgabe 2

- Tobt euch aus!
- 1. Verwendet Variablen (z. B. Punktestand eines Quizes)
- 2. Verwendet Rückfragen
- 3. Benutzt MaryXML, um die Sprachausgabe zu verbessern.
- 4. Baut etwas mit einem Lichtschalter.
- 5. Setzt Frames ein.
- 6. Verwendet Backchanneling und Grounding.

Ausdrücke - Operatoren

- für Zahlen und ! für Booleans

+ für Zahlen und Strings

-, *, /, % für Zahlen

<, >, <=, >= macht bool aus int oder real

==, != prüft Gleichheit beliebiger Typen

&&, || logisches „und“ und „oder“ (für booleans)

a ? x : y Bedingung => a muss bool sein

Zum Nachlesen

CLT Sprachtechnologie GmbH : DialogOS 1.1 Handbuch

(<http://www.coli.uni-saarland.de/courses/pd/dialogos/Handbuch.pdf>)

Smith, Ronnie W. ; Hipp, D. Richard (1994): Spoken Natural Language Dialog Systems. A Practical Approach

Nass, Clifford; Gong, Li: Ten Principles for Designing Human-Computer Dialog Systems in Dahl, Deborah (Hrsg.) (2004): Practical Spoken Dialog Systems

<http://www.debacher.de/wiki/DialogOS>

Jurafsky, Daniel; Martin, James H.(2006): Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition.

<https://dialogosose17.github.io/tutorials.html>

Demberg, Vera (2016/17): Vorlesung. Einführung in die Computerlinguistik. Foliensätze 7 und 8

Levinson, Steven C. ; Torreira, Francisco (2015): Timing in turn-taking and its implications for processing models of language

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4464110/>

Costello, Lou; Abbot, Bud: Who's on First

<https://www.youtube.com/watch?v=kTcRRaXV-fg&t=328s>

Ausschnitt aus Star Trek, Hello Computer

<https://www.youtube.com/watch?v=hShY6xZWVGE>
