Sprachsynthese: Textnormalisierung

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

20. Oktober 2022

Inhalt

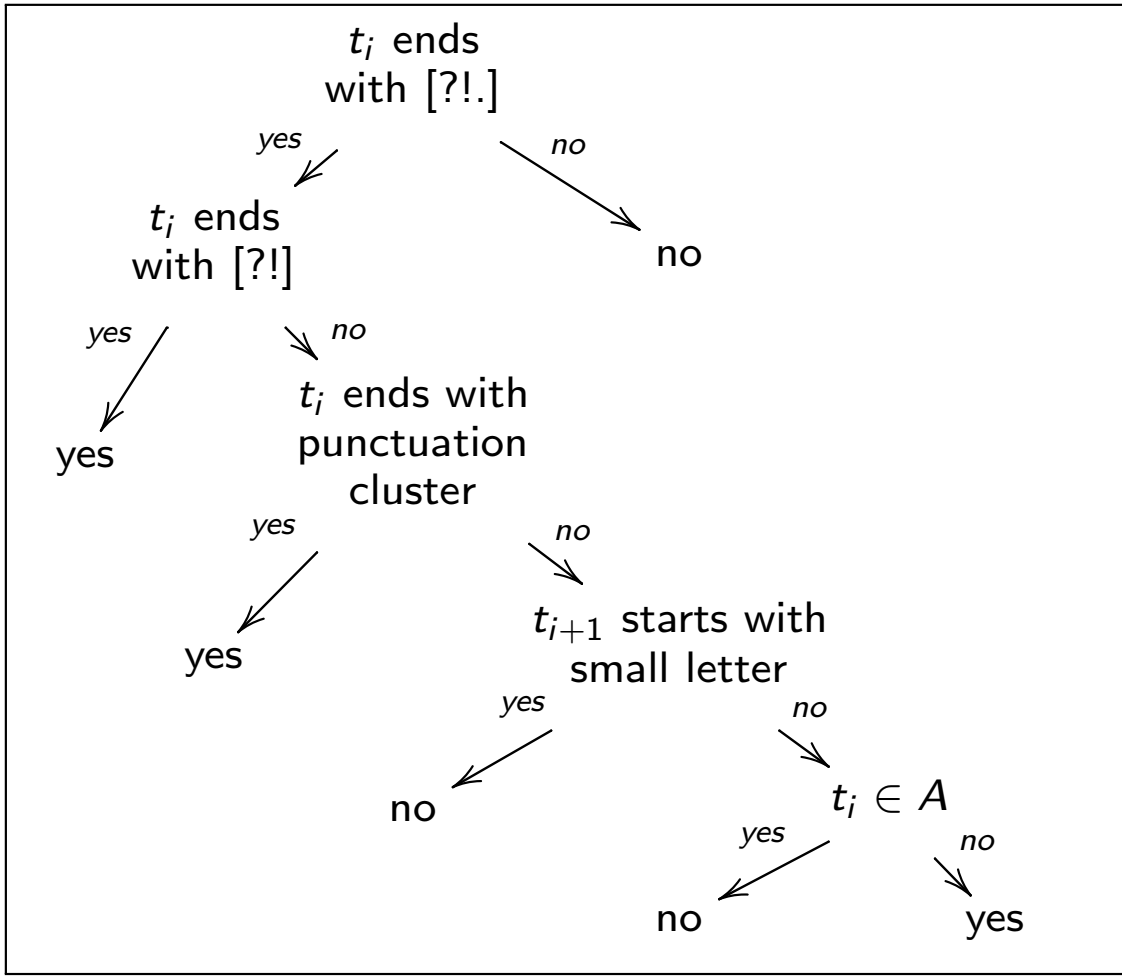
Textnormalisierung =
Vorverarbeitung des Text-Inputs in 'aussprechbare' Wortketten

- Satzsegmentierung
- Tokenisierung
- Wortnormalisierung, Finite-State-Methoden
- Ausgabe

Satzsegmentierung

- **Disambiguierung des Punkts:**
  Satzgrenze, Abkürzung, Ordnungszahl, Datum, '. . . '
- **hierbei hilfreich:**
  - Kleinschreibung nach '.' $\longrightarrow$ keine Satzgrenze
  - Abkürzungs-Mustererkennnung: Abk. enden auf '.', treten sonst nicht ohne '.' auf und
    - sind einbuchstabig oder
    - enthalten einen weiteren '.' oder
    - enthalten keinen Vokal oder
    - die Buchstabenfolge widerspricht der Silbenphonotaktik (*rcal.*)

Technik:
For each word t_i in sentence:
decide if sentence ends after t_i



Technik: Entscheidungsbaum Satzende nach Wort t_i?; A:= {Abkürzung, Titel}.

Tokenisierung

- Zerlegung eines Satzes in **Tokens**
- *Token:* durch Leerzeichen begrenzte Zeichenfolge + Abtrennung von Satzzeichen
- Vorverarbeitung:
  - z.B. Trennung inhomogener Zeichenfolgen
  - *5cm* $\longrightarrow$ *5 cm*
- Beispiel: *'Der Bolzen war 10cm lang - oder 12cm?'* $\longrightarrow$ *[Der] [Bolzen] [war] [10] [cm] [lang] [-] [oder] [12] [cm] [?]*

Wortnormalisierung

- **Wortnormalisierung:** Nicht-Standard-Wörter $\longrightarrow$ (aussprechbare) Standard-Wörter
  Beispiele: *'USA'* $\longrightarrow$ *'[u] [es] [a]'*, *'3-4'* $\longrightarrow$ *'drei bis vier'*
- **1. Expansion von Zahlen:**
  - römisch $\longrightarrow$ arabisch durch Berechnung
  - arabisch $\longrightarrow$ Wort mit Hilfe eines **Finite-State-Transducers**
  - kontextabhängig: Realisierung ziffernweise, als Bruchzahl, Datum, Jahreszahl, Uhrzeit, oder Telephonnummer
    - durchsuche Wortumgebung auf **Schlüsselphrasen**: *Nummer, Uhr, im Jahre, nach Christus, anrufen, wählen, . . .*
  - kontextabhängig: Flexion von Ordinalzahlen (*der 10. Mai, am 10. Mai*) kontextabhängig
  - Problem römische Zahlen: *Johannes X, Kapitel X, Mister X*

Wortnormalisierung: Finite-State-Methoden

Technik: Finite-State-Methoden

- Verfahren zum
  Analysieren (Parsen): Endlicher Automat (DFA)
  Transformieren: Finite-State-Transducer (FST)
- **Definitionen**
  - **Alphabet** $\sum$**:** endliche Menge von Zeichen
  - **'Wort'** w**:** Verkettung von Zeichen aus $\sum$
    $\sum^*$: Menge aller 'Wörter' über Alphabet $\sum$
  - **Sprache** L über $\sum$: Teilmenge von $\sum^*$,
    d.h. Menge von 'Wörtern'

Wortnormalisierung: Finite-State-Methoden

Technik: Endlicher Automat (DFA)

- **Entscheidungsproblem:** entscheide, ob 'Wort' $w \in L$
- **Endlicher Automat:** $\mathcal{A} = (Q, \sum, q_0, F, \Delta)$
  - Q: endliche Zustandsmenge
  - $\sum$: endliches Eingabe-Alphabet
  - $q_0 \in Q$: Startzustand
  - $F \subseteq Q$: Menge von Finalzuständen
  - $\Delta : Q \times \sum \longrightarrow Q$: Übergangsrelation; $\Delta(q_x, s) = q_y$ ("vom Zustand q_x gelangt man durch Einlesen des Zeichens s in Zustand q_y")
- Sprache $L(\mathcal{A})$: Menge aller von $\mathcal{A}$ akzeptierten 'Wörter'

Wortnormalisierung: Finite-State-Methoden

- 'Wort' w ist Element von $L(\mathcal{A})$, wenn in $\mathcal{A}$ nach Einlesen des letzten Zeichens von w ein Finalzustand erreicht werden kann
- **deterministischer** endlicher Automat DFA: Δ ist für jedes Zustand-Eingabesymbol-Paar eindeutig

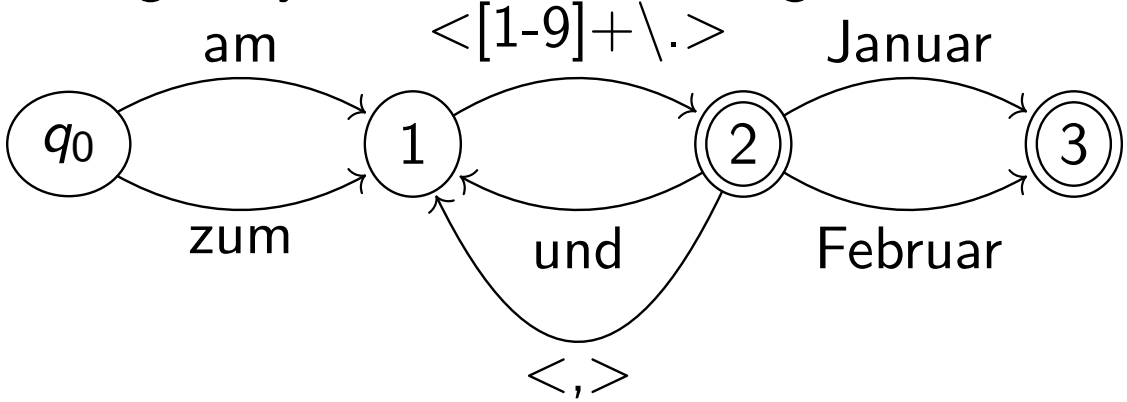


Abbildung: DFA-Ausschnitt für Sprache "Dativ". Menge von 'Wörtern' (hier: Tokensequenzen), für die Ordnungszahlen im Dativ *-ten* stehen müssen. q_0: Startzustand, 2, 3: Finalzustände

Wortnormalisierung: Finite-State-Methoden

Technik: Finite State Transducer (FST)

- Übersetzung einer Zeichenkette in eine andere
- **Finite State Transducer** $\mathcal{T} = (\sum_1, \sum_2, Q, q_0, F, \Delta)$
  - $\sum_1$: Eingabealphabet
  - $\sum_2$: Ausgabealphabet
  - Q: endliche Zustandsmenge
  - $q_0 \in Q$: Startzustand
  - $F \subseteq Q$: Finalzustände
  - $\Delta : Q \times \sum_1 \times \sum_2 \longrightarrow Q$: Übergangsrelation mit Ersetzung; $\Delta(q_x, i : o) = q_y$ ("Vom Zustand q_x gelangt man durch Einlesen des Eingabezeichens i in Zustand q_y. Dabei wird i durch das Ausgabezeichen o ersetzt.")

Wortnormalisierung: Finite-State-Methoden

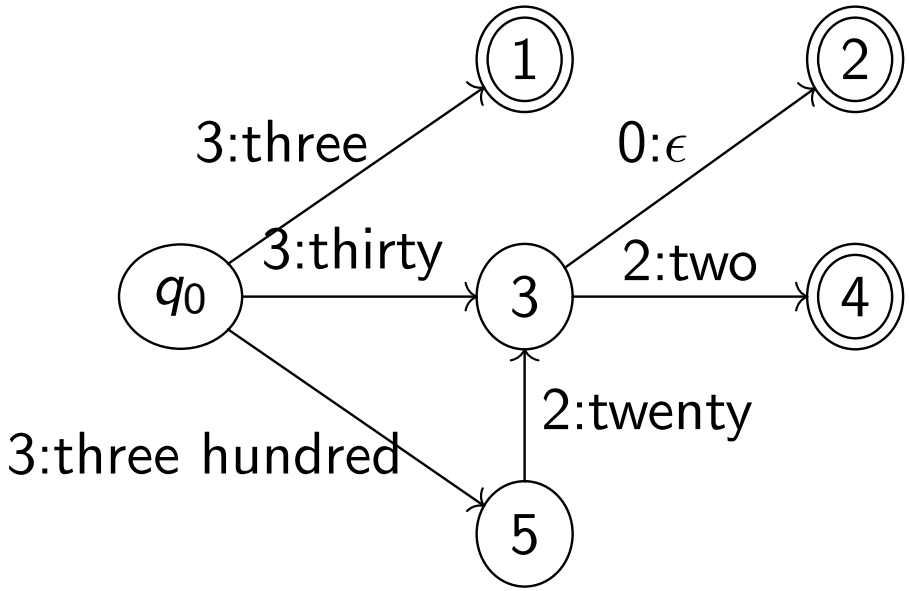


Abbildung: FST-Ausschnitt zur Umwandlung von Zahlen. Beispiel: *320* $\longrightarrow$ *three hundred twenty*.

Wortnormalisierung

- **2. Erkennung und Aussprache von Akronymen:**
  - +/− ausbuchstabiert: *UNO vs. USA*
  - Expansion auch von Wortteilen (*CD-ROM*)
- **3. Abkürzungen:** Lexikon zur Expansion nötig
- **4. Hybrid-Wörter:** *B-2-Bomber, 80er, §20(B)*, und
- **5. Multi-Wörter:** URL's, Mailadressen, und
- **6. Nicht-Buchstaben:** *4-5* $\longrightarrow$ *minus, bis*?

Technik: Heuristische Behandlung 4.-6.:

- Zerlegung an Zeichendiskontinuitäten
  (*B-2-Bomber* $\longrightarrow$ *B, –, 2, –, Bomber*)
- *Finite-State-Transducer* zur Normalisierung der einzelnen Teile

z.B. Autos.

```
<SENTENCE>
    <TOKEN>
        <STRING>z.</STRING>
        <NRM>zum</NRM>
    </TOKEN>
    <TOKEN>
        <STRING>B.</STRING>
        <NRM>Beispiel</NRM>
    </TOKEN>
    <TOKEN>
        <STRING>Autos</STRING>
        <NRM>Autos</NRM>
    </TOKEN>
    <TOKEN>
        <STRING>.</STRING>
        <NRM>.</NRM>
    </TOKEN>
</SENTENCE>
```

Beispielverarbeitung

BAS webservices
https://clarin.phonetik.uni-muenchen.de/BASWebServices/
G2P (Reichel, Kisler, 2014): *Files 2_textnormalisierung_G2P.txt, FelixBurghardt2.txt*
Optionen: lng=deu-DE, oform=tab|tcf, textnorm=yes