

Syntax natürlicher Sprachen

Tutorium

Einführung Syntax

Sarah Anna Uffelmann

20.10.2023

Organisatorisches

Kursseiten

<https://github.com/awisiorek/syntax-2324>

<https://moodle.lmu.de/course/view.php?id=29190>

Kursaufbau

- 1) Vorlesung: Theorie
- 2) Übung: praktische Übungen
- 3) Tutorium: Themenzusammenfassung, weitere Übungen, Fragen klären

Email

S.Uffelmann@campus.lmu.de

Diese Woche wichtig: INSTALLATIONEN

Anleitung siehe 01-setup.ipynb

Programm heute

Setup

- Was installieren wir?
- Wie arbeitet man mit dem Jupyter Hub Webservice?
- Wie kann man ein Git Repository klonen und updaten?
- Wie arbeitet man mit Jupyter Notebooks?

Inhalt

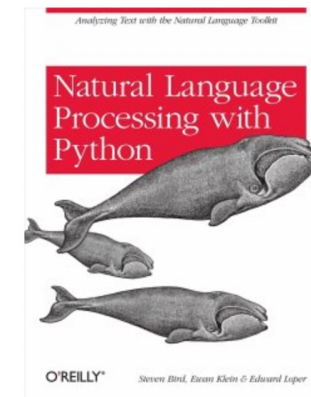
- Konstituenten
- Komplemente vs. Adjunkte
- Exkurs Python: `pop()` und `*` beim Printen von Listen
- Aufgaben

Setup

Was ist installieren wir?

(1) NLTK

- = Natural Language Toolkit
- Sammlung von open source Bibliotheken für Natural Language Processing mit Python
- entwickelt an der University of Pennsylvania seit 2001
- Website: www.nltk.org
- NLTK-Buch: www.nltk.org/book



Was ist installieren wir?

(2) Jupyter Notebooks

- webbasierte Software von Project Jupyter
- zum Entwickeln und kommentieren von Code
- eignet sich gut für Lehre und Übungen
- ➡ Ihr werdet die Klausur in einem Jupyter Notebook schreiben!
- Project Jupyter: <https://jupyter.org/>

Was ist installieren wir?

(3) spaCy

- Open Source Library für Natural Language Processing
 - Wir brauchen es nur zu Darstellungszwecken bei Abhängigkeitsgrammatiken
 - Zum Nachlesen: <https://spacy.io/>
- ➡ NLTK und Jupyter Notebooks sollte auf Eurem Rechner laufen.
Wenn es mit spaCy Probleme gibt, ist das nicht weiter schlimm.

Wie arbeitet man mit dem Jupyter Hub Webservice?

- ➡ Nur relevant, wenn bei Euch Jupyter Notebooks und/oder NLTK nicht läuft
- DHVLab-Account anlegen: Anleitung siehe 01-setup.ipynb
- Git Repository über das Terminal im DHVLab klonen
- Notebooks durch Anklicken öffnen. Die benötigten Programme und Libraries sind bereits vorinstalliert.

Wie kann man ein Git Repository klonen und updaten?

Klonen

- In der Konsole in das Verzeichnis navigieren, in das das Repository geklont werden soll
- Befehl: **git clone <URL>** (<https://github.com/awisiorek/syntax-2324>)

Updaten

- In das geklonte Repository navigieren
- Befehl: **git pull**

Alternative

- GitHub Desktop verwenden: <https://desktop.github.com/>

Wie arbeitet man mit Jupyter Notebooks?

Speichern (wenn man Git nicht verwenden will):

- Notebook im Browser öffnen (auf der Kursseite)
- Ansicht „Raw“ auswählen
- rechte Maustaste: speichern (als txt-Datei)

Öffnen mit der Konsole:

- in das Verzeichnis navigieren, wo das NB liegt
- Befehl: `jupyter notebook example.ipynb`
- Alternativ: in einem Programm öffnen, z.B. Visual Studio Code

Inhalt

Syntax

- Teilgebiet der Grammatik
- beschreibt Regeln für die Bildung wohlgeformter Sätze
- beschäftigt sich mit der Struktur von Sätzen (und Teilen von Sätzen)
- **Ziel:** Gesetzmäßigkeiten herausarbeiten, nach denen Wörter zu Wortgefügen und Sätzen zusammengefügt werden (Dürscheid)

Syntax

Verortung der Syntax innerhalb der Grammatik:

Lautstruktur -> Phonologie

Wortstruktur -> Morphologie

Satzstruktur -> **Syntax**

Konstituenten

Wir können Sätze zerlegen in Konstituenten:

- ➡ - sprachliche Einheiten, die Teile einer größeren Einheit sind
(Wörter, Phrasen, Teilsätze)
- feststellbar über Konstituententests

Kastendiagramm

Der Hund meines Nachbarn jagt die graue Katze.

Jedes Wort ist bereits selbst eine Konstituente, aber wir können sie zu größeren Einheiten (ebenfalls Konstituenten) zusammenfassen.

Im Kastendiagramm ersetzen wir nach und nach Konstituenten durch einzelne Wörter, bis wir zu einem 2-Wort-Satz kommen.

(Eine Nominalphrase und eine Verbalphrase – Phrasen werden nächste Woche behandelt!)

Kastendiagramm

- | Der | Hund | meines | Nachbarn | jagt | die | graue | Katze |
- | Der Hund | meines Nachbarn | jagt | die graue Katze |
- | Der Hund meines Nachbarn | jagt | die graue Katze |
- | Der Hund meines Nachbarn | jagt die graue Katze |
- | Der Hund meines Nachbarn jagt die graue Katze |

Kastendiagramm

Der	Hund	meines	Nachbarn	jagt	die	graue	Katze

Kastendiagramm

Der	Hund	meines	Nachbarn	jagt	die	graue	Katze
Der	Hund	dessen		jagt	die	graue	Katze

Kastendiagramm

Der	Hund	meines	Nachbarn	jagt	die	graue	Katze
Der	Hund	dessen		jagt	die	graue	Katze
Der	Elefant			jagt	die	graue	Katze

Kastendiagramm

Der	Hund	meines	Nachbarn	jagt	die	graue	Katze
Der	Hund	dessen		jagt	die	graue	Katze
Der	Elefant			jagt	die	graue	Katze
Der	Elefant			jagt	sie		

Kastendiagramm

Der	Hund	meines	Nachbarn	jagt	die	graue	Katze
Der	Hund	dessen		jagt	die	graue	Katze
Der	Elefant			jagt	die	graue	Katze
Der	Elefant			jagt	sie		
Er				jagt	sie		

Kastendiagramm

Der	Hund	meines	Nachbarn	jagt	die	graue	Katze
Der	Hund	dessen		jagt	die	graue	Katze
Der	Elefant			jagt	die	graue	Katze
Der	Elefant			jagt	sie		
Er				jagt	sie		
Er				geht			

Konstituententests

Zum **Identifizieren der Konstituenten** eines Satzes verwenden wir Konstituententests. Wir verändern dabei Teile Satzes unter Einhaltung der Wohlgeformtheit (Grammatikalität) des Satzes.

- Substitutionstest
- Permutationstest
- Eliminierungstest
- Koordinationstest

Vorsicht: Die Tests sind nicht immer verlässlich, sondern geben nur Anhaltspunkte!

Konstituententests: Substitutionstest

Austauschbarkeit im gleichen Kontext

Der Hund jagt die Katze.

Wie können wir Konstituenten des Satzes mit Hilfe des **Substitutionstests** ermitteln?

Konstituententests: Substitutionstest

Austauschbarkeit im gleichen Kontext

Der Hund jagt die Katze.

Zum Beispiel:

Er jagt die Katze.

-> „Der Hund“ ist eine Konstituente

Der Hund riecht den Braten.

-> „jagt die Katze“ ist eine Konstituente

Konstituententests: Permutationstest

Verschieben von Wortfolgen

Der Hund jagt die Katze.

Wie können wir Konstituenten des Satzes mit Hilfe des **Permutationstests** ermitteln?

Konstituententests: Permutationstest

Verschieben von Wortfolgen

Der Hund jagt die Katze.

Zum Beispiel:

Die Katze jagt der Hund.

-> „Der Hund“ und „die Katze“ sind Konstituenten

Konstituententests: Eliminierungstest

Wortfolgen weglassen

Der Hund jagt die Katze.

Wie können wir Konstituenten des Satzes mit Hilfe des **Eliminierungstest** ermitteln?

Konstituententests: Eliminierungstest

Wortfolgen weglassen

Der Hund jagt die Katze.

Zum Beispiel:

Der Hund jagt _____ .

-> „die Katze“ ist eine Konstituente

Konstituententests: Koordinationstest

Ergänzung von Konstituenten des gleichen Typs

Der Hund jagt die Katze.

Wie können wir Konstituenten des Satzes mit Hilfe des **Koordinationstest** ermitteln?

Konstituententests: Koordinationstest

Ergänzung von Konstituenten des gleichen Typs

Der Hund jagt die Katze.

Zum Beispiel:

Der Hund jagt die Katze und das Eichhörnchen.

-> „die Katze“ ist eine Konstituente

Komplement vs. Adjunkt

Komplement

- obligatorische (valenzgeforderte) Erweiterung
- Anzahl der Komplemente abhängig von der Valenz

z. B. verlangt das transitive Verb „betrachten“ ein Subjekt und ein Objekt:

Sie betrachtet das Bild.

-> „das Bild“ ist Komplement zu „betrachtet“

Komplement vs. Adjunkt

Adjunkt

- nicht-obligatorische Erweiterung
- Anzahl der Adjunkte ist beliebig

z. B. verlangt die Phrase „der Baum“ keine Erweiterung, aber wir können sie durch Adjunkte ergänzen:

Der schöne, große, grüne Baum.

-> „schöne“, „große“ und „grüne“ sind Adjunkte

Tests zur Unterscheidung von Komplementen und Adjunkten

Eliminierungstest

Er füttert die Hühner im Stall.

* „Er füttert im Stall.“

-> **nicht wohlgeformt**, also ist „die Hühner“ **Komplement**

„Er füttert die Hühner.“

-> **wohlgeformt**, also ist „im Stall“ **Adjunkt**

Tests zur Unterscheidung von Komplementen und Adjunkten

Problem beim Eliminierungstest

Er sieht die Hühner.

„Er sieht.“

-> ~~wohlgeformt, also ist „die Hühner“ Adjunkt~~ **Nein!**

-> „Er sieht“ kann zwar alleine stehen, doch trotzdem ist „sehen“ ein transitives Verb und „die Hühner“ in diesem Satz ein Komplement.

-> Auf die Tests ist nicht zu 100% Verlass

Tests zur Unterscheidung von Komplementen und Adjunkten

Adverbialsatztest

Er füttert die Hühner im Stall.

„Er fütterte die Hühner, als er im Stall war.“

-> **wohlgeformt**, also ist „im Stall“ ein **Adjunkt**

* „Er fütterte im Stall, als er die Hühner war.“

-> **nicht wohlgeformt**, also ist „die Hühner“ ein **Komplement**

Tests zur Unterscheidung von Komplementen und Adjunkten

Geschehen-Test

-

Er füttert die Hühner im Stall.

„Er fütterte die Hühner, und das geschah im Stall.“

-> **wohlgeformt**, also ist „im Stall“ ein **Adjunkt**

* „Er fütterte im Stall, und das geschah die Hühner.“

-> **nicht wohlgeformt**, also ist „die Hühner“ ein **Komplement**

Exkurs: pop()

Python-Methode für Listen

➡ entfernt (und speichert) ein Element einer Liste - per Default das letzte Element

```
tiere = ["huhn", "eichhörnchen", "ente", "hund"]  
print("Liste vor pop()", tiere)  
  
pop_tier = tiere.pop()  
print(pop_tier)  
print("Liste nach pop():", tiere)
```

Output:

```
Liste vor pop() ['huhn', 'eichhörnchen', 'ente', 'hund']  
hund  
Liste nach pop(): ['huhn', 'eichhörnchen', 'ente']
```

Exkurs: pop()

Python-Methode für Listen



um ein bestimmtes Element zu löschen, übergeben wir den Index des Elements

```
tiere = ["huhn", "eichhörnchen", "ente", "hund"]
print("Liste vor pop()", tiere)

pop_tier = tiere.pop(1)
print(pop_tier)
print("Liste nach pop():", tiere)
```

Output:

```
Liste vor pop() ['huhn', 'eichhörnchen', 'ente', 'hund']
eichhörnchen
Liste nach pop(): ['huhn', 'ente', 'hund']
```

Exkurs: * beim Printen von Listen in Python

```
tiere = ["huhn", "eichhörnchen", "ente", "hund"]  
  
print("ohne *:\t", tiere)  
print("mit *:\t", *tiere)
```

```
ohne *: ['huhn', 'eichhörnchen', 'ente', 'hund']  
mit *:  huhn eichhörnchen ente hund
```