

Syntax natürlicher Sprachen

Tutorium

Dependenzgrammatik & Dependency Parsing

Sarah Anna Uffelmann

24.11.2023

Konstituenz vs. Dependenz

Zwei verschiedene Ansätze zur syntaktischen Analyse

Konstituenz:

- Aus welchen syntaktischen Einheiten (Konstituenten) besteht ein Satz?
- Untersucht **Strukturregeln** zur Erklärung des Aufbaus eines Satzes

Dependenz:

- In welcher syntaktischen Beziehung stehen Wörter in einem Satz?
- Untersucht **Abhängigkeitsverhältnisse** zwischen Wörtern eines Satzes

Dependenzrelation

zweistellige Relation zwischen zwei Wörtern X und Y:

- Y hängt von X ab \rightarrow Y ist **Dependent** von X
- X kontrolliert / regiert Y \rightarrow X ist **Kopf** von Y

z.B. "Die" und „Blume“ in "Die Blume blüht."

„Blume“ ist **Kopf** von „Die“

„Die“ ist **Dependent** von „Blume“

Dependenzstruktur

- Der **Wurzelknoten (Root)** eines Satzes ist das **Verb**.
- Kein Wort hängt von sich selbst ab.
- In einer **Phrase** hängen alle Wörter vom Phrasenkopf ab
- Ein **Dependent** hat nur einen **Kopf**, aber ein Kopf kann mehrere Dependents haben.

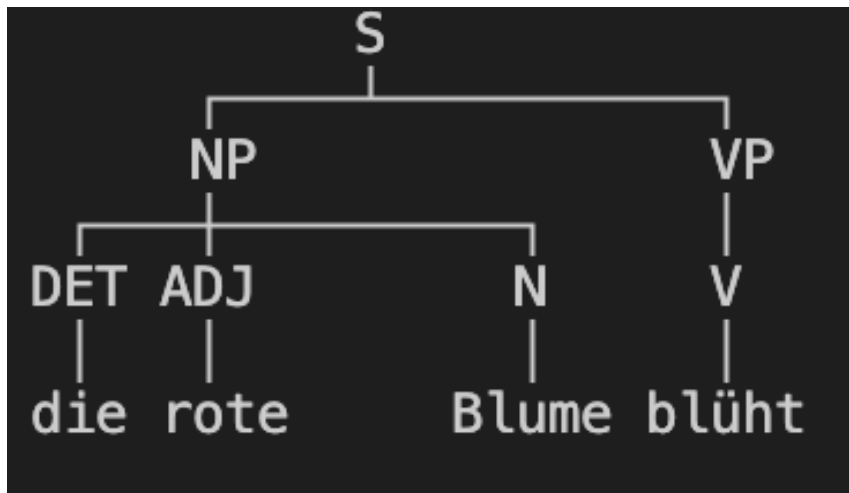
z.B. "Die rote Blume blüht."

„Blume“ hat zwei **Dependents** : „ Die“ und „rote“

„Die“ und „rote“ haben je einen **nur Kopf**: „Blume“ (= Phrasenkopf der NP)

Konstituenz vs. Dependenz

Konstituenz



S → NP VP

NP → Det ADJ N

VP → V

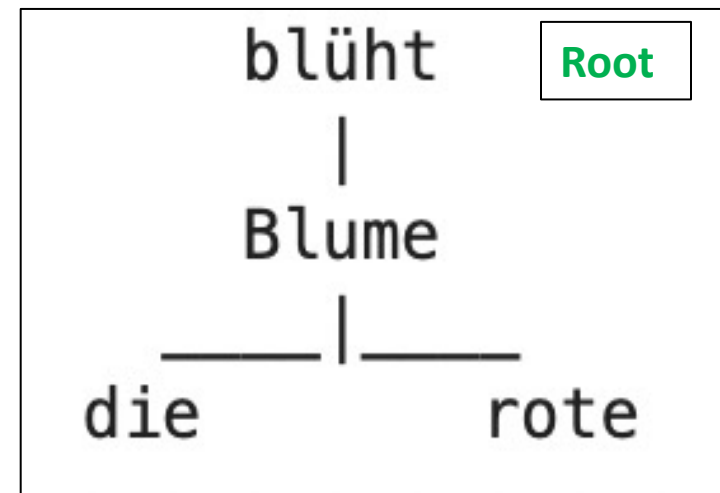
Det → "die"

N → "Blume"

ADJ → "rote"

V → "blüht"

Dependenz



Darstellung
mit NLTK

Kopf → Dependent

"blüht" → „Blume“

"Blume" → "die" | "rote"

Rektion vs. Modifikation

Rektion

- > **Kopf kann nicht ohne Dependent auftreten**
(und ein Dependent kann nie ohne Kopf auftreten)
- > Bilaterale Dependenz: Kopf und Dependent sind voneinander abhängig

„Sie betrachtet das Bild.“

* „Sie betrachtet.“ **Nicht wohlgeformt**

* „Sie das Bild.“ **Nicht wohlgeformt**

- > Dependent ist **Komplement** des Kopfes

(„das Bild“ ist Komplement von „betrachtet“: „betrachtet“ ist Kopf von „Bild“)

Rektion vs. Modifikation

Modifikation

-> Kopf kann ohne Dependent auftreten

-> Unilaterale Dependenz

„Sie betrachtet das Bild am Nachmittag.“

„Sie betrachtet das Bild.“ Wohlgeformt

-> Dependent ist Adjunkt bzw. Attribut des Kopfes

(„am Nachmittag“ ist Adjunkt von „betrachtet“: „betrachtet“ ist Kopf von „Nachmittag“.)

Arten von Dependents

obligatorischer Dependent

= Komplement

„Er füttert **die Hühner** im Stall.“

fakultativer Dependent

Komplement, aber kann je nach Kontext auch weggelassen werden:

„Er sieht **die Hühner**.“

„Er sieht.“

optionaler Dependent

= Adjunkt

„Er füttert die Hühner **im Stall**.“

Primacy of Content Words

„Er füttert die Hühner im Stall.“

Konstituenz:

- P ist Kopf einer PP
- „im Stall“ ist eine PP mit „im“ als Phrasenkopf

Dependenz:

- klassisch: „im“ ist Kopf von „Stall“
- **Primacy of Content Words:** „Stall“ ist Kopf von „im“ (im UD-Schema)
- > Nur Inhaltswörter sind Köpfe

Shift-Reduce Dependency Parsing

- Ein Satz wird Wort für Wort geparkt von einem **Buffer** (= Wortliste)
- Wörter können vom Buffer auf einen **Stack** (= Stapel) gelegt werden
- Operationen:
 - **SHIFT**: Wort vom Buffer auf den Stack legen
 - **REDUCE**: Relation zwischen den beiden obersten Elementen auf dem Stack hinzufügen und den Dependents vom Stack löschen
- Es gibt zwei REDUCE-Operationen:
 - **LEFTARC**: Kopf ist rechts vom Dependents, Bogen von rechts nach links
 - **RIGHTARC**: Kopf ist links vom Dependents, Bogen von links nach rechts


der Baum


gibt mir

Shift-Reduce Dependency Parsing

Beginn:

- Alle Wörter sind im Buffer
- Stack ist mit ROOT initialisiert
- Es sind noch keine Bögen (Dependenzrelationen) vorhanden
- Erste Operation: SHIFT

REDUCE-LEFTARC:

ist immer möglich

REDUCE-RIGHTARC:

nur dann möglich, wenn der Dependent der Relation selbst nicht Kopf einer der noch offenen Relation ist.

Diese Einschränkung verhindert, dass ein Wort zu früh vom Stack genommen wird.

Shift-Reduce Dependency Parsing

Satz: „Das Mädchen sieht das Huhn.“

Buffer: [Das, Mädchen, sieht, das, Huhn]

Stack: [ROOT]

Buffer	Stack	Operation
[Das, Mädchen, sieht, das, Huhn]	[ROOT]	SHIFT

Das Mädchen sieht das Huhn.

Shift-Reduce Dependency Parsing

Buffer	Stack	Operation
[Das, Mädchen, sieht, das, Huhn]	[ROOT]	SHIFT
[Mädchen, sieht, das, Huhn]	[ROOT, Das]	SHIFT

Das Mädchen sieht das Huhn.

Shift-Reduce Dependency Parsing

Buffer	Stack	Operation
[Das, Mädchen, sieht, das, Huhn]	[ROOT]	SHIFT
[Mädchen, sieht, das, Huhn]	[ROOT, Das]	SHIFT
[sieht, das, Huhn]	[ROOT, Das, Mädchen]	LEFTARC



Das Mädchen sieht das Huhn.

Shift-Reduce Dependency Parsing

Buffer	Stack	Operation
[Das, Mädchen, sieht, das, Huhn]	[ROOT]	SHIFT
[Mädchen, sieht, das, Huhn]	[ROOT, Das]	SHIFT
[sieht, das, Huhn]	[ROOT, Das, Mädchen]	LEFTARC
[sieht, das, Huhn]	[ROOT, Mädchen]	SHIFT



Das Mädchen sieht das Huhn.

Shift-Reduce Dependency Parsing

Buffer	Stack	Operation
[Das, Mädchen, sieht, das, Huhn]	[ROOT]	SHIFT
[Mädchen, sieht, das, Huhn]	[ROOT, Das]	SHIFT
[sieht, das, Huhn]	[ROOT, Das, Mädchen]	LEFTARC
[sieht, das, Huhn]	[ROOT, Mädchen]	SHIFT
[das, Huhn]	[ROOT, Mädchen, sieht]	LEFTARC



Das Mädchen sieht das Huhn.

Shift-Reduce Dependency Parsing

Buffer	Stack	Operation
[Das, Mädchen, sieht, das, Huhn]	[ROOT]	SHIFT
[Mädchen, sieht, das, Huhn]	[ROOT, Das]	SHIFT
[sieht, das, Huhn]	[ROOT, Das, Mädchen]	LEFTARC
[sieht, das, Huhn]	[ROOT, Mädchen]	SHIFT
[das, Huhn]	[ROOT, Mädchen, sieht]	LEFTARC
[das, Huhn]	[ROOT, sieht]	SHIFT



Das Mädchen sieht das Huhn.

Shift-Reduce Dependency Parsing

Buffer	Stack	Operation
[Das, Mädchen, sieht, das, Huhn]	[ROOT]	SHIFT
[Mädchen, sieht, das, Huhn]	[ROOT, Das]	SHIFT
[sieht, das, Huhn]	[ROOT, Das, Mädchen]	LEFTARC
[sieht, das, Huhn]	[ROOT, Mädchen]	SHIFT
[das, Huhn]	[ROOT, Mädchen, sieht]	LEFTARC
[das, Huhn]	[ROOT, sieht]	SHIFT
[Huhn]	[ROOT, sieht, das]	SHIFT



Das Mädchen sieht das Huhn.

Shift-Reduce Dependency Parsing

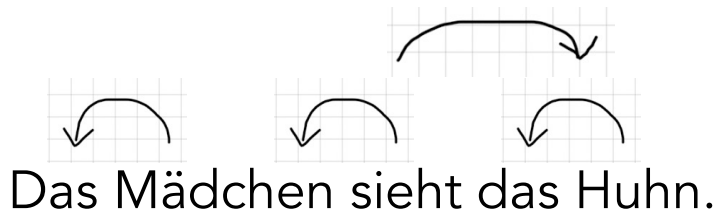
Buffer	Stack	Operation
[Das, Mädchen, sieht, das, Huhn]	[ROOT]	SHIFT
[Mädchen, sieht, das, Huhn]	[ROOT, Das]	SHIFT
[sieht, das, Huhn]	[ROOT, Das, Mädchen]	LEFTARC
[sieht, das, Huhn]	[ROOT, Mädchen]	SHIFT
[das, Huhn]	[ROOT, Mädchen, sieht]	LEFTARC
[das, Huhn]	[ROOT, sieht]	SHIFT
[Huhn]	[ROOT, sieht, das]	SHIFT
[]	[ROOT, sieht, das, Huhn]	LEFTARC



Das Mädchen sieht das Huhn.

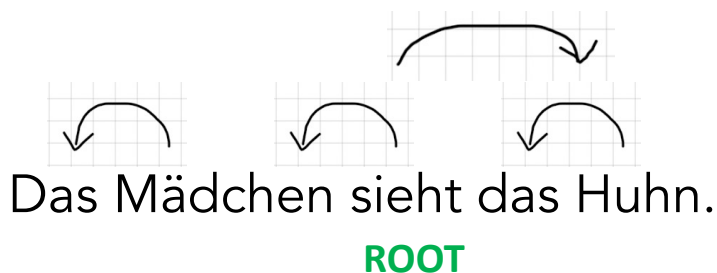
Shift-Reduce Dependency Parsing

Buffer	Stack	Operation
[Das, Mädchen, sieht, das, Huhn]	[ROOT]	SHIFT
[Mädchen, sieht, das, Huhn]	[ROOT, Das]	SHIFT
[sieht, das, Huhn]	[ROOT, Das, Mädchen]	LEFTARC
[sieht, das, Huhn]	[ROOT, Mädchen]	SHIFT
[das, Huhn]	[ROOT, Mädchen, sieht]	LEFTARC
[das, Huhn]	[ROOT, sieht]	SHIFT
[Huhn]	[ROOT, sieht, das]	SHIFT
[]	[ROOT, sieht, das, Huhn]	LEFTARC
[]	[ROOT, sieht, Huhn]	RIGHTARC



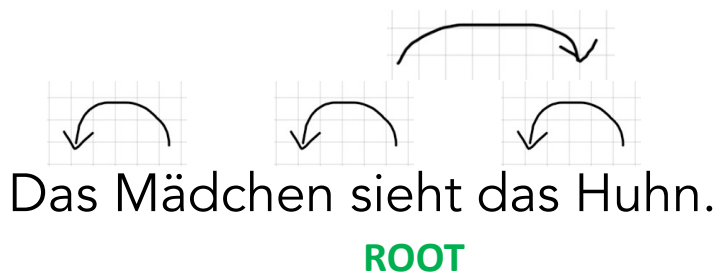
Shift-Reduce Dependency Parsing

Buffer	Stack	Operation
[Das, Mädchen, sieht, das, Huhn]	[ROOT]	SHIFT
[Mädchen, sieht, das, Huhn]	[ROOT, Das]	SHIFT
[sieht, das, Huhn]	[ROOT, Das, Mädchen]	LEFTARC
[sieht, das, Huhn]	[ROOT, Mädchen]	SHIFT
[das, Huhn]	[ROOT, Mädchen, sieht]	LEFTARC
[das, Huhn]	[ROOT, sieht]	SHIFT
[Huhn]	[ROOT, sieht, das]	SHIFT
[]	[ROOT, sieht, das, Huhn]	LEFTARC
[]	[ROOT, sieht, Huhn]	RIGHTARC
[]	[ROOT, sieht]	RIGHTARC



Shift-Reduce Dependency Parsing

Buffer	Stack	Operation
[Das, Mädchen, sieht, das, Huhn]	[ROOT]	SHIFT
[Mädchen, sieht, das, Huhn]	[ROOT, Das]	SHIFT
[sieht, das, Huhn]	[ROOT, Das, Mädchen]	LEFTARC
[sieht, das, Huhn]	[ROOT, Mädchen]	SHIFT
[das, Huhn]	[ROOT, Mädchen, sieht]	LEFTARC
[das, Huhn]	[ROOT, sieht]	SHIFT
[Huhn]	[ROOT, sieht, das]	SHIFT
[]	[ROOT, sieht, das, Huhn]	LEFTARC
[]	[ROOT, sieht, Huhn]	RIGHTARC
[]	[ROOT, sieht]	RIGHTARC
[]	[ROOT]	DONE



Shift-Reduce Dependency Parsing

Satz: „Kauf Tickets nach München.“

Buffer: [Kauf, Tickets, nach, München]

Stack: [ROOT]

Buffer	Stack	Operation
[Kauf, Tickets, nach, München]	[ROOT]	SHIFT

Kauf Tickets nach München.

Shift-Reduce Dependency Parsing

<u>Buffer</u>	<u>Stack</u>	<u>Operation</u>
[Kauf, Tickets, nach, München]	[ROOT]	SHIFT
[Tickets, nach, München]	[ROOT, Kauf]	

Welche Operation ist jetzt die richtige?

Kauf Tickets nach München.

Shift-Reduce Dependency Parsing

Buffer	Stack	Operation
[Kauf, Tickets, nach, München]	[ROOT]	SHIFT
[Tickets, nach, München]	[ROOT, Kauf]	

Möglichkeiten:

- Rightarc: „Kauf“ wird als Root markiert und vom Stack genommen.
- Shift: Wir legen das nächste Wort auf den Stack.

Kauf Tickets nach München.

Shift-Reduce Dependency Parsing

Buffer	Stack	Operation
[Kauf, Tickets, nach, München]	[ROOT]	SHIFT
[Tickets, nach, München]	[ROOT, Kauf]	

Möglichkeiten:

- Rightarc: „Kauf“ wird als Root markiert und vom Stack genommen.
- Shift: Wir legen das nächste Wort auf den Stack.

Richtig ist **SHIFT**, da das Parsen des restlichen Satzes nicht mehr möglich wäre, wenn wir „Kaufe“ vom Stack nehmen.

Kauf Tickets nach München.

Shift-Reduce Dependency Parsing

Buffer	Stack	Operation
[Kauf, Tickets, nach, München]	[ROOT]	SHIFT
[Tickets, nach, München]	[ROOT, Kauf]	SHIFT

Kauf Tickets nach München.

Buffer	Stack	Operation
[Kauf, Tickets, nach, München]	[ROOT]	SHIFT
[Tickets, nach, München]	[ROOT, Kauf]	SHIFT
[nach, München]	[ROOT, Kauf, Tickets]	

Jetzt wieder SHIFT, da mit RIGHTARC „Tickets“ zu früh vom Stack genommen werden würde. Das Parsen von „nach München“ wäre dann nicht mehr möglich.

Kauf Tickets nach München.

Shift-Reduce Dependency Parsing

Buffer	Stack	Operation
[Kauf, Tickets, nach, München]	[ROOT]	SHIFT
[Tickets, nach, München]	[ROOT, Kauf]	SHIFT
[nach, München]	[ROOT, Kauf, Tickets]	SHIFT

Kauf Tickets nach München.

Shift-Reduce Dependency Parsing

Buffer	Stack	Operation
[Kauf, Tickets, nach, München]	[ROOT]	SHIFT
[Tickets, nach, München]	[ROOT, Kauf]	SHIFT
[nach, München]	[ROOT, Kauf, Tickets]	SHIFT
[München]	[ROOT, Kauf, Tickets, nach]	SHIFT

Kauf Tickets nach München.

Shift-Reduce Dependency Parsing

Buffer	Stack	Operation
[Kauf, Tickets, nach, München]	[ROOT]	SHIFT
[Tickets, nach, München]	[ROOT, Kauf]	SHIFT
[nach, München]	[ROOT, Kauf, Tickets]	SHIFT
[]	[ROOT, Kauf, Tickets, nach, München]	

Welche Operation führen wir als nächstes durch?

Kauf Tickets nach München.

Shift-Reduce Dependency Parsing

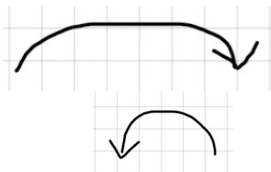
Buffer	Stack	Operation
[Kauf, Tickets, nach, München]	[ROOT]	SHIFT
[Tickets, nach, München]	[ROOT, Kauf]	SHIFT
[nach, München]	[ROOT, Kauf, Tickets]	SHIFT
[]	[ROOT, Kauf, Tickets, nach, München]	LEFTARC



Kauf Tickets nach München.

Shift-Reduce Dependency Parsing

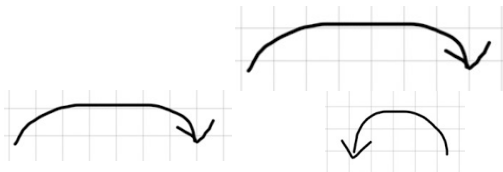
Buffer	Stack	Operation
[Kauf, Tickets, nach, München]	[ROOT]	SHIFT
[Tickets, nach, München]	[ROOT, Kauf]	SHIFT
[nach, München]	[ROOT, Kauf, Tickets]	SHIFT
[]	[ROOT, Kauf, Tickets, nach, München]	LEFTARC
[]	[ROOT, Kauf, Tickets, München]	RIGHTARC



Kauf Tickets nach München.

Shift-Reduce Dependency Parsing

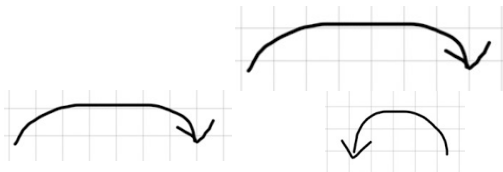
Buffer	Stack	Operation
[Kauf, Tickets, nach, München]	[ROOT]	SHIFT
[Tickets, nach, München]	[ROOT, Kauf]	SHIFT
[nach, München]	[ROOT, Kauf, Tickets]	SHIFT
[]	[ROOT, Kauf, Tickets, nach, München]	LEFTARC
[]	[ROOT, Kauf, Tickets, München]	RIGHTARC
[]	[ROOT, Kauf, Tickets]	RIGHTARC



Kauf Tickets nach München.

Shift-Reduce Dependency Parsing

Buffer	Stack	Operation
[Kauf, Tickets, nach, München]	[ROOT]	SHIFT
[Tickets, nach, München]	[ROOT, Kauf]	SHIFT
[nach, München]	[ROOT, Kauf, Tickets]	SHIFT
[]	[ROOT, Kauf, Tickets, nach, München]	LEFTARC
[]	[ROOT, Kauf, Tickets, München]	RIGHTARC
[]	[ROOT, Kauf, Tickets]	RIGHTARC
[]	[ROOT, Kauf]	RIGHTARC

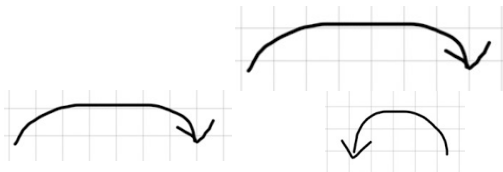


Kauf Tickets nach München.

ROOT

Shift-Reduce Dependency Parsing

Buffer	Stack	Operation
[Kauf Tickets, nach, München]	[ROOT]	SHIFT
[Tickets, nach, München]	[ROOT, Kauf]	SHIFT
[nach, München]	[ROOT, Kauf, Tickets]	SHIFT
[]	[ROOT, Kauf, Tickets, nach, München]	LEFTARC
[]	[ROOT, Kauf, Tickets, München]	RIGHTARC
[]	[ROOT, Kauf, Tickets]	RIGHTARC
[]	[ROOT, Kauf]	RIGHTARC
[]	[ROOT]	DONE



Kauf Tickets nach München.

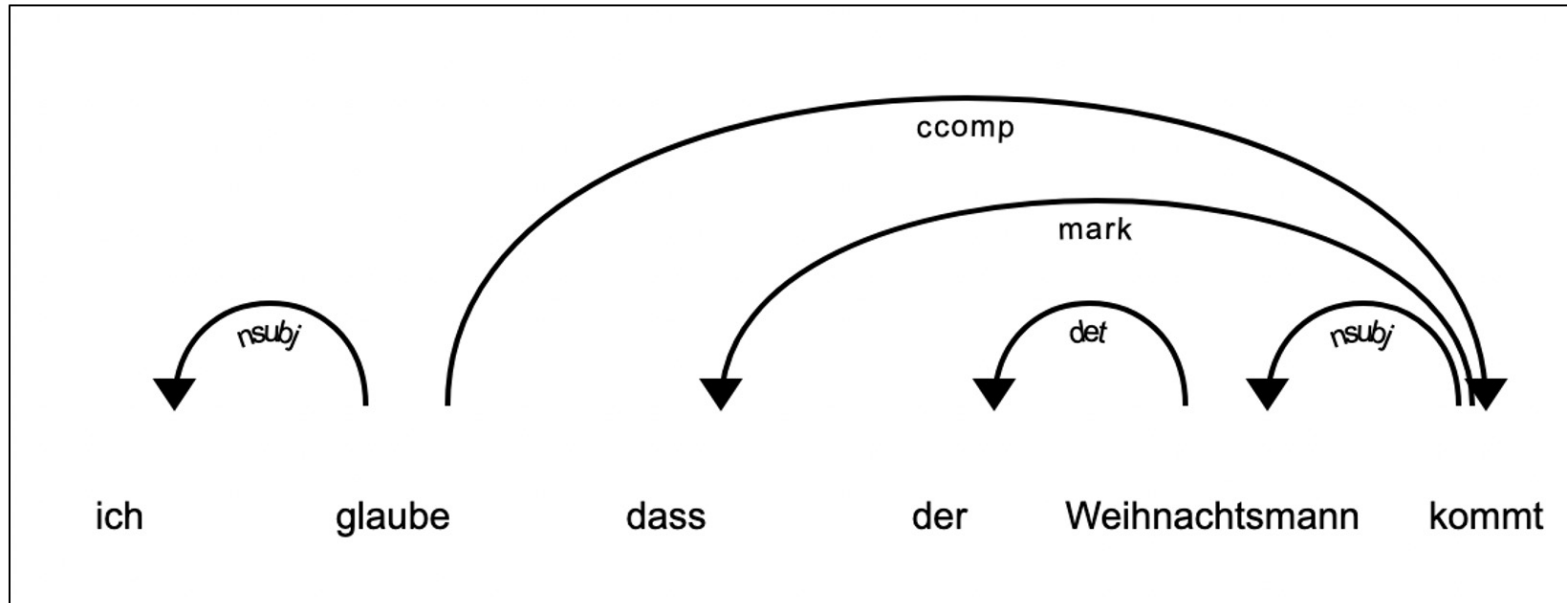
ROOT

Shift-Reduce Dependency Parsing

Projektive Struktur:

Für jede Dependenzrelation im Satz gilt, dass es einen Pfad vom Kopf zu allen Wörtern zwischen Kopf und Dependent gibt.

Beispiel:



Z.B. können wir jedes Wort zwischen „glaube“ (Kopf) und „kommt“ (Dependent) von „glaube“ aus erreichen.

Shift-Reduce Dependency Parsing

Nicht-projektive Struktur:

Es gibt Abhängigkeitsrelationen im Satz, für die die Bedingung für projektive Strukturen nicht gilt, d.h. wir können **nicht** bei allen Abhängigkeitsrelationen im Satz jedes Wort zwischen Kopf und Abhängiges erreichen.

Beispiel:



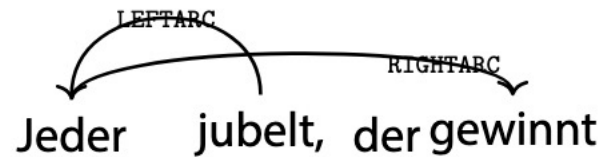
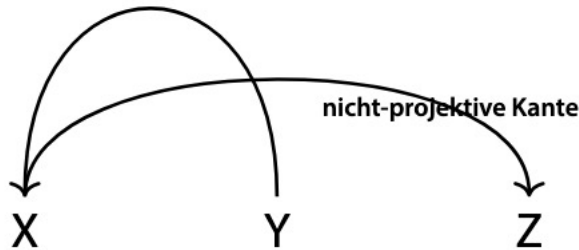
In diesem Beispiel ist "talk" Kopf von "about", und „yesterday“ steht zwischen Kopf und Abhängiges.

Es gibt keinen Pfad vom Kopf („talk“) zu „yesterday“, daher handelt es sich um eine nicht-projektive Struktur.

vgl. Vorlesungsfolien

Shift-Reduce Dependency Parsing

Beim Dependency-Parsing können nicht-projektive Strukturen problematisch sein. Der übergangsbasierte **Shift-Reduce-Dependency-Parser** kann beispielsweise mit **nicht-projektiven Strukturen nicht umgehen**.



vgl. Vorlesungsfolien

Was ist das Problem bei Parsen einer solchen Struktur?

Wir würden hier X und Y auf den Stack shiften, dann REDUCE-LEFTARC anwenden und den Dependenten (X) vom Stack nehmen. Wenn wir X vom Stack nehmen, können wir aber den Rest des Satzes nicht mehr parsen!