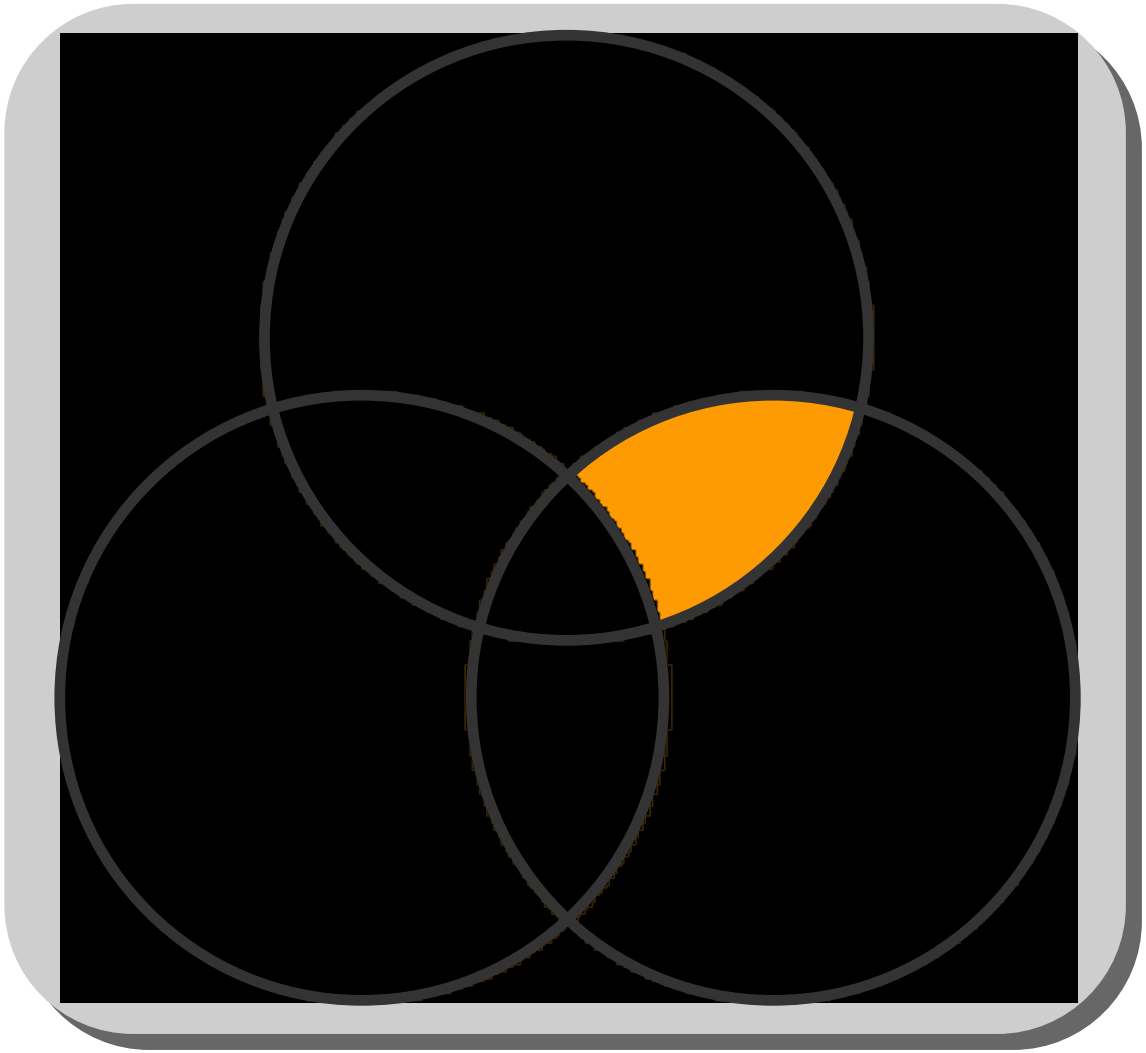
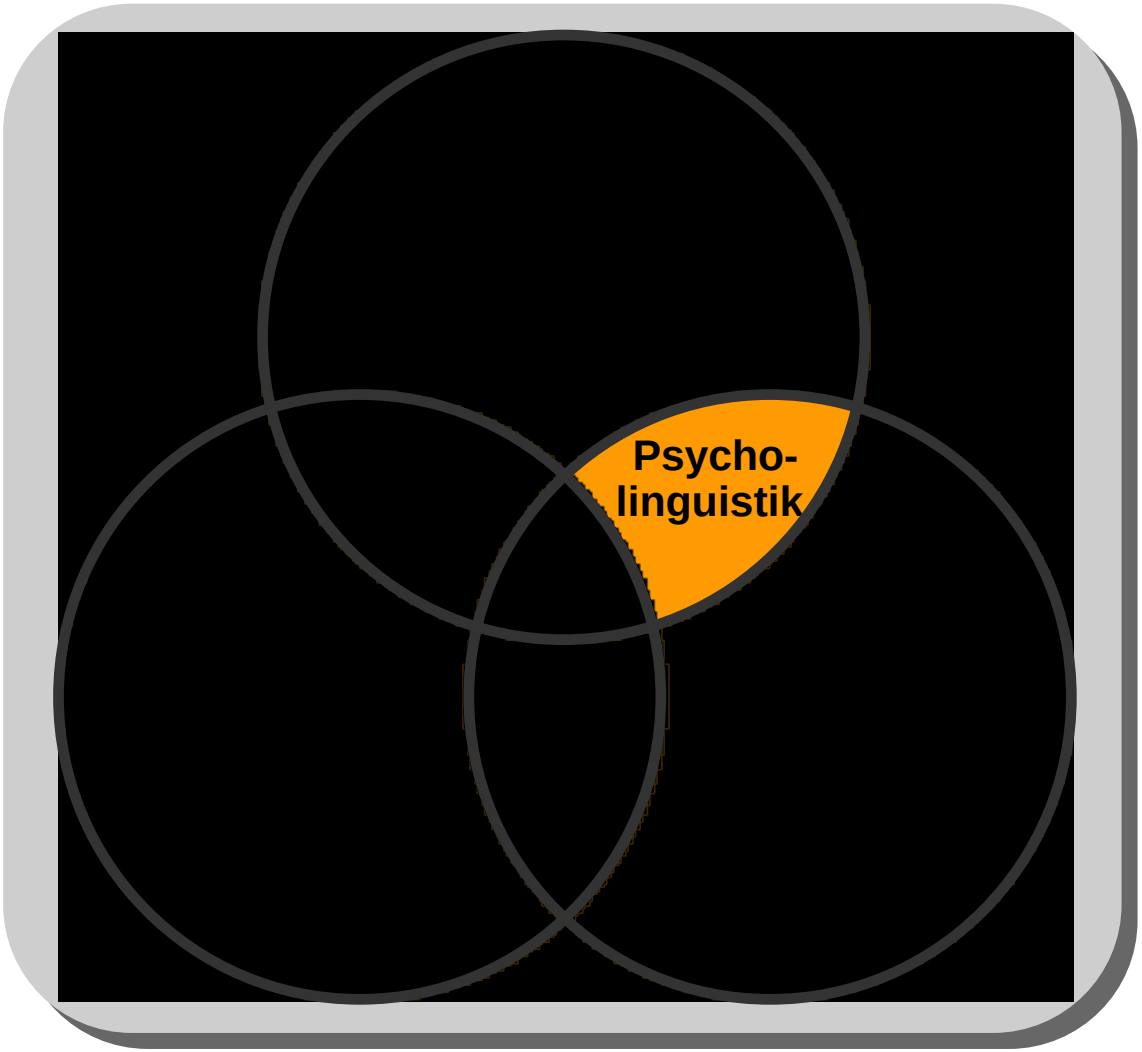
Einführung in die Computerlinguistik

Zusammenfassung

NACHBARWISSENSCHAFTEN



NACHBARWISSENSCHAFTEN



Forschungsgebiete der CL

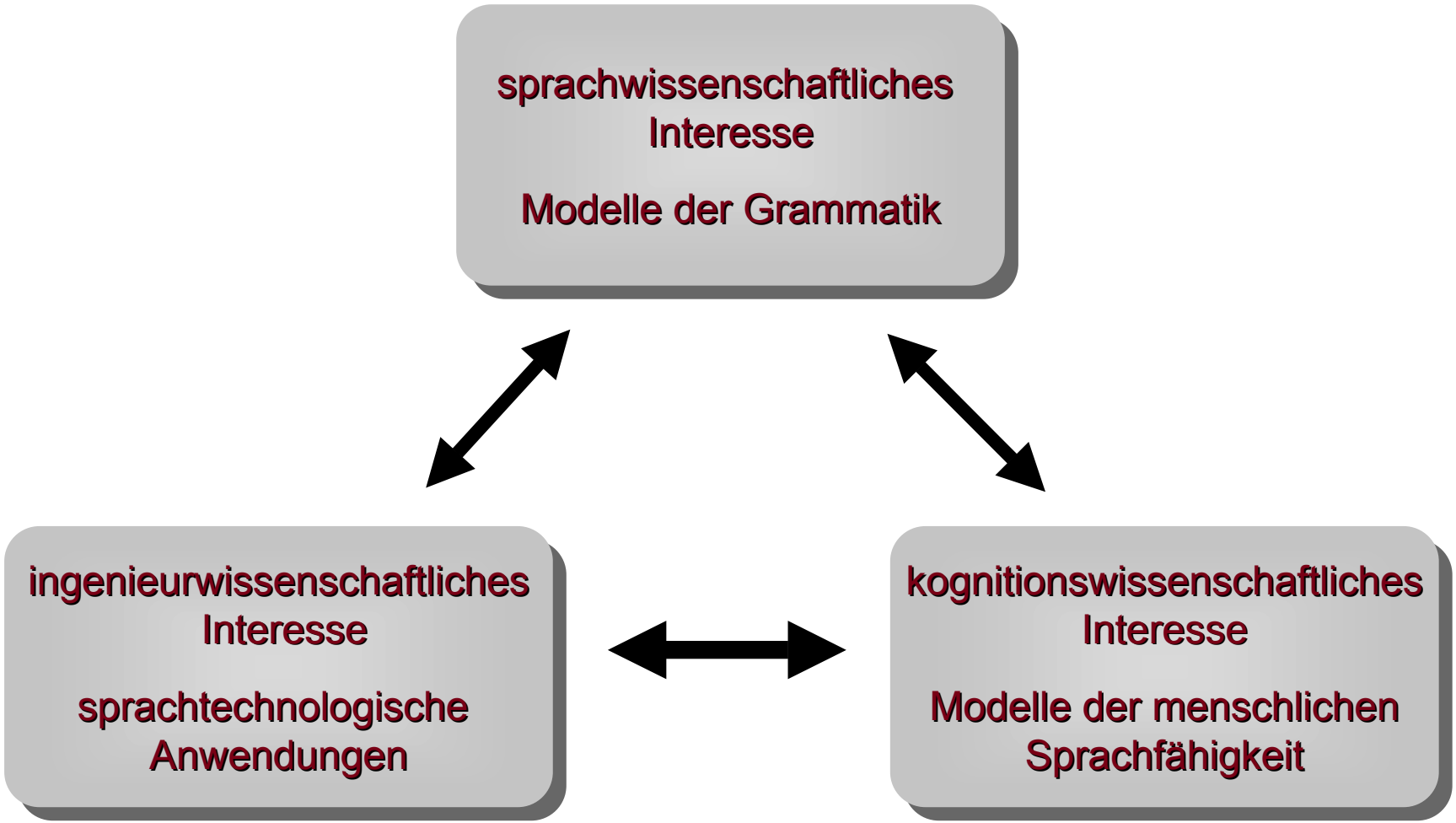
- Theoretische Computerlinguistik
  - Teilgebiet der Linguistik
  - Entwurf, Implementierung und Untersuchung berechenbarer Modelle natürlicher Sprache
  - Ziel: Beitrag zur Verbesserung der zugrundeliegenden linguistischen und psychologischen Theorien

  $\Rightarrow$ Computational Linguistics (CL)

- Angewandte Computerlinguistik
  - Interdisziplinäres Forschungsgebiet zwischen Informatik und Linguistik
  - Maschinelle Verarbeitung natürlicher Sprache
  - Ziel: Softwareanwendungen, die einen Teil der natürlichen Sprache simulieren

  $\Rightarrow$ Natural Language Processing (NLP)

Motivationen



Fragestellungen der theoretischen CL

- Wie können Phänomene der natürlichen Sprache adäquat repräsentiert werden? Welche Mechanismen stehen zur Beschreibung der Phänomene zur Verfügung?

- Welche Eigenschaften muss ein Formalismus aufweisen, um relevante Aspekte natürlicher Sprache angemessen repräsentieren zu können? Wie komplex ist der entsprechende Formalismus?

- Welche Komplexität weist die natürliche Sprache bzw. ein bestimmter Phänomenbereich derselben auf, und inwieweit kann die Komplexität effektiv bewältigt werden?

$\Rightarrow$ Wie sieht eine adäquate Modellierung natürlicher Sprache aus? Welche Komplexität besitzt sie?

Fragestellungen der angewandten CL

- Wie kann sprachliches Wissen erfolgreich auf einer Maschine modelliert werden?

- Was sind die Probleme, die sich bei der konkreten Implementierung stellen, und wie können sie bewältigt werden?

- Welche Formalismen eignen sich für die Modellierung welcher (evtl. einzelsprachlichen) Phänomene bzw. welcher Aspekte einer Sprache?

$\Rightarrow$ Wie sieht ein konkreter Algorithmus zur Verarbeitung natürlichsprachlicher Äußerungen aus?

Methoden der CL

Symbolische Methoden

- Parsing ist die Analyse natürlicher Sprache anhand von Grammatiken auf Basis der *Theorie der Automaten und formalen Sprachen*.
- Grammatikformalismen basieren auf *formalen Logiken* zur Repräsentation und Verarbeitung linguistischen Wissens (Inferenz).

Statistische Methoden

- Statistische Modelle über grossen Textmengen (Korpora) basieren auf *Wahrscheinlichkeitstheorie* und evt. *Informationstheorie*.

Hybride Methoden (gemischte Methoden)

- Statistisches Parsing verbindet z.B. symbolische und statistische Methoden

Subsymbolische Methoden

- Neuronale Netze

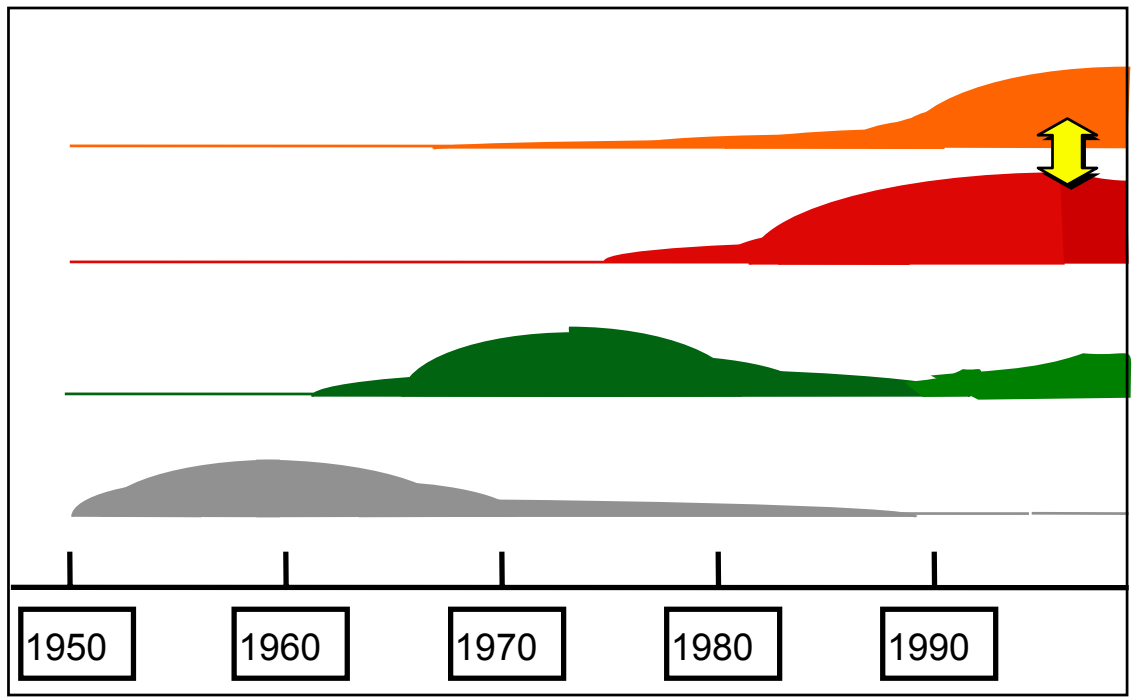
Hauptansätze der CL

statistische und konnektionistische Methoden in der CL

deklarative linguistische Formalismen in der CL

spezielle Verfahren für die CL

direkte Programmierung, keine Trennung von Beschreibung und Verarbeitung



Probleme der CL

Ambiguität (Mehrdeutigkeit) führt zur Explosion der Analysen:

- Polysemie: *Bank* (Gebäude, Institution, Sitzgelegenheit)
- Komposita: *Aluminiumherstellung* z.B. *alu+mini+umher+stellung* (+11)
- Skopus: ((*alte Männer*) *und Frauen*) vs. (*alte* (*Männer und Frauen*))
- PP-Zuordnung: *Peter kauft **das Auto mit Heckspoiler**.*
  *Peter **kauft** das Auto **mit Kreditkarte**.*
  *Peter **kauft** das Auto **mit Gabi**.*

Präpositionalphrase

Robustheit erfordert Fehlertoleranz und vollständiges Wissen

- mangelnde Fehlertoleranz
- unvollständige Lexika
  Funktionsfähigkeit
- unvollständige Grammatiken

} behindern die von CL-Anwendungen

Dilemma: je robuster (vollständiger), desto mehr Ambiguitäten.

Komponenten eines Sprachmodells

1. Spracherkennung und -ausgabe (**Phonetik** und **Phonologie**)

2. Struktur und Verarbeitung von Wortformen (**Morphologie**)

3. Satzstruktur (**Syntax**)

4. Bedeutung einzelner Wörter (**Lexikalische Semantik**) und ihrer Kombination (**Kompositionelle Semantik**)

5. Wissen über Äußerungszusammenhänge (**Diskurs**) und über konventionelle Verhaltensweisen (**Pragmatik**)

6. Hintergrundwissen/Weltwissen

Wie komplex ist eine Sprache?

Um entscheiden zu können, welche Mittel man zur Lösung eines Problems benötigt, muss man sich über die Komplexität des Problems klar werden.

In der *Theorie der Automaten und formalen Sprachen* werden Sprachen in eine Komplexitätshierarchie gebracht:

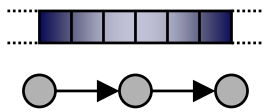
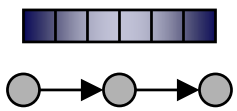
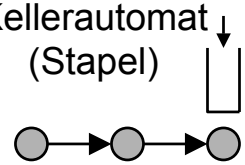
- Typ 0: rekursiv aufzählbar
- Typ 1: kontextsensitiv
- Typ 2: kontextfrei
- Typ 3: regulär

Chomsky-Hierarchie

Natürliche Sprache gilt als schwach kontextsensitiv.

Den Sprachklassen werden *Automatenklassen* zugeordnet.

Die Chomsky-Hierarchie

Sprache	*Automat*	*Grammatik*	*Erkennung*	*Abhängigkeit*
rekursiv aufzählbar	Turing Maschine 	unbeschränkt Baa → ε	unentscheidbar	beliebig
kontext-sensitiv	Linear gebunden 	kontext-sensitiv At → aA	NP-vollständig 	überkreuzt 
kontext-frei	Kellerautomat (Stapel) 	kontextfrei S → gSc	polynomiell 	eingebettet 
regulär	Endlicher Automat 	regulär A → cA	linear 	strikt lokal 

nach D. Searls

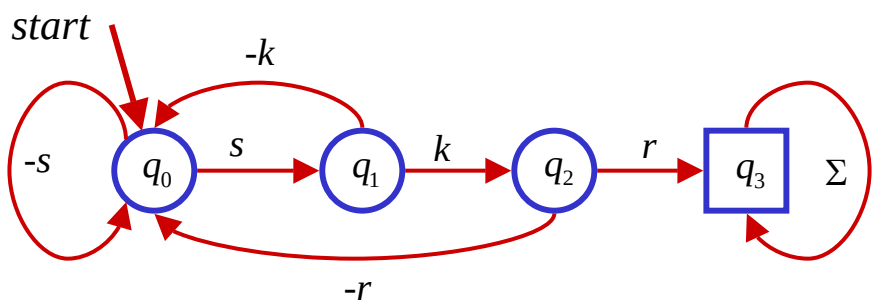
Endliche Automaten

Endliche Automaten sind die einfachste Automatenklasse.

Sie bestehen aus Zuständen, die mit beschrifteten Übergängen verbunden sind (Übergangsfunktion).

Eine Kette von Symbolen aus einem Alphabet Σ gilt als akzeptiert, wenn beginnend mit dem Startzustand die ganze Kette verarbeitet werden kann und der Automat sich dann in einem Endzustand befindet.

Der abgebildete Automat erkennt Sprachen, deren Ketten die Teilkette *skr* enthalten.



z.B. für ein Alphabet $\Sigma = \{a,b,c,s,k,r\}$, $L = \Sigma^* skr \Sigma^*$, wobei Σ^* die Menge aller Ketten ist, die sich aus beliebigen Symbolen aus dem Alphatet Σ zusammensetzt. Der Stern heisst Kleenscher Stern. Beispielsweise ist $aabcskrab \in L$, aber $aabcska \notin L$.

Kontextfreie Grammatiken

Kontextfreie Grammatiken sind neben den einfacheren regulären Grammatiken die wichtigsten Grammatiken für die Computerlinguistik.

Die Regeln einer kontextfreien Grammatik definieren zwei Relationen:

- Unmittelbare Dominanz zwischen Mutterkategorie und Tochterkategorien
- Lineare Präzedenz zwischen Schwesterkategorien

Hier ist eine kontexfreie Grammatik für ein Fragment der deutschen Sprache:

$S \rightarrow NP\ VP$
$NP \rightarrow Det\ N$
$VP \rightarrow V\ NP$
$Det \rightarrow$ Jeder
$Det \rightarrow$ eine
$N \rightarrow$ Mann
$N \rightarrow$ Frau
$V \rightarrow$ liebt

Legende

S	Satz
NP	Nominalphrase
VP	Verbalphrase
Det	Determiner (Artikel)
N	Nomen (Substantiv)
V	Verb

Parsing

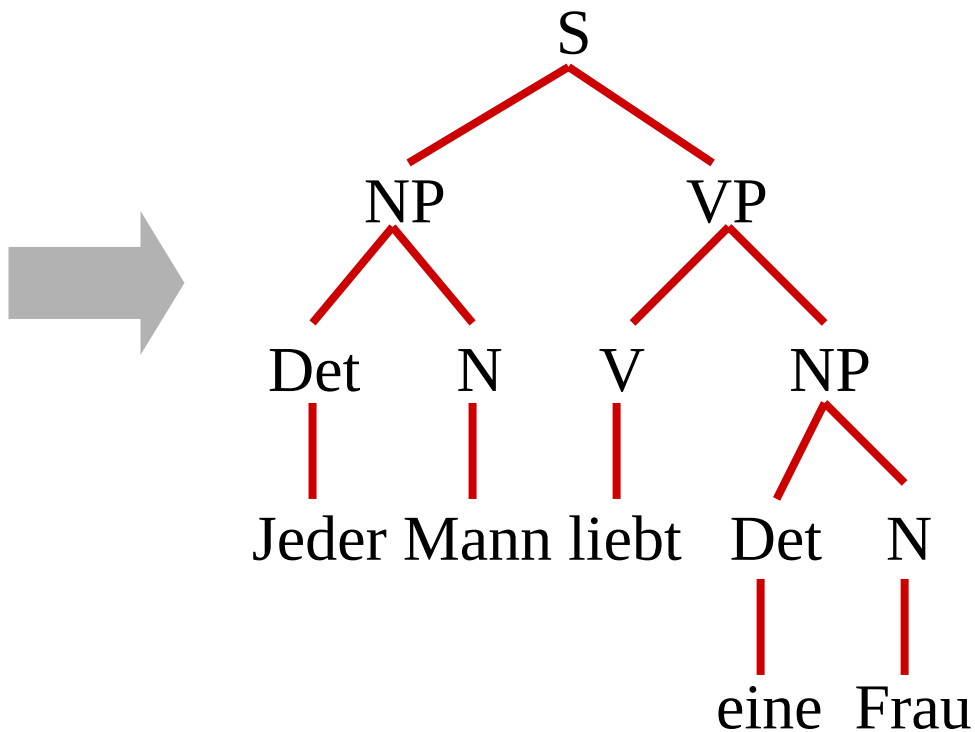
engl. *to parse*: „grammatisch zerlegen“

Ein Parser ist ein Automat, der auf Basis einer Grammatik für eine Kette einen Ableitungsbaum (parse tree) erzeugt.

Grammatik

&

„Jeder Mann liebt eine Frau“



Parsingstrategien

Parsingstrategien unterscheiden sich durch die Reihenfolge, in der bei der Konstruktion des Syntaxbaums die Knoten im Baum besucht werden (Traversierung).

- top-down
- bottom-up
- left-corner

depth-first	left-to-right
breadth-first	right-to-left

Deklarativ vs. Prozedural

Eine Grammatik ist eine ***deklarative*** Beschreibung der wohlgeformten Syntaxbäume einer Sprache.

Eine deklarative Beschreibung stellt einen logischen Sachverhalt dar.

Ein Algorithmus ist eine Folge von Anweisungen (eine Prozedur), wie man in endlich vielen Schritten von einem Ausgangszustand (zu lösendes Problem) zu einem Zielzustand (gelöstes Problem) kommt.

Ein Parser verwendet einen Algorithmus, um Grammatiken zu interpretieren: eine ***prozedurale*** Parsingstrategie.

Anwendungen (1)

Korrekturprogramme: Rechtschreibkorrektur, Grammatikkorrektur Korrekturvorschläge, Verbesserung von Texterfassung mittels OCR.

Computergestützte Lexikographie: Hilfe bei der Erstellung und Pflege von Lexika; Akquisition lexikalischer Information, Repräsentation lexikalischer Information, Bereitstellung der lexikalischen Information für Anwendungen.

Volltextsuche (Information Retrieval): Indexkonstruktion, Auswertung von Suchanfragen, Retrievalmodell

Textmining: Strukturierung großer Textkollektionen, Textklassifikation, Schlüsselwortextraktion, Aufbau einer Taxonomie

Textklassifikation: Erlernen von Klassenprofilen anhand von Trainingsdaten, Klassifikationsalgorithmus

Informationsextraktion: Identifizierung relevanter Information in Texten, Instantiierung von Templates

Textzusammenfassung: Reduktion / Verdichtung, Textproduktion

Anwendungen (2)

Sprachsynthesesysteme: Produktion gesprochener Sprache aus geschriebener Sprache, Computerarbeitsplatz für Blinde, telefonische Auskunftsysteme, Navigationsysteme

Spracherkennungssysteme: Diktiersysteme, telefonische Auskunftsysteme; Signalanalyse, Geräuschfilterung, Adaption an verschiedene Sprecher

Natürlichsprachliche Retrieval-Schnittstellen: z.B. natürlichsprachliche Anfragen an Bibliothekskataloge

Dialogsysteme: ELIZA, automatische Auskunftsysteme, natürlichsprachliche Benutzerschnittstellen

Sprachlehr- und -lernsysteme: Hilfe bei dem Erwerb von Fremdsprachen; Anpassung an das individuelle Arbeitstempo und den Kenntnisstand, ortsungebunden, zeitlich flexibel, objektiv, nicht ermüdend

Fazit: Beruf ComputerlinguistIn

Computerlinguistische Arbeit erfordert Wissen aus mehreren Bereichen:

- Linguistik
- Informatik
- Mathematik
- Philosophie
- Logik
- Informationswissenschaft

Sofern es um bestimmte Anwendungsdomänen geht, können natürlich weitere Fachbereiche involviert sein:
Philologie(n), Biologie, Soziologie, Forensik, Kryptologie, …

Je nach Spezialisierung kann der Schwerpunkt einzelner ComputerlinguistInnen stark auf bestimmte der genannten Bereiche verlagert sein.