

Wörter, Texte und Frequenzen

Annotation: Tools und Pipelines

Andreas Blombach, Philipp Heinrich

Lehrstuhl für Korpus- und
Computerlinguistik

<https://www.linguistik.phil.fau.de>



Friedrich-Alexander-Universität
Philosophische Fakultät und
Fachbereich Theologie

Korpusannotation: Wortebene

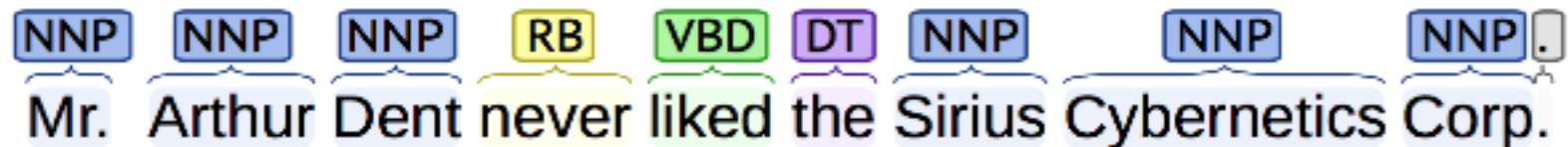
- Jedem (laufenden) Wort wird eine Kategorie zugeordnet
→ sog. **Tagging** (= Etikettierung)
 - Voraussetzung: Text muss in Wörter zerlegt sein
- **Tokenisierung**
 - **Token** = Wort, Zahl, Symbol (☺), Satzzeichen, ...
 - im Gegensatz zu **Typen** = verschiedene Wörter
- Kann schwieriger sein, als man vermuten würde ...

@Mia1234 #semibk [1] Das schließt direkt an die vorige Frage von @DieMaJa22 an. In jedem Fall gibt es (wie auch in der Sitzung ... @Mia1234 #semibk [2]am BspChats gezeigt) starkeHinweise darauf, dass(wie auch imRealLife) diverseFaktoren die sprVariation beeinflussen: <http://tinyurl.com/3umxkuh>

<https://sites.google.com/site/empirist2015/> (Beißwenger et al. 2016)

Annotation auf Wortebene

- Zentral: Wortartenannotierung = **POS-Tagging**
 - Substantiv (**noun**), Adjektiv, Verb, Adverb, Pronomen, Präposition, Konjunktion, Zahl, Satzzeichen, ...
 - engl. POS = **part of speech**
- Tagset = Kategorienschema
 - meist feinere Unterschiede: Sg/Pl, inf./fin./imp., ...



- auch: Lemmatisierung (hier kein Tagset!), semantische Kategorien, emotionale Valenz, Schwierigkeitsgrad (CEFR), ...

Deutsch: STTS-Tagset

| | |
|----------------|---|
| ADJA | attributives Adjektiv |
| ADJD | adverbiales / prädikatives Adjektiv |
| ADV | Adverb <i>schon, bald, doch</i> |
| APPR | Präposition / Zirkumposition links |
| APPRART | Präposition mit Artikel fusioniert <i>zum</i> |
| APPO | Postposition <i>zufolge, wegen</i> |
| APZR | Zirkumposition rechts <i>von ... an</i> |
| ART | bestimmter oder unbestimmter Artikel |
| CARD | Kardinalzahlen (Ordinalzahl = ADJA) |
| FM | Fremdsprachliches Material |
| ITJ | Interjektion <i>mhm, ach, tja</i> |
| KOUI | unterordnende Konj. mit <i>zu</i> + Inf |
| KOUS | unterordnende Konjunktion mit Satz |
| KON | nebenordnende Konjunktion <i>und, oder</i> |
| KOKOM | Vergleichskonjunktion <i>als, wie</i> |
| NN | normales Nomen |
| NE | Eigennamen |
| PDS | substituierendes Demonstrativpron. |
| PDAT | attribuierendes Demonstrativpron. |
| PIS | substituierendes Indefinitpron. |
| PIAT | attrib. Indefinitpron. ohne Determiner |
| PIDAT | attrib. Indefinitpron. mit Determiner |
| PPER | Personalpronomen (nicht reflexiv) |
| PPOSS | substituierendes Possessivpronomen |
| PPOSAT | attribuierendes Possessivpronomen |
| PRELS | substituierendes Relativpronomen |
| PRELAT | attribuierendes Relativpronomen |

| | |
|----------------|--|
| PRF | reflexives Personalpronomen |
| PWS | substituierendes Interrogativpron. |
| PWAT | attribuierendes Interrogativpronomen |
| PWAV | adverbiales Interrogativ-/Relativpron. |
| PAV | Pronominaladverb <i>dafür, deswegen</i> |
| PTKZU | <i>zu</i> vor Infinitiv |
| PTKNEG | Negationspartikel <i>nicht</i> |
| PTKVZ | abgetrennter Verbzusatz <i>kommt ... an</i> |
| PTKANT | Antwortpartikel <i>ja, nein, danke</i> |
| PTKA | Partikel bei Adjektiv/Adverb <i>am, zu</i> |
| TRUNC | Kompositions-Erstglied <i>Unter- und ...</i> |
| VVFIN | finite Verb, voll (= lexikalisch) |
| VVIMP | Imperativ, voll |
| VVINFIN | Infinitiv, voll |
| VVIZU | Infinitiv mit <i>zu</i> , voll |
| VVPP | Partizip Perfekt, voll |
| VAFIN | finite Hilfsverb |
| VAIMP | Imperativ, Hilfsverb |
| VAINFIN | Infinitiv, Hilfsverb |
| VAPP | Partizip Perfekt, Hilfsverb |
| VMFIN | Finite Modalverb |
| VMINFIN | Infinitiv, Modalverb |
| VMPP | Partizip Perfekt, Modalverb |
| XY | Nichtwort mit Sonderzeichen <i>3:7, H2O</i> |
| \$, | Komma <i>,</i> |
| \$. | Satzbeendende Interpunktion <i>.?!;:</i> |
| \$(| sonstige Satzzeichen (intern) <i>- [] ()</i> |

| | |
|----------------|--|
| CC | Coordinating conjunction |
| CD | Cardinal number |
| DT | Determiner |
| EX | Existential <i>there</i> |
| FW | Foreign word |
| IN | Preposition / subordinating conjunction |
| IN/that | Subordinating conjunction <i>that</i> |
| JJ | Adjective (positive) |
| JJR | Adjective (comparative) |
| JJS | Adjective (superlative) |
| LS | List item marker |
| MD | Modal verb |
| NN | Noun, singular or mass |
| NNS | Noun, plural |
| NP | Proper noun, singular |
| NPS | Proper noun, plural |
| PDT | Predeterminer |
| POS | Possessive ending ('s) |
| PP | Personal pronoun |
| PP\$ | Possessive pronoun |
| RB | Adverb |
| RP | Particle |
| SYM | Symbol (mathematical/ scientific) |
| TO | to (any usage) <i>fly to Paris, ready to go, ...</i> |
| UH | Interjection |
| # | Pound sign £ |
| \$ | Dollar sign \$ |

| | |
|------------------|--|
| VB | Verb <i>be</i> , base form |
| VBD | Verb <i>be</i> , past tense |
| VBG | Verb <i>be</i> , gerund/ progressive |
| VBN | Verb <i>be</i> , past participle |
| VBP | Verb <i>be</i> , non-3rd pers. sg. present |
| VBZ | Verb <i>be</i> , 3rd pers. sg. present tense |
| VH | Verb <i>have</i> , base form |
| VHD | Verb <i>have</i> , past tense |
| VHG | Verb <i>have</i> , gerund/ progressive |
| VHN | Verb <i>have</i> , past participle |
| VHP | Verb <i>have</i> , non-3rd pers. sg. present |
| VHZ | Verb <i>have</i> , 3rd pers. sg. present tense |
| VV | Lexical verb, base form |
| VVD | Lexical verb, past tense |
| VVG | Lexical verb, gerund/ progressive |
| VVN | Lexical verb, past participle |
| VVP | Lexical verb, non-3rd pers. sg. present |
| VVZ | Lexical verb, 3rd pers. sg. present tense |
| WDT | Wh-determiner |
| WP | Wh-pronoun |
| WP\$ | Possessive wh-pronoun |
| WRB | Wh-adverb |
| SENT | Sentence-final punctuation . ! ? |
| , | Comma , |
| : | Colon, semi-colon : ; |
| (AA) | Comma ([]) |
| ` ` A ' ' | Comma " ... ' ... " |

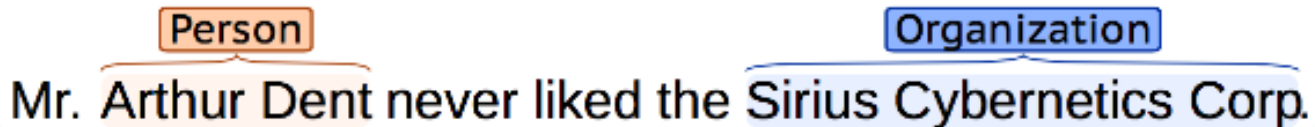
Universal-Dependencies-Tags (sprachübergreifend)

| Open class words | Closed class words | Other |
|------------------|--------------------|--------------|
| <u>ADJ</u> | <u>ADP</u> | <u>PUNCT</u> |
| <u>ADV</u> | <u>AUX</u> | <u>SYM</u> |
| <u>INTJ</u> | <u>CCONJ</u> | <u>X</u> |
| <u>NOUN</u> | <u>DET</u> | |
| <u>PROPN</u> | <u>NUM</u> | |
| <u>VERB</u> | <u>PART</u> | |
| | <u>PRON</u> | |
| | <u>SCONJ</u> | |

- ADJ: adjective
- ADP: adposition
- ADV: adverb
- AUX: auxiliary
- CCONJ: coordinating conjunction
- DET: determiner
- INTJ: interjection
- NOUN: noun
- NUM: numeral
- PART: particle
- PRON: pronoun
- PROPN: proper noun
- PUNCT: punctuation
- SCONJ: subordinating conjunction
- SYM: symbol
- VERB: verb
- X: other

Segmente und Strukturen

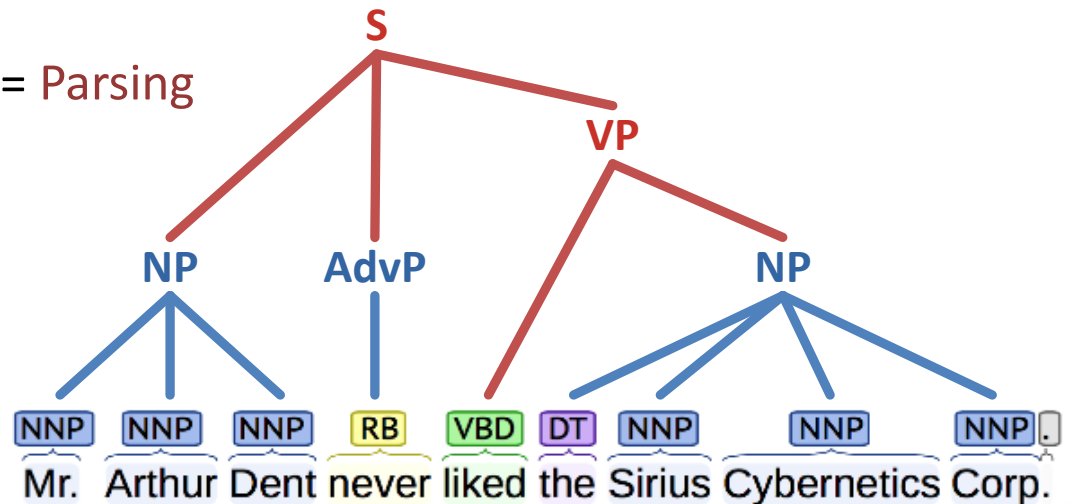
- Erkennung von speziellen Wortfolgen (Segmenten bzw. *spans*) und ihre Kategorisierung
- z.B. Eigennamen (**NER** = *named entity recognition*)



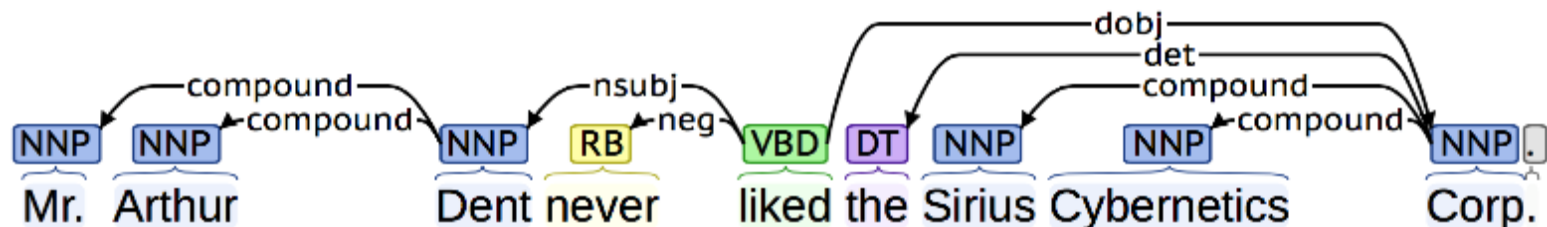
- Was könnten weitere interessante Segmente sein?
- Kann auch als Tagging operationalisiert werden
 - | | | | | | | | | |
|-----|---------------|---------------|-------|-------|-----|--------------|--------------|--------------|
| O | B-PERS | I-PERS | O | O | O | B-ORG | I-ORG | I-ORG |
| Mr. | Arthur | Dent | never | liked | the | Sirius | Cybern. | Corp. |

Strukturen: Parsing

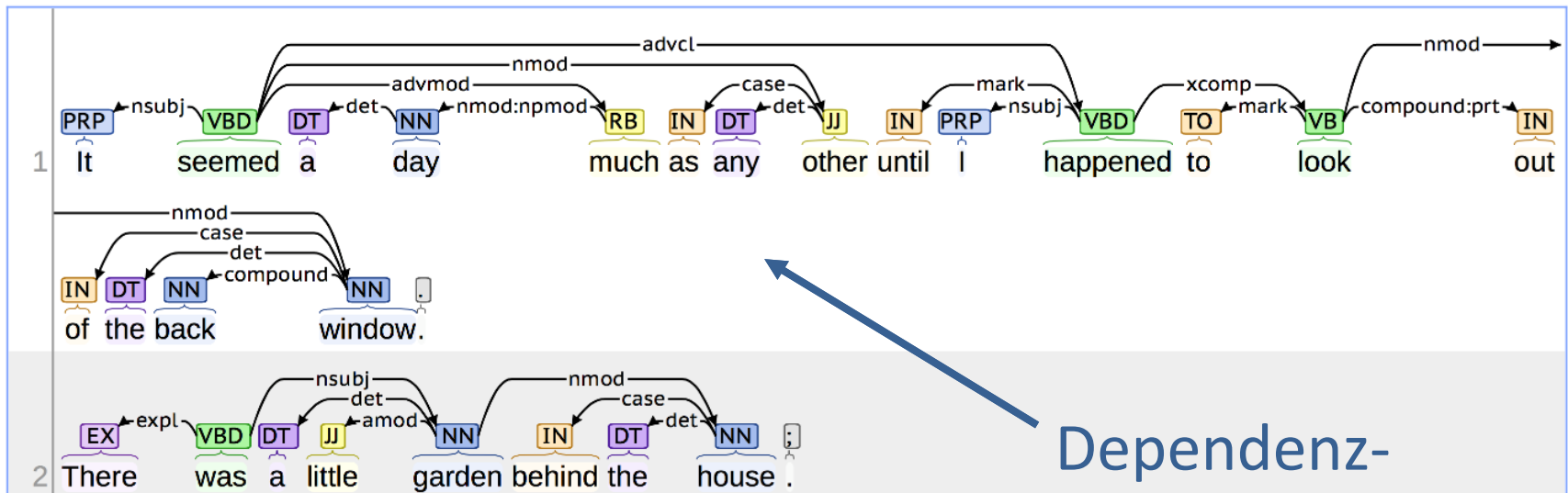
- Erkennung der Satzstruktur = **Parsing**
- **Phrasenstruktur** als baumförmige Hierarchie



- alternativ: „minimale“ Phrasen als flache Segmente → **Chunk-Parsing**
- **Dependenz-Parsing** findet direkte Abhängigkeiten zwischen Wörtern



Beispiel: Syntaktische Analyse



Dependenz-
Graph

Zum Ausprobieren:

- <http://corenlp.run/>
- <https://explosion.ai/demos/displacy>
- <http://nlp.uoregon.edu/trankit>

Manuelle Annotation

- Kleine Korpora werden oft manuell annotiert
 - z.B. digitale Editionen, Reden eines Präsidenten, ...
- **Annotationsschema** und -kategorien (**Tagset**)
- **Richtlinien** (**Guidelines**)
 - detaillierte Beschreibung und Abgrenzung der Zielkategorien (z.B. für [STTS](#))
 - zusätzlich: Beispielsammlung für schwierige Einzelfälle
- Annotationswerkzeuge (meist Web-basiert)
 - z.B. INCEpTION (<https://inception-project.github.io>), Prodigy (<https://prodi.gy>)
- **Inter-Annotator Agreement** (IAA)
 - wichtig! – überprüft Reliabilität und Validität der Annotation
 - Flüchtigkeitsfehler vs. systematische Differenzen
 - Adjudikation für Endfassung der Annotation

Automatische Annotation

- Für größere Korpora ist eine manuelle Annotation zu teuer und zeitaufwendig
- Auch in den Digital Humanities ...
 - Romane von Charles Dickens ca. 4 Mio. Wörter
 - Deutsches Gutenberg-Archiv > 100 Mio. Wörter
 - Early English Books (EEBO) > 500 Mio. Wörter
 - Times Online 1780–1900 ca. 4.000 Mio. Wörter

Automatische Annotation

- Erfolgreichster Ansatz: **maschinelle Lernverfahren**
 - ab ca. 1990 Einsatz von statistischen Modellen („**statistical revolution**“)
 - aktuell große Fortschritte mit **Deep Learning**
- **Trainingskorpus** (manuell annotiert)
 - wichtig: Konsistenz der Annotationen (→ IAA)
 - Flüchtigkeitsfehler scheinen weniger problematisch
- Evaluation auf separatem Testkorpus
 - Gefahr der Überanpassung an das Trainingskorpus
 - zusätzliches **development set** für Optimierung der Lernverfahren (**tuning**)
 - Kreuzvalidierung (**cross-validation**) nutzt alle Daten für Training & Evaluation
- Weiterführend: <https://web.stanford.edu/~jurafsky/slp3/>

Repräsentationsformat: XML

```
<?xml version="1.0" encoding="UTF-8"?>
<corpus>
  <story title="The Garden">
    <p>
      <s>
        <token pos="PP" lemma="it">It</token>
        <token pos="VBD" lemma="seem">seemed</token>
        <token pos="DT" lemma="a">a</token>
        <token pos="NN" lemma="day">day</token>
        <token pos="RB" lemma="much">much</token>
        <token pos="IN" lemma="as">as</token>
        <token pos="DT" lemma="any">any</token>
        <token pos="JJ" lemma="other">other</token>
        <token pos="IN" lemma="until">until</token>
        <token pos="PP" lemma="I">I</token>
        ...
      </s>
    </p>
  </story>
</corpus>
```

Repräsentationsformat: Vertical text (.vrt)

```
<corpus>
<text title="The Garden" author="Stefan Evert" author_sex="male"
      date="1991-08-05">
<p num="1">
<s>
It      PP    it
seemed VBD    seem
a       DT    a
day     NN    day
much    RB    much
as      IN    as
any     DT    any
other   JJ    other
until   IN    until
I       PP    I
...
</s>
</p>
</text>
</corpus>
```

Repräsentationsformat: CoNLL-Format(e)

story: "The Garden"

paragraph #1

| | | | |
|---|--------|------|------|
| 1 | It | PP | it |
| 2 | seemed | VBD | seem |
| 3 | a | DT | a |
| 4 | fine | JJ | fine |
| 5 | day | NN | day |
| 6 | . | SENT | . |

| | | | |
|---|----------|------|----------|
| 1 | There | EX | there |
| 2 | was | VBD | be |
| 3 | an | DT | a |
| 4 | elephant | NN | elephant |
| 5 | . | SENT | . |

this is the end of the file

aktuell: CoNLL-U (<https://universaldependencies.org/format.html>)

Übersicht: Tools

Manuelle Annotation: Tools

- WebAnno / **INCEpTION** (linguistischer Fokus):
 - <https://webanno.github.io/webanno/documentation/>
 - <https://www.youtube.com/user/webanno>
 - <https://inception-project.github.io>
 - <https://youtube.com/playlist?list=PL5Hz5pttaj96SlXHGRZf8KzIYvpVHloL->
- **prodigy** (linguistischer Fokus):
 - <https://prodi.gy>
- **CATMA** (literaturwissenschaftlicher Fokus)
 - z.B. Annotation von wörtlicher und indirekter Rede
 - <https://fortext.net/routinen/lerneinheiten/manuelle-annotation-mit-catma>

Automatische Annotation: komplette Pipelines (1)

- Stanford **CoreNLP** (<https://stanfordnlp.github.io/CoreNLP/>)
 - langlaufendes Projekt, Java
 - Tokenisierung, POS-Tagging, Lemmatisierung, NER, Parsing, Koreferenzauflösung, Sentiment-Analyse, ...
- **Stanza** (<https://stanfordnlp.github.io/stanza/>)
 - Python, Deep Learning, Interface zu CoreNLP (z.B. für Koreferenzauflösung relevant)
 - Tokenisierung, POS-Tagging, Lemmatisierung, NER, Dependenzparsing, Sentiment-Analyse
- **spaCy** – „fastest in the world“ (<https://spacy.io>)
 - Python, Deep Learning
 - Tokenisierung, POS-Tagging, Lemmatisierung, NER, Dependenzparsing

Automatische Annotation: komplette Pipelines (2)

- Trankit (<https://github.com/nlp-uoregon/trankit>)
 - Python, Deep Learning
 - mehrsprachige Annotation möglich
 - Tokenisierung, POS-Tagging, Lemmatisierung, NER, Dependenzparsing
- Apache OpenNLP (<https://opennlp.apache.org/>)
 - Java
 - Tokenisierung, POS-Tagging, Lemmatisierung, NER, Dependenzparsing, Koreferenzauflösung
- UDPipe (<http://ufal.mff.cuni.cz/udpipe>)
 - C++/Python, als Bibliothek für diverse C++, C#, Python, Perl und Java verfügbar
 - UD steht für *Universal Dependencies*
 - Tokenisierung, POS-Tagging, Lemmatisierung, Dependenzparsing

Automatische Annotation: Tokenisierung und Tagging

- reine **Tokenisierer**
 - Python: [SoMaJo](#) (DE, EN)
 - generischer Tokenisierer: [Unitok](#)
 - Tokenisierer von NLTK ist bestenfalls mittelmäßig
 - wichtig: Tokenisierung und weitere Verarbeitung müssen kompatibel sein!
- Part-of-speech-**Tagger** (oft mit eigenem Tokenisierer)
 - [TreeTagger](#) (schnell, einfach, viele Sprachen, inkl. Lemmatisierung)
 - [RNNTagger](#) (Deep-Learning-Nachfolger des TreeTaggers, Python, inkl. Lemm.)
 - Python: [SoMeWeTa](#) (DE, EN, FR)
 - Twitter data (EN): [TweetNLP](#)
 - und viele weitere spezialisierte Tokenisierer und Tagger für diverse Sprachen
- Eigene Pipeline im Webservice erstellen:
https://weblicht.sfs.uni-tuebingen.de/weblichtwiki/index.php/Main_Page