Parser und Tagger

Stanford Parser

- Java-Implementierung eines probabilistischen NLP-Parser
- Kann sowohl Dependenzstrukturen als auch Phrasenstrukturen ausgeben
- GNU General Public License
- Download: `http://nlp.stanford.edu/software/lex-parser.shtml`
- Auf unseren Servern unter `/resources/processors/parser/stanfordparser-3.6.0`
- Modelle für Englisch, Deutsch, Französisch, Chinesisch, Arabisch, Italienisch, Bulgarisch, Portugiesisch, ...
- Online-Version: `http://nlp.stanford.edu:8080/parser/`

Dependenzen im Stanford Parser

- Mehrere Formalismen verfügbar
- Alle Dependenzen sind binäre Relationen
- Tokens werden zusammen mit dem Index gezeigt
  Beispiel:
  `nsubj(makes-8, Bell-1)`
- Standard: *Universal Dependencies*
  - Insgesamt: 40 grammatische Funktionen
  - sprachunabhängig

Dependenztypen

- Universal Dependencies - alle Dependenzen werden individuell dargestellt
  Beispiel:
  `prep(based-7, in-8)`
  `pobj(in-8, LA-9)`
- Enhanced Universal Dependencies - u.a.: manche Dependenzen (z.B. mit Präpositionen, Konjunktionen) werden zusammengefasst.
  Beispiel:
  `prep:in(based-7, LA-9)`

Eingabe-Datei

- Einfache Textdatei
- Ein oder mehrere Sätze pro Zeile (keine Leerzeilen zwischen den Zeilen)
- Mehrere Sätze in einer Zeile werden mit einem Sentence Splitter getrennt

Beispiel:

```
The strongest rain ever recorded in India shut down
the financial hub of Mumbai, snapped communication
lines, closed airports and forced thousands of people
to sleep in their offices or walk home during the
night, officials said today.
```

Generierung der Ausgabe

```
:~$ java -mx200m edu.stanford.nlp.parser.lexparser.\
LexicalizedParser -outputFormat "wordsAndTags,penn,typedDependencies
"englishPCFG.ser.gz text
```

- **englishPCFG.ser.gz** : Vortrainiertes Modell für Englisch
- **-outputFormat** : Format für die Ausgabe (Mehrere Optionen durch Kommata getrennt)
- **text** : Eingabe-Datei mit den Sätzen
- **-sentences newline** : optional. Unterdrückt das Starten des Sentence Splitters

Das Ergebnis wird auf STDOUT ausgegeben.

Das Ausgabeformat

Die Optionen für `outputFormat` im Beispiel:

- `wordsAndTags`: Text mit POS-Tags
- `penn`: Syntaktische Baumstrukturen
- `typedDependencies`: Grammatische Relationen in Dependenzformat (Enhanced Universal Dependencies)

Ausgabe-Datei – Teil 1 und 2

```
The/DT strongest/JJS rain/NN ever/RB recorded/VBN in/IN
India/NNP shut/VBD down/RP the/DT financial/JJ hub/NN
of/IN Mumbai/NNP ,/, snapped/VBD communication/NN lines/NNS
,/, closed/VBD airports/NNS and/CC forced/VBD thousands/NNS
of/IN people/NNS to/TO sleep/VB in/IN their/PRP$ offices/NNS
or/CC walk/VB home/NN during/IN the/DT night/NN ,/,
officials/NNS said/VBD today/NN ./.
```

```
(ROOT
  (S
    (S
      (NP
        (NP (DT The) (JJS strongest) (NN rain))
        (VP
          (ADVP (RB ever))
          (VBN recorded)
          (PP (IN in)
            (NP (NNP India)))))
      (VP
        (VP (VBD shut)
          (PRT (RP down))
          (NP
            (NP (DT the) (JJ financial) (NN hub))
            (PP (IN of)
              (NP (NNP Mumbai)))))
        (, ,)
```

Ausgabe-Datei – Teil 2

```
      (VP (VBD snapped)
        (NP (NN communication) (NNS lines)))
      (, ,)
      (VP (VBD closed)
        (NP (NNS airports)))
      (CC and)
      (VP (VBD forced)
        (NP
          (NP (NNS thousands))
          (PP (IN of)
            (NP (NNS people))))
        (S
          (VP (TO to)
            (VP
              (VP (VB sleep)
                (PP (IN in)
                  (NP (PRP$ their) (NNS offices))))
              (CC or)
              (VP (VB walk)
                (NP (NN home))
                (PP (IN during)
                  (NP (DT the) (NN night)))))))))
  (, ,)
  (NP (NNS officials))
  (VP (VBD said)
    (NP (NN today)))
  (. .)))
```

Ausgabe-Datei – Teil 3 I

```
det(rain-3, The-1)
amod(rain-3, strongest-2)
nsubj(shut-8, rain-3)
nsubj(snapped-16, rain-3)
nsubj(closed-20, rain-3)
nsubj(forced-23, rain-3)
advmod(recorded-5, ever-4)
acl(rain-3, recorded-5)
case(India-7, in-6)
nmod:in(recorded-5, India-7)
ccomp(said-40, shut-8)
compound:prt(shut-8, down-9)
det(hub-12, the-10)
amod(hub-12, financial-11)
dobj(shut-8, hub-12)
case(Mumbai-14, of-13)
nmod:of(hub-12, Mumbai-14)
conj:and(shut-8, snapped-16)
ccomp(said-40, snapped-16)
compound(lines-18, communication-17)
```

Ausgabe-Datei – Teil 3 II

```
dobj(snapped-16, lines-18)
conj:and(shut-8, closed-20)
ccomp(said-40, closed-20)
dobj(closed-20, airports-21)
cc(shut-8, and-22)
conj:and(shut-8, forced-23)
ccomp(said-40, forced-23)
dobj(forced-23, thousands-24)
nsubj(sleep-28, thousands-24)
nsubj(walk-33, thousands-24)
case(people-26, of-25)
nmod:of(thousands-24, people-26)
mark(sleep-28, to-27)
xcomp(forced-23, sleep-28)
case(offices-31, in-29)
nmod:poss(offices-31, their-30)
nmod:in(sleep-28, offices-31)
cc(sleep-28, or-32)
xcomp(forced-23, walk-33)
conj:or(sleep-28, walk-33)
dobj(walk-33, home-34)
case(night-37, during-35)
det(night-37, the-36)
```

Ausgabe-Datei – Teil 3 III

```
nmod:during(walk-33, night-37)
nsubj(said-40, officials-39)
root(ROOT-0, said-40)
nmod:tmod(said-40, today-41)
```

Notationen im Stanford Parser

- Menge der POS-Tags und -Kategorien
  - Englisch – aus Penn Treebank
  - Chinesisch – aus Penn Chinese Treebank
  - Deutsch – aus NEGRA Korpus
- Menge der grammatischen Funktionen
  - Universal Dependecies
    `http://universaldependencies.org/`
  - ältere, sprachspezifische Formalismen auch noch verfügbar

Wichtige Dependenzen I

- **`nsubj`**: nominal subject
  - Subjekt in einer Phrase
  - Hauptwort ist nicht unbedingt ein Verb (z.B. *The flower is blue* - **`nsubj(blue,flower)`**)
- **`nsubjpass`**: passive nominal subject
  - Subjekt in einer Phrase mit dem Verb im Passiv

Wichtige Dependenzen II

- `dobj`: direct object
  - Direktes Objekt in einer Phrase
  - Immer im Akkusativ
- `iobj`: indirect object
  - Indirektes Objekt in einer Phrase
  - Immer im Dativ

Wichtige Dependenzen III

- `advmod`: adverbial modifier
  - Adverbialer Modifikator
- `amod`: adjectival modifier
  - Adjektivaler Modifikator
- `nmod`: nominal modifier
  - Nominaler Modifikator

Parser-Ausgabe exportieren

- Man kann Konstituentenparsebäume in das CoNLL-Format exportieren
- Option: `-conllx`
- Mit `-keepPunct` wird die Interpunktion behalten
- Beispiele:

```
:~$ java edu.stanford.nlp.trees.UniversalEnglishGrammatical\
Structure -treeFile file.tree -collapsedTree -conllx -keepPunct
```

Weitere Stanford NLP Tools

Stanford bietet nicht nur den Parser

- POS Tagger
  `http://nlp.stanford.edu/software/tagger.shtml`
- Named Entity Recognizer
  `http://nlp.stanford.edu/software/CRF-NER.shtml`
- Word Segmenter
  `http://nlp.stanford.edu/software/segmenter.shtml`
- ...
- zusamengefasst in CoreNLP:
  `http://stanfordnlp.github.io/CoreNLP/`

Übung 10