



Dependenzparsing

Inhalt

- Was ist Dependenzgrammatik?
- Dependenzgrammatik und Phrasenstrukturgrammatik
- Maltparser
- Syntaxnet/Parsey McParseface
- Übung
- Quellen

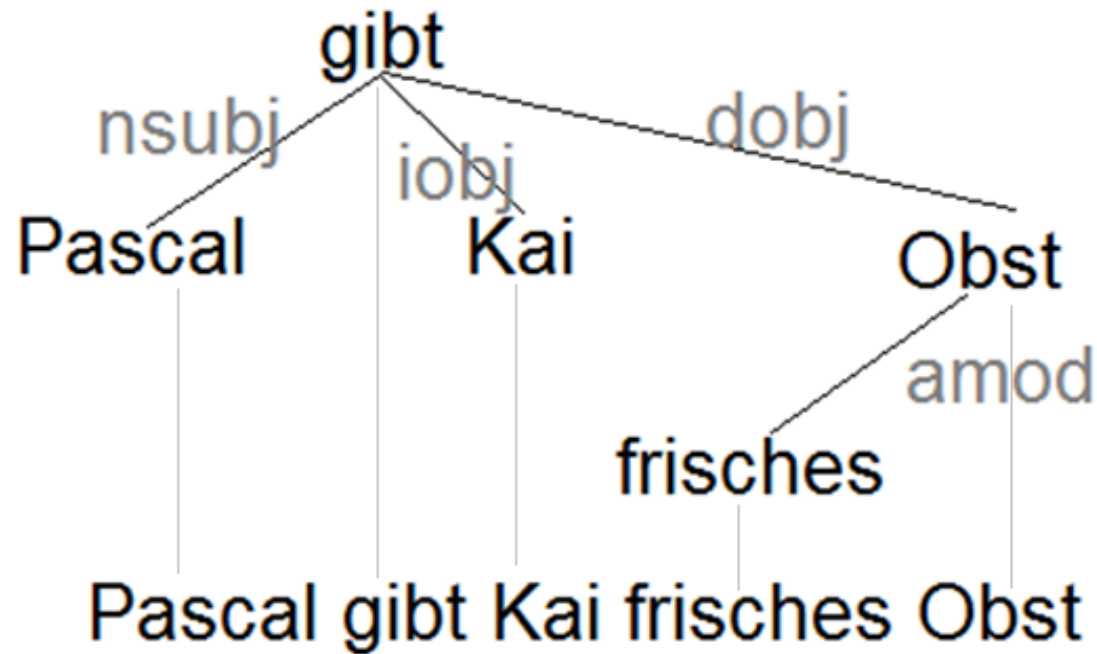
Was ist Dependenzgrammatik?

- Theorie Entwickelt von Lucien Tesnière
- Grundidee stammt schon aus dem Mittelalter von Thomas von Erfurt



Was ist Dependenzgrammatik?

dreistelliges Verb:



Was ist Dependenzgrammatik?

einstelliges Verb:



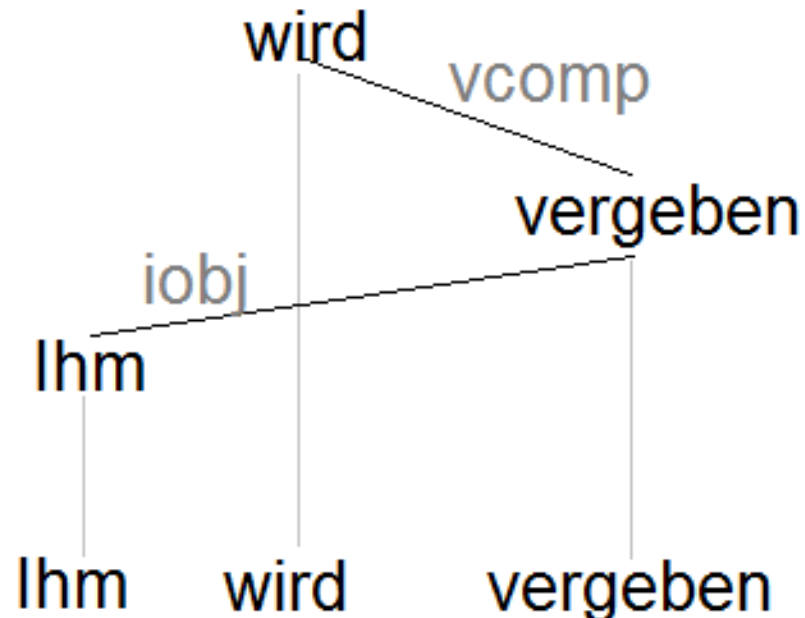
Was ist Dependenzgrammatik?

Satz mit Prädikativ:



Was ist Dependenzgrammatik?

- Die bisherigen Beispiele waren projektive Bäume
- Hier ein Beispiel für einen nicht projektiven Baum:



Dependenzgrammatik und Phrasenstrukturgrammatik

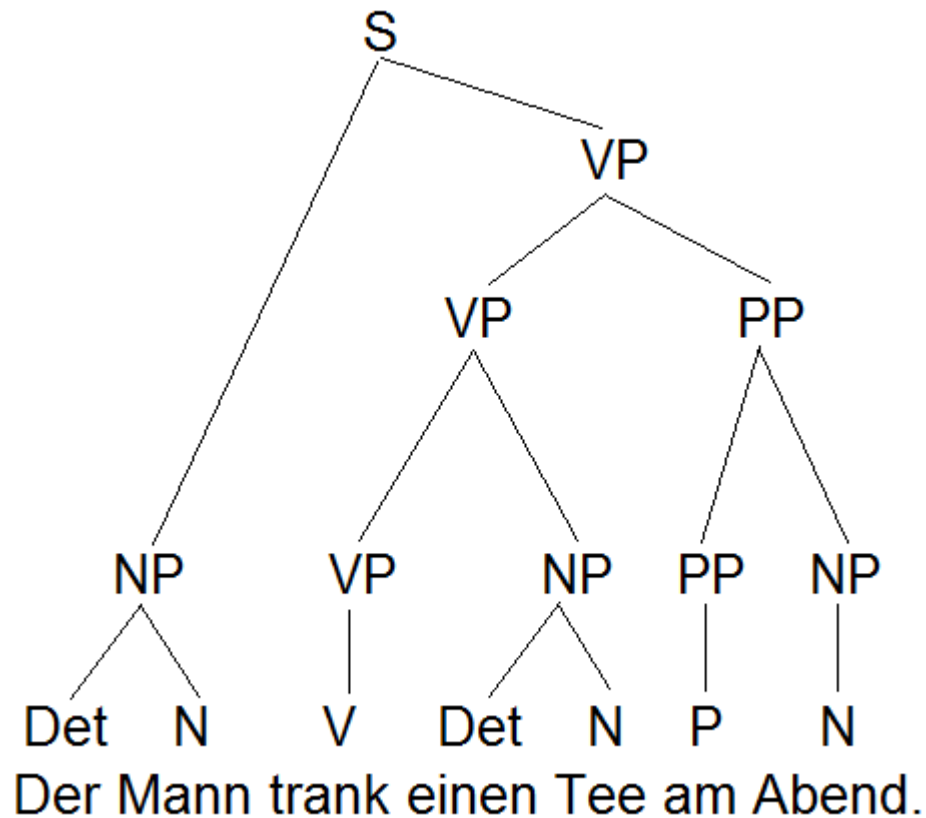
- Tesnières Werk *Éléments de syntaxe structurale* erst 1959 posthum veröffentlicht
- Die Linguistik wurde zu dieser Zeit aber schon von Chomskys Phrasenstrukturgrammatik beherrscht

Dependenzgrammatik und Phrasenstrukturgrammatik

- Die Prager Schule beschäftigte sich mit der Dependenzgrammatik.
- Phrasenstrukturgrammatik funktioniert für das Englische sehr gut.
- Für Sprachen mit freierer Wortstellung nicht ganz so gut

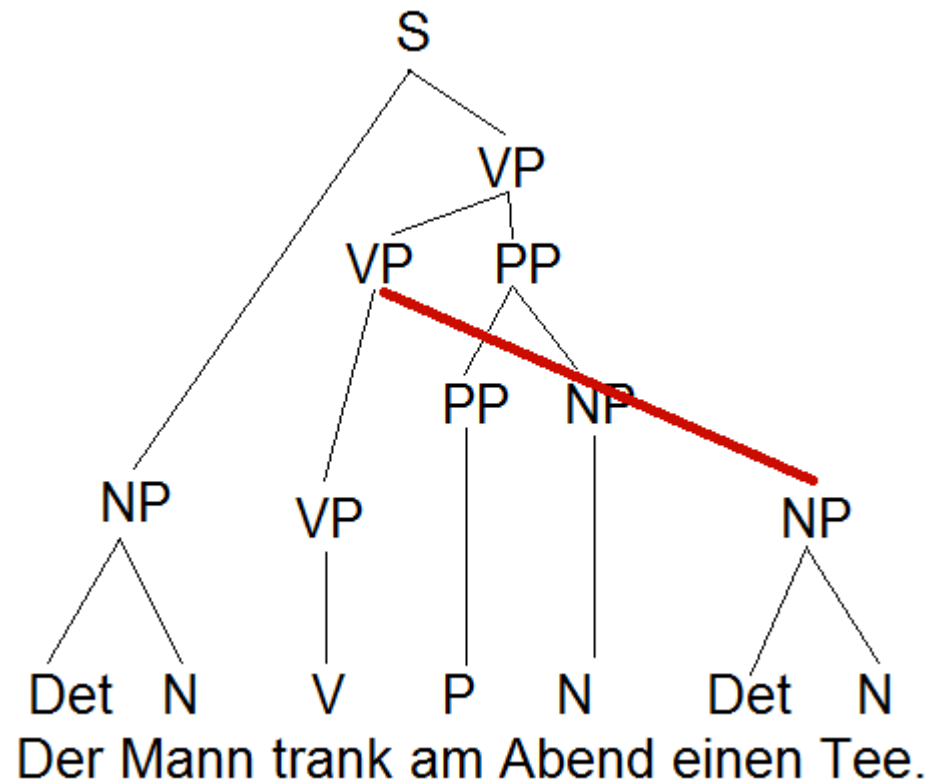
Dependenzgrammatik und Phrasenstrukturgrammatik

Der Mann trank einen Tee gestern Abend.

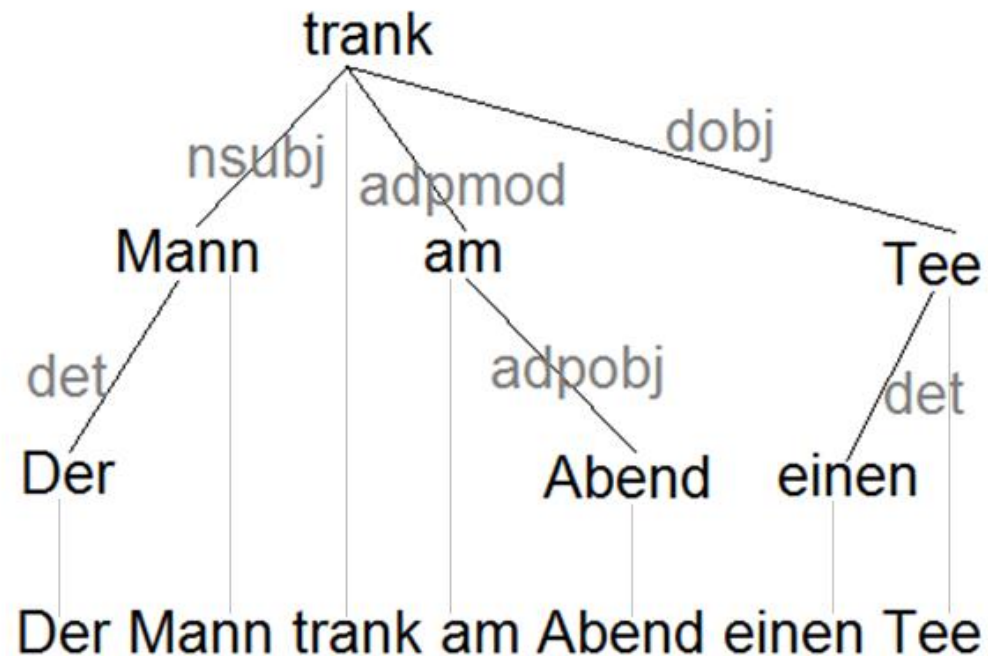


Dependenzgrammatik und Phrasenstrukturgrammatik

Der Mann am Abend einen Tee.



Dependenzgrammatik und Phrasenstrukturgrammatik



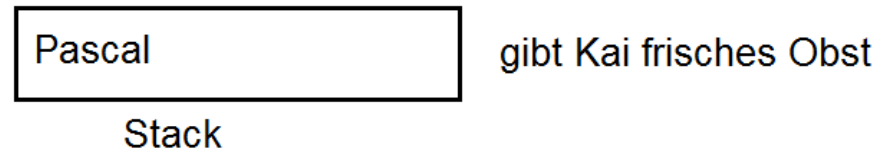
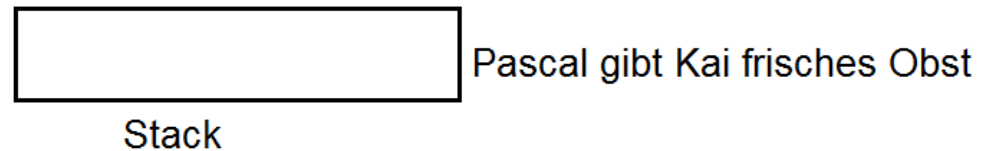
Maltparser

- Durch Dependenzparser erhielt die Dependenzgrammatik neue Bedeutung.
- Maltparser erster gut funktionierender Dependenzparser
- Entwickelt von Joakim Nivre

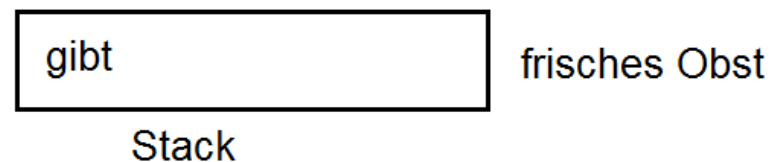
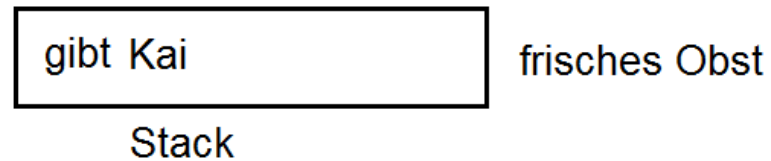


Maltparser

- Shift :

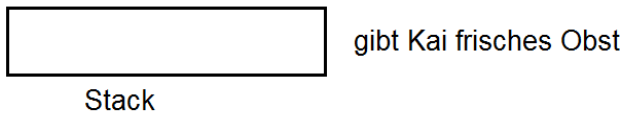
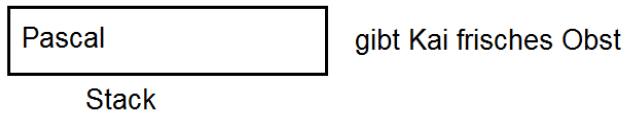


- Reduce:

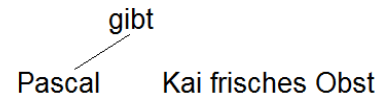


Maltparser

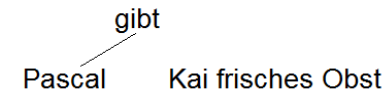
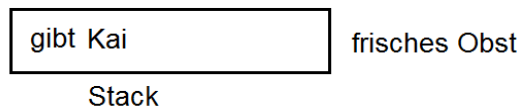
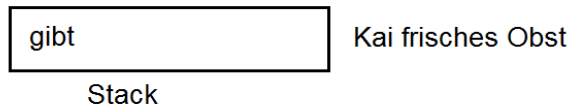
- Left-Arc:



Pascal gibt Kai frisches Obst



- Right-Arc:



Maltparser



Pascal gibt Kai frisches Obst

Pascal gibt Kai frisches Obst

shift

Maltparser

Pascal gibt Kai frisches Obst

Pascal gibt Kai frisches Obst

Left-Arc

Maltparser

gibt Kai frisches Obst

shift

gibt
Pascal Kai frisches Obst

Maltparser

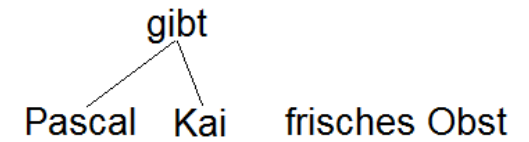
gibt Kai frisches Obst

Pascal ^{gibt} Kai frisches Obst

Right-Arc

Maltparser

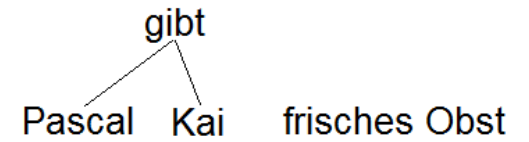
gibt Kai frisches Obst



reduce

Maltparser

frisches Obst

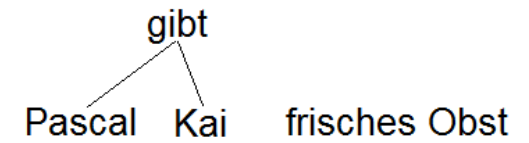


shift

Maltparser

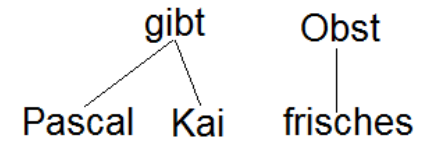
gibt frisches

Obst



Left-Arc

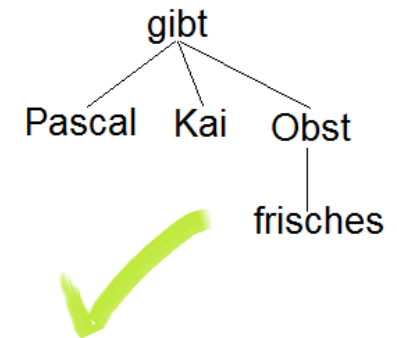
Maltparser



Right-Arc

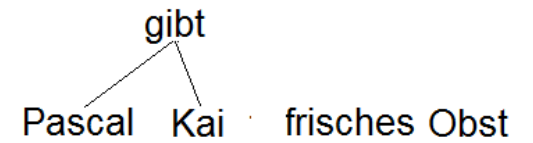
Maltparser

gibt Obst



Maltparser

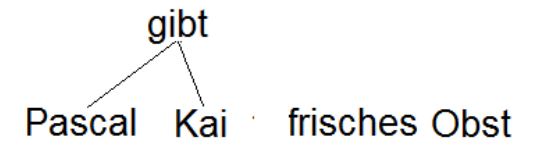
gibt Kai frisches Obst



reduce

Maltparser

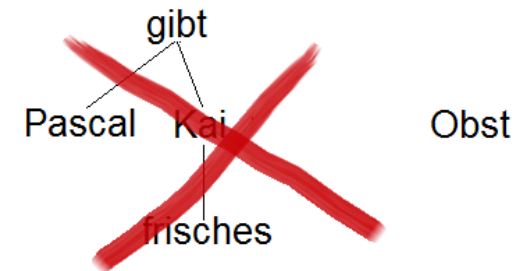
gibt Kai frisches Obst



Right-Arc

Maltparser

gibt Kai frisches Obst



Maltparser

- Ein zuvor trainierter Klassifikator entscheidet über Parsingaktionen
- Features zum Entscheiden sind zum Beispiel POS-Tags
- Trainiert wird der Klassifikator auf Baumbanken
- Dem Parser liegt keine Grammatik zugrunde!

Syntaxnet/ParseyMcParseface



Syntaxnet/Parsey McParseface



6	CESSNOCK	SCR - 8,16,17	JOCKEY	NSW	S-TA
INTERIM DIV				3.80	3.20
2	HORSEY MCHORSEFACE		KEARNEY LATHAM	1.90	1.70
5	THE BIG DREAMER		GRANT BUCKLEY	2.50	2.80

Syntaxnet/Parsey McParseface

Syntaxnet:

- Framework für neuronale Netze
- Von Google entwickelt und der Öffentlichkeit zur Verfügung gestellt

Parsey McParseface:

- Ein von Google mit Syntaxnet trainierter Abhängigkeitsparser
- Mithilfe von neuronalen Netzen verbesserte Variante des Maltparsers

Syntaxnet/Parsey McParseface

Sprache	Anzahl der Tokens	Maltparser	Syntaxnet
Alt Griechisch	206000	67,8%	73,2%
Chinesisch	123000	67,7%	71,2%
Tschechisch	1503000	80,3%	85,9%
Englisch	254000	79,9%	80,38%
Finnisch	181000	74,2%	79,6%
Hebräisch	115000	76,2%	78,7%
Kasachisch	4000	44,4%	44,0%
Tamil	8000	58,8%	55,4%

Prozentzahlen geben an, wie viel Prozent der Wörter mit dem richtigen Vaterknoten verbunden wurden und mit der richtigen Dependenzrelation gelabelt wurden.

Übung

Parsey McParseface benutzen:

echo „This is a test.“ |syntaxnet/demo.sh

```
echo 'Bob brought the pizza to Alice.' | syntaxnet/demo.sh
```

```
Input: Bob brought the pizza to Alice .
```

```
Parse:
```

```
brought VBD ROOT
```

```
+-- Bob NNP nsubj
```

```
+-- pizza NN dobj
```

```
| +-- the DT det
```

```
+-- to IN prep
```

```
| +-- Alice NNP pobj
```

```
+-- . . punct
```

Übung

Baumbanken für weitere Sprachen:

<http://universaldependencies.org/>

(Baumbanken für das Französische, Deutsche, Spanische, Schwedische und Koreanische stehen schon im Dockercontainer zur Verfügung.)

Übung

- `cd universal_treebanks_v1.0`
- `cd de`
- `mv de-universal-train.conll ..`
- `cd ..`
- `mv de-universal-train.conll ..`
- `cd ..`
- `mv de-universal-train.conll maltparser-1.9.1`
- `cd maltparser-1.9.1`

Übung

Trainieren von Maltparser:

```
java -Xmx6g -jar maltparser-1.9.1.jar -c test -i de-universal-train.conll -m learn
```

hinter -c steht, wie das neue Modell heißen soll. Die Modellendung .mco wird weggelassen

gibt java mehr Arbeitsspeicher

```
root@b900f47b54ec:/opt/tensorflow/syntaxnet/maltparser-1.9.1# java -Xmx6g -jar maltparser-1.9.1.jar -c test -i de-universal-train.conll -m learn
-----
MaltParser 1.9.1
-----
MALT (Models and Algorithms for Language Technology) Group
Uaxjo University and Uppsala University
Sweden
-----

Started: Tue Dec 12 17:01:20 UTC 2017
Transition system : Arc-Eager
Parser configuration : Nivre with allow_root=true, allow_reduce=false and enforce_tree=false
Oracle : Arc-Eager
Data Format : file:///opt/tensorflow/syntaxnet/maltparser-1.9.1/test/conllx.xml

.          1          0s          2MB
.         10          0s          3MB
.        100          0s          5MB
.       1000          2s          6MB
.       2000          3s          8MB
.       2200          3s          7MB

Creating Liblinear model odm0.liblinear.moo
- Read all training instances.
- Train a parser model using LibLinear.
- Optimize the memory usage
java -Xmx6g -jar maltparser-1.9.1.jar -c test i
de-universal-train.conll m learn
- Save the Liblinear model odm0.liblinear.moo
Learning time: 00:00:13 (13492 ms)
Finished: Tue Dec 12 17:01:34 UTC 2017
```

hinter -m steht der Modus. Entweder learn zum Modell Trainieren oder parse zum Parsen

hinter -i steht die Inputdatei. Im Falle des Lernmodus muss das eine annotierte Baumbank sein.

Übung

Eingabe Datei für Maltparser erstellen:

- `cd ..`
- `echo 'Das ist ein Test.' | cmd/tree-tagger-german | python convert.py > testdatei.conll`
- `mv testdatei.conll maltparser-1.9.1`
- `cd maltparser-1.9.1`

Übung

Parsen mit Maltparser:

```
java -jar maltparser-1.9.1.jar -c test -i testdatei.conll -o testergebnis.conll -m parse
```

Hinter -c steht das Modell, das zum Parsen benutzt werden soll. Wieder ohne die Endung .mco

Im Parsemodus muss die Inputdatei eine vorbereitete Datei sein, die maltparser parsen kann.

Hinter -o steht der Name, den die Outputdatei haben soll.

```
root@b900f47b54ec:/opt/tensorflow/syntaxnet/maltparser-1.9.1# java -jar maltparser-1.9.1.jar -c test -i testdatei.conll -o testergebnis.conll -m parse
-----
MaltParser 1.9.1
-----
MALT (Models and Algorithms for Language Technology) Group
Uaxjo University and Uppsala University
Sweden
-----
Started: Tue Dec 12 17:29:17 UTC 2017
Transition system : Arc-Eager
Parser configuration : Nivre with allow_root=true, allow_reduce=false and enforce_tree=false
Data Format : /test/conllx.xml
.          1          1s          12MB
.          1          1s          12MB
Parsing time: 00:00:01 (1041 ms)
Finished: Tue Dec 12 17:29:18 UTC 2017
```

Parsemodus

Übung

- **Format für Eingabe :**
Id Tab Wort Tab Unterstrich Tab POS-
Tag Tab POS-Tag
- Ids beginnen bei jedem neuen Satz
wieder mit 1
- Beispiel:

1	Ich	_	PRON	PRON
2	esse	_	VERB	VERB
3	einen	_	DET	DET
4	Apfel	_	NOUN	NOUN
5	.	_	.	.

Übung

POS-Tags:

- ADJ: Adjektive
- ADP: Adpositionen (Präpositionen und Postpositionen)
- ADV: Adverb
- AUX: Hilfsverben
- CCONJ: Konjunktionen
- DET: Determiner
- INTJ: Interjection
- NOUN: Substantiv
- NUM: Numeral
- PART: Partikel
- PRON: Pronomen
- PROPN: Eigename
- PUNCT: Interpunktionszeichen
- SCONJ: Subjunktion
- SYM: Symbol
- VERB: Verb
- X: andere

Übung

- **Format der Ausgabe:**

hinter der Eingabe ist jetzt die Id des Vaterknotens und die Universal dependency relation notiert

- **Beispiel:**

1	Ich	_	PRON	PRON	_	2	nsubj	_	_
2	esse	_	VERB	VERB	_	0	ROOT	_	_
3	einen	_	DET	DET	_	4	det	_	_
4	Apfel	_	NOUN	NOUN	_	2	dobj	_	_
5	.	_	.	.	_	2	p	_	_

Übung

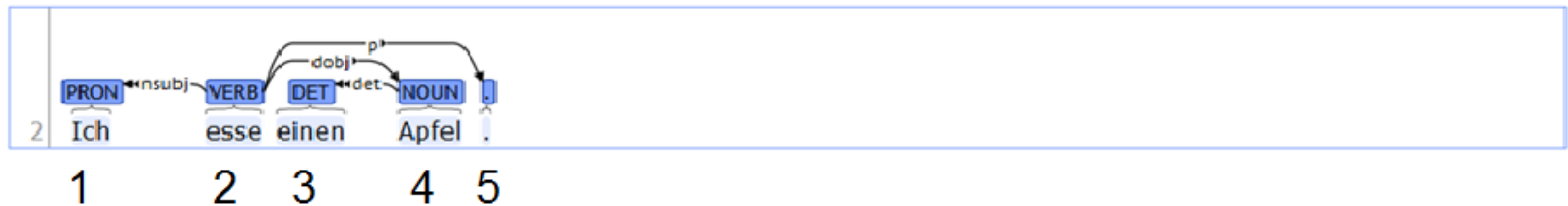
Genaue Erklärung hier:

<http://universaldependencies.org/forma.html>

Übung

Automatische Visualisierung:

<http://spyysalo.github.io/conllu.js/>



Input (editable):

1	Ich	-	PRON	PRON	-	2	nsubj	-	-
2	esse	-	VERB	VERB	-	0	ROOT	-	-
3	einen	-	DET	DET	-	4	det	-	-
4	Apfel	-	NOUN	NOUN	-	2	dobj	-	-
5	.	-	.	.	-	2	p	-	-

Übung

Verschiedene Parsalgorithmen:

features	-F	stringenum	save	Feature model specification	
				<i>nivreeager</i>	Nivre arc-eager default model
				<i>nivrestandard</i>	Nivre arc-standard default model
				<i>covnonproj</i>	Covington non-projective default model
				<i>covproj</i>	Covington projective default model
				<i>stackproj</i>	Stack projective default model
				<i>stackeager</i>	Stack projective default model
				<i>stacklazy</i>	Stack projective default model
				<i>planar</i>	Planar arc-eager default model
				<i>2planar</i>	2-Planar arc-eager default model

Übung

nivre						
allow_root	-nr	bool	true	save	Allow root	
allow_reduce	-ne	bool	false	save	Allow reduce	
enforce_tree	-nt	bool	false	process	Enforce tree constraint	
multiplanar						
planar_root_handling	-prh	enum	normal	save	Root handling	
					relaxed	Root dependents not attached during parsing (attached with default label afterwards).
					normal	Root dependents attached by RightArc transition during parsing (unattached tokens attached with default label afterwards).
planar						
connectedness	-pcon	enum	none	save	Enforce connectedness	
					none	Don't enforce connectedness at all, words whose head the parser doesn't know will be linked to the root node. With this option, the parser will work with planar dependency forests. A forest may be seen as a tree by considering all the roots linked to the dummy root node, but it needn't be planar when seen this way.
					reduceonly	The last node in a connected component cannot be reduced. No restrictions on shift transitions. This option guarantees that the dependency graph obtained counting links to the dummy root node is planar and connected.
					full	Enforce full connectedness by not only not allowing to reduce the last node in a component, but not allowing to shift the last word if the graph is not connected. The produced graph will be connected and planar even without considering the dummy root node.
acyclicity	-pacy	bool	true	save	Enforce acyclicity	
no_covered_roots	-pcov	bool	false	save	Disallow covered roots	
2planar						
reduceonswitch	-2pr	bool	false	save	Force reduce after switch	
covington						
allow_root	-cr	bool	true	save	Allow root	
allow_shift	-cs	bool	false	save	Allow shift transition	

Übung

1. Parsey McParseface mit demo.sh ausprobieren
2. Maltparser auf der deutschen Baumbank trainieren
3. Deutsche Eingabesätze vorbereiten und mit dem Deutschen Modell parsen
4. Modelle mit unterschiedlichen Konfigurationen ausprobieren

<http://www.maltparser.org/options.html>

Quellen

- <http://docplayer.org/34490628-Einfuehrung-in-die-funktionale-dependenzgrammatik-anhand-des-fdg-parsers-fuer-das-englische.html>
- https://en.wikipedia.org/wiki/Lucien_Tesni%C3%A8re#/media/File:Lucien_Tesni%C3%A8re_1954.jpg
- 11.11. 2017 10:12 (Bild von Tensière)
- Heinz J. Weber Dependenzgrammatik Ein Arbeitsbuch Tübingen 1992
- <https://de.wikipedia.org/wiki/Dependenzgrammatik>
- 11.11.2017 11:50
- <http://www.ling.uni-potsdam.de/tcl/ct14/vorlesungen/13%20Dependenzparsing.pdf>
- (Vorlesung von Herrn Professor Koller)
- <http://www.maltparser.org/userguide.html#inout>
- <https://github.com/tensorflow/models/blob/master/research/syntaxnet/Dockerfile>
- <https://stp.lingfil.uu.se/~nivre/>
- (Bild von Joakim Nivre)
- http://www.bing.com/images/search?view=detailV2&ccid=nF%2b5AHQe&id=3F34ACE1F5C5FFABC1B636869A497F769A1E1CA8&thid=OIP.nF-5AHQejn5gl1T4_syLbgEsC3&q=Boaty+McBoatface&simid=608023661411174667&selectedIndex=31&ajaxhist=0
- (Bild von britischen Forschungsschiff)
- <https://eandt.theiet.org/content/articles/2017/06/boaty-mcboatface-completes-first-mission/>
- (Bild von Boaty McBoatface U-Boot)
- <https://github.com/tensorflow/models/tree/master/research/syntaxnet>
- (Syntaxnet github Seite)
- <http://www.maltparser.org/>
- (Maltparser Seite)
- https://www.reddit.com/r/pics/comments/7e0eqt/trainy_mctrainface/
- (Bild von Trainy McTrainface)
- <http://www.wackyraces.net/horsey-mchorseface/>
- (Bild von Horsey McHorseface)

Quellen

- <https://stanfordnlp.github.io/CoreNLP/>
- <http://spyysalo.github.io/conllu.js/>
- Macht Bäume aus dem ConLL Format
- <http://stp.lingfil.uu.se/~nivre/docs/maltparser.pdf>
- <http://universaldependencies.org/u/pos/all.html>
- Erklärungen der POS-Tags für Maltparser
- <http://spyysalo.github.io/conllu.js/>
- Auf dieser Seite kann man sich Conll Format Dateien als gemalte Bäume anzeigen lassen.
- <https://stackoverflow.com/questions/1565388/increase-heap-size-in-java>
- Erklärt, wie man Java mehr Arbeitsspeicher zur Verfügung stellt.
- <https://github.com/ryanmcd/uni-dep-tb>
- <https://pdfs.semanticscholar.org/ab91/5047ce04ff059cb5cbf8ba2b7cfcadceed2e.pdf> (Old School vs. New School: Comparing Transition-Based Parsers with and without Neural Network Enhancement Miryam de Lhoneux, Sara Stymne and Joakim Nivre Department of Linguistics and Philology Uppsala University)
- <https://linguistics.stackexchange.com/questions/6863/how-is-the-f1-score-computed-when-assessing-dependency-parsing>