Corpus Linguistics
(L615)
Syntactic Searching (2)

Markus Dickinson
Department of Linguistics, Indiana University
With much thanks to Detmar Meurers & Sandra Kübler
Motivation for syntax

From last time: “search for a form of werden and a predicative adjective in the same sentence”

Query:

[word = "w[e\mid\u]rde([n\mid$t])?" \mid word = "wirst" \mid word = "wird" \mid word="geworden"]

[]

[tpos = "ADJD"]

within s
Searching in tree structures

A problem for this example:

- Too many examples of adverbial use
  - STTS does not distinguish between adverbs and predicative adjectives
- But this information is present in syntactic annotation:
  - Predicative adjectives project to an adjectival phrase with grammatical function "PRED" (predicate)
Tools for Syntactic Searching

We will look at three tools:

- Tregex & Tsurgeon: http://nlp.stanford.edu/software/tregex.shtml
- MaltEval: http://w3.msi.vxu.se/~jni/malteval/
- TIGERSearch: http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/ (no longer maintained)

We will focus on Tregex

- Other tools are available, including some of the annotation tools mentioned before
Two different programs are available:
  ▶ Tregex: search for patterns
  ▶ Tsurgeon: change trees

Multiple ways to run Tregex:
  1. GUI (graphical user interface)
  2. Command line
  3. Within programs

Tutorial at: http://nlp.stanford.edu/software/tregex/
The_Wonderful_World_of_Tregex.ppt
Tregex

Basic example

> ./tregex.sh 'VP < VBZ < NP' examples/atree

Pattern string:
VP < VBZ < NP

Parsed representation:
Root VP
  and
    < VBZ
    < NP

Reading trees from file(s) examples/atree
(VP (VBZ Try)
  (NP
    (NP (DT this) (NN wine))
    (CC and)
    (NP (DT these) (NNS snails)))

There were 1 matches in total.
Tregex syntax

README-tregex.txt lists an extensive list of patterns which can be matched (see also: http://nlp.stanford.edu/~manning/courses/ling289/Tregex.html)

The basic **node descriptions** include:

- A << B (A dominates B)
- A < B (A immediately dominates B)
- A $ B (A is a sister of B (and not equal to B))
- A .. B (A precedes B)
- A . B (A immediately precedes B)
- ...
Tregex syntax

Regular expressions

Label descriptions can be:

- literal strings
- regular expressions, within forward slashes (/regex/)  
  - This can be very slow!

Notes:

- Special __ string (two underscores) matches any node
- @ is used to match only the basic category, e.g., @NP matches NP-SBJ

```bash
> ./tregex.sh '@NP < (PP << (DT $ NNP)))' \
examples/wsj_0001.mrg
```

- ! is used to negate a symbol (e.g., !NP)
Tregex syntax

Scope of relations

In a chain of relations, all relations are relative to the first node in the chain. For example, \((S < VP < NP)\) means an S over a VP and also over an NP. If instead what you want is an S above a VP above an NP, you should write \(S < (VP < NP)\).

- Note the use of parentheses to delineate node descriptions
- Also: \& can is optional in such cases: \(S < VP \& < NP\)
Chain of relations

Note here how the VP is sister to NP-SBJ

```bash
> ./tregex.sh \
  'VP < (VBZ < is) $ (NP-SBJ <<# (Vinken > NNP))' \
  examples/wsj_0001.mrg
```

Pattern string:
VP < (VBZ < is) $ (NP-SBJ <<# (Vinken > NNP))

Parsed representation:
Root VP
  and
    < VBZ
      < is
      $ NP-SBJ
        <<# Vinken
          > NNP
```

...
Tregex syntax

Boolean operators

Operators over relations: & (and), | (or), ! (negation), ? (optionality)

▶ e.g., (NP < NN | < NNS) matches an NP node over either NN or NNS
▶ e.g., (NP !<< NNP) matches an NP not dominating NNP

To specify operator precedence, [ and ] can group descriptions:

▶ e.g., (NP [< NN | < NNS] & > S) matches an NP:
  ▶ over NN or NNS, and
  ▶ under S
▶ “Without brackets, & takes precedence over |, and equivalent operators are left-associative.”
Tregex example

What does this example from the tutorial do?

\[ NP \leftarrow /NN.?/ > (PP \ll (<\# (IN ![ < of | < on]))) \]
Some command-line options

-C only count matches, don’t print
-w print whole matching tree, not just matching subtree
-f print filename
-i <filename> read search pattern from <filename> rather than the command line
-s print each match on one line, instead of multi-line pretty-printing
-u only print labels of matching nodes, not complete subtrees
-t print terminals only
Tregex GUI

Search pattern: @NP < NNP
Pattern: NP < NNP


Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29.

References

Meurers & Müller 2007
Fronted Particles
Extraposition and Subjacency
Multiple Fronting
Tsurgeon allows one to modify trees, e.g., to delete PPs directly under VPs

Apply delete:
VP < PP=prep
delete prep

NB: the = is used to name nodes, within Tregex & Tsurgeon
Tsurgeon example

examples/atree:

(VP (VP (VBZ Try) (NP (NP (DT this) (NN wine)) (CC and) (NP (DT these) (NNS snails))))) (PUNCT .)

examples/exciseNP:

NP < (NP=np < NNS) < (NP=np1 < NN)

excise np np
excise np1 np1
Tsurgeon example

examples/renameVerb:

VBZ=vbz $ NP

relabel vbz MYVERB

example call & output:

> ./tsurgeon.sh -treeFile examples/atree \
examples/exciseNP examples/renameVerb

(VP
 (VP (MYVERB Try)
   (NP (DT this) (NN wine)
    (CC and)
     (DT these) (NNS snails)))
 (PUNCT .))
Exercises

Let's work through the exercises here:
http://www.ling.ohio-state.edu/~cbrew/inst08/ptb-search.pdf
For dependency relations, you could adapt Tregex or TIGERSearch, but some tools exist for basic searching.

MaltEval is designed for comparison of parse output to a gold standard, but can be used to visualize & search through dependency trees.
java -jar MaltEval.jar -v 1 -g swedish_talbanken05_test.conll

Options include:

- **v** allows one to turn the visualization mode on
- **g** specify the gold standard file
- **s** specify the parsed file

If visualization is turned on, only one of gold/parsed file needs to be specified
MaltEval
Initial visualization
MaltEval

Searching for a dependency label

[Diagram of MaltEval Tree Viewer]

MaltEval

- Search in: Gold-standard
- Search by: Deprel
- Search for: KA
- Result: 8
- Search direction: <<

Gold-standard: /Users/md7/research/dep/data/swedish/talbanken05/test/swedish_talbanken05_test.conll

Sentence:

8 Makars övriga inkomster är B-inkomster och skall som tidigare sambeskattas.
11 Frivillig särskattning tillämpas sista gången i samband med 1971 års taxering.
12 Observera att ansökan härom skall göras senast den 1 juli 1971.
13 Kommunalkatteavdraget slopas.
14 För fysiska personer, dödsbon och familjestiffter slopas rätten att göra avdrag för kommunalkatten fr. o. m. den självdeklaration som skall avlä...
TIGERSearch

(If time ...)

Graphical search interface:

- allows searching in two-dimensional tree structures
- 2 forms of query design:
  - text mode (logical query language):
    
    
    \[
    \text{#n1:\{cat="SIMPX"\} \ast \{word=("werden"|"wird")\}}
    \& \text{#n1 \ast \{pos="ADJD"\}}
    \]
  - graphical mode (construct partial trees)

- searchable relations: direct dominance, dominance (transitive closure), direct linear precedence, linear precedence (transitive closure)
- limited negations
TIGERSearch graphical interface
As with POS annotation, you need to know how phenomena are annotated

- Goal: search for forms of “werden” + adjectival phrase with grammatical function PRED (predicate)
  - Result: 504 sentences
  - Inspection of the first 20 sentences: 10 correct sentences, 9 with 2 clauses, 1 passive sentence

- Restrict search to both occurrences in the same clause
  - Result: 244 sentences
  - Inspection of the first 20 sentences: 18 correct, 2 passives (verb complex not restricted to only head)
Passive Example
Limitations of query tools

- We can only search for phenomena that are present in the annotation
  - Generally, we cannot search for phenomena that involve elided or deleted words, phrases, etc. (unless added)

There are particular phenomenon involving negation which are not possible in TIGERSearch:

- We cannot search for subjectless sentences, e.g. Ihm ist kalt. (To him is cold.)
  - Query: find all trees which do not have an NX node that has ON as function label

- We cannot search for coordinated sentences with a subject gap in the second conjunct
  - query: find all trees which have a KOORD node and a FKONJ with no NX node that has ON as a function
Evasive Strategies

Inchoative werden:

- Assumption: we cannot restrict the verb complex to have only a head (to exclude remaining passives)
- But actually: TIGERSearch allows to restrict a node to have arity one

Determinerless PPs:

- Assumption: we cannot restrict the NX not to have a determiner
- But actually: TIGERSearch allows to mark a direct dominance relation to be either the left corner or the right corner
  - query: find all PXs with NPs where the left corner daughter is not a determiner
With some knowledge of how queries work, we can look at further syntactic constructions

- Fronted particles
- Extraposition & subjacency
- Multiple fronting
Until recently, almost all grammarians assumed that verb particles cannot be fronted, unless it is a predicative particle like *auf* in *aufmachen* (*open make* = ‘to open’).

Sometimes the non-frontability is even used to define particle verbs.

The claim, however, is not correct as the following example illustrates:

(1) *Los ging es schon in dieser Woche.* (taz, 11.10.1995)
    *PART went it already in this week*
    ‘It already started this week.’

References

Meurers & Müller 2007

Fronted Particles: The Issue
Meurers & Müller 2007
Can We Find Such Examples in a Corpus?

- Basic query: Search for verb particle followed by verb.
  - Reasoning: Since particle and verb are usually written as one word when realized clause finally, this query finds examples for particles in the Vorfeld.
  - Pro: Query doesn’t require syntactically annotated corpus.
  - Con: False positives, requiring extensive manual filtering.

- A query based on the German treebank TIGER (Brants et al. 2004):

```
[pos="PTKVZ"] . [pos=finite]
```

This searches for a node with part of speech particle immediately preceding a finite verb.
Refining the Treebank Query

The query [pos="PTKVZ"] . [pos=finit] wrongly matches relative and interrogative clauses such as:

(2) dem Anfang der neunziger Jahre Hohn und Spott which beginning of nineties years mock and sneer zuteil wurde
PART become
‘which were mocked at the beginning of the nineties’

A query with more explicit reference to specifics of the syntactic annotation is required to exclude such examples.
A More Explicit Treebank Query

#s:[cat="S"] > #part:[pos="PTKVZ"] &
#part . [pos=finite] &
#s >@l #leftcorner &
#leftcorner:[pos= ! (prorel | prointer | conjunction)]

1. Search for a sentence node (#s) that dominates a particle (#part)
2. which is adjacent to a finite verb.
3. Search for the left edge of the #s.
4. The part-of-speech of the left edge may not be a relative pronoun, interrogative pronoun or conjunction.

Caveat: The more elaborate a query, the stronger its dependence on the specifics and quality of the syntactic annotation.
A More Explicit Treebank Query

\[
\text{#s:}[\text{cat}='S'] > \text{#part:}[\text{pos}='PTKVZ'] & \\
\text{#part} . [\text{pos}=\text{finite}] & \\
\text{#s} >@l \text{#leftcorner} & \\
\text{#leftcorner:}[\text{pos} = ! (\text{prorel} | \text{prointer} | \text{conjunction})]
\]

1. Search for a sentence node (#s) that dominates a particle (#part)
2. which is adjacent to a finite verb.
3. Search for the left edge of the #s.
4. The part-of-speech of the left edge may not be a relative pronoun, interrogative pronoun or conjunction.

Caveat: The more elaborate a query, the stronger its dependence on the specifics and quality of the syntactic annotation.
A More Explicit Treebank Query

#s:[cat="S"] > #part:[pos="PTKVZ"] &
#part . [pos=finite] &
#s >@l #leftcorner &
#leftcorner:[pos= !(prorel | prointer | conjunction)]

1. Search for a sentence node (#s) that dominates a particle (#part)
2. which is adjacent to a finite verb.
3. Search for the left edge of the #s.
4. The part-of-speech of the left edge may not be a relative pronoun, interrogative pronoun or conjunction.

Caveat: The more elaborate a query, the stronger its dependence on the specifics and quality of the syntactic annotation.
A More Explicit Treebank Query

#s:[cat="S"] > #part:[pos="PTKVZ"] & #part . [pos=finite] & #s @! #leftcorner & #leftcorner:[pos= ! (prorel | prointer | conjunction)]

1. Search for a sentence node (#s) that dominates a particle (#part)
2. which is adjacent to a finite verb.
3. Search for the left edge of the #s.
4. The part-of-speech of the left edge may not be a relative pronoun, interrogative pronoun or conjunction.

Caveat: The more elaborate a query, the stronger its dependence on the specifics and quality of the syntactic annotation.
A More Explicit Treebank Query

#s:[cat="S"] > #part:[pos="PTKVZ"] & #part . [pos=finite] & #s >@l #leftcorner & #leftcorner:[pos= ! (prorel | prointer | conjunction)]

1. Search for a sentence node (#s) that dominates a particle (#part)
2. which is adjacent to a finite verb.
3. Search for the left edge of the #s.
4. The part-of-speech of the left edge may not be a relative pronoun, interrogative pronoun or conjunction.

Caveat: The more elaborate a query, the stronger its dependence on the specifics and quality of the syntactic annotation.
A More Explicit Treebank Query

#s:[cat="S"] > #part:[pos="PTKVZ"] & #part . [pos=finite] & #s >@l #leftcorner & #leftcorner:[pos= !(prorel | prointer | conjunction)]

1. Search for a sentence node (#s) that dominates a particle (#part)
2. which is adjacent to a finite verb.
3. Search for the left edge of the #s.
4. The part-of-speech of the left edge may not be a relative pronoun, interrogative pronoun or conjunction.

Caveat: The more elaborate a query, the stronger its dependence on the specifics and quality of the syntactic annotation.
Extrapolation and Subjacency: The Issue

(3) \[ NP \text{ Many books } [PP \text{ with } [stories t]] t’ \] were sold [that I wanted to read].

(4) \[ NP \text{ Many proofs } [PP \text{ of } [the theorem t]] t’ \] appeared [that I wanted to think about].

Baltin (1981) and Chomsky (1986, p. 40):

- Relative clause cannot be related to \( t \), since Subjacency excludes crossing of more than one barrier.

This view is very common, including recent textbooks (Haegeman 1994, p. 422, Klenk 2003, p. 96, Baltin To Appear).
Extraposition is a Non-Local Dependency

Adjuncts

Extraposition may cross arbitrarily many NP boundaries (cf. Müller 1999, 2004):

(5) Karl hat mir [ein Bild [einer Frau \(\_i\)] gegeben, [die schon lange tot ist]].
    Karl has me a picture of a woman given who already dead is

(6) Karl hat mir [eine Fälschung [des Bildes [einer Frau \(\_i\)]]]
    gegeben, [die schon lange tot ist]].
    Karl has me a forgery of a picture of a woman given who already long dead is

(7) Karl hat mir [eine Kopie [einer Fälschung [des Bildes [einer Frau \(\_i\)]]]
    gegeben, [die schon lange tot ist]].
    Karl has me a copy of a forgery of a picture of a woman given who already long dead is

‘Karl gave me a copy of a forgery of the picture of a woman who has been dead for a long time.’
Extraposition is a Non-Local Dependency
Complements

- Some proposals assume that extraposed adjuncts, such as those in the previous example, are base-generated and that coreference/coindexation is established via special mechanisms (Kiss 2005).
- But it is also possible to extrapose sentential complements:

(8) Ich habe [von [der Vermutung \[\_i\] \] gehört, [daßes
I have of the assumption heard that
Zahlen gibt, die die folgenden Bedingungen
it numbers exist which the following
erfüllen\[\_i\].
conditions satisfy

and such extraposition is equally non-local.
Extrapolation is a Non-Local Dependency

Complements

(9) Ich habe [von [einem Beweis [der Vermutung _i]]] gehört, I have of the proof of the assumption heard [daßes Zahlen gibt, die die folgenden Bedingungen numbers exist which the following erfüllen], conditions satisfy

(10) Ich habe [von [dem Versuch [eines Beweises [der I have of the attempt of a proof of the Vermutung _i]]] gehört, [daßes Zahlen gibt, die die assumption heard that it numbers gives that folgenden Bedingungen erfüllen]], the following conditions satisfy

‘I have heard of the attempt to prove the assumption that there are numbers for which the following conditions hold.’

Selectional restrictions of the matrix head have to be ensured so that one cannot avoid establishing a relation between the governing noun and the extrapolated element.
Can We Find Such Examples in a Corpus?

- A basic query using an unannotated corpus: Search for nouns that take sentential complements + *daß*
  - Reasoning:
    - We know many verbs that govern an object clause.
    - Heavy clauses tend to be extraposed, so looking for nominalizations that actually appear with the complement clause should provide some results.
  - Pro: Query doesn’t require syntactically annotated corpus.
  - Con: False positives, requiring extensive manual filtering.

- Can we improve on this with a query based on a treebank?
A Query Based on the TIGER Treebank

1. Search for an NP node (#xp), that immediately dominates an object clause (OC).
2. #xp is immediately dominated by a NP or PP node.
3. #xp is discontinuous, that is, the object clause is usually extraposed.

We get 12 hits in the 2003 version of the Tiger corpus (40018 sentences).

- We get this result in 1.1 seconds + time for typing the query in comparison to several hours that are needed for the manual approach.
A Query Based on the TIGER Treebank

\[
\text{#xp:[cat="NP"] >OC [ ] \& [cat=("NP"|"PP")]} > \text{#xp \& discontinuous(#xp)}
\]

1. Search for an NP node (#xp), that immediately dominates an object clause (OC).
2. #xp is immediately dominated by a NP or PP node.
3. #xp is discontinuous, that is, the object clause is usually extraposed.

We get 12 hits in the 2003 version of the Tiger corpus (40018 sentences).

- We get this result in 1.1 seconds + time for typing the query in comparison to several hours that are needed for the manual approach.
A Query Based on the TIGER Treebank

```
#xp: [cat=""NP"] > OC [ ] &
[cat=(""NP""|""PP")]> #xp &
discontinuous(#xp)
```

1. Search for an NP node (#xp), that immediately dominates an object clause (OC).
2. #xp is immediately dominated by a NP or PP node.
3. #xp is discontinuous, that is, the object clause is usually extraposed.

We get 12 hits in the 2003 version of the Tiger corpus (40018 sentences).

- We get this result in 1.1 seconds + time for typing the query in comparison to several hours that are needed for the manual approach.
A Query Based on the TIGER Treebank

\[
\begin{align*}
#xp: & [\text{cat=}'NP''] > \text{OC} & [ ] \ &\ & \ & \ & \& \ & \ & \ & \ & \ & \ & \ & \ & \ & \ & \ & \ & \ & \ & \ & \ & \ & \ & \ & \ & \ & \ & \ & \ & \ & \ & \ & \ & dis\text{continuous}(#xp) \\
\end{align*}
\]

1. Search for an NP node (#xp), that immediately dominates an object clause (OC).
2. #xp is immediately dominated by a NP or PP node.
3. #xp is discontinuous, that is, the object clause is usually extraposed.

We get 12 hits in the 2003 version of the Tiger corpus (40018 sentences).

- We get this result in 1.1 seconds + time for typing the query in comparison to several hours that are needed for the manual approach.
Multiple Fronting: The Issue

- German is a V2 Language: Usually only one constituent can be placed in front of the finite verb.

  (11)  
  a. Maria stellt Max Peter vor.  
      Maria introduces Max Peter \textsc{part}  
      ‘Maria introduces Max to Peter.’  
  b. * Maria Max stellt Peter vor.  
      Maria Max introduces Peter \textsc{part}  

- Sometimes this property of German is used as a constituent test: Those elements that can be fronted together form a constituent.
However, sentences with multiple fronted constituents exist (Müller 2003):

(12) [Trocken] [durch die Stadt] kommt man am
dry through the city comes one on
Wochenende auch mit der BVG. (taz berlin, 10.07.1998)
weekends also with the BVG

(13) [Gezielt] [Mitglieder] [im Seniorenbereich] wollen die
targeted members in the senior group wants the
Kendoka allerdings nicht werben. (taz, 07.07.1999)
Kendoka however not advertise

(14) [Wenig] [mit Sprachgeschichte] hat der dritte Beitrag in
little with language history has the third article in
dieser Rubrik zu tun,
this group to do
(Zeitschrift für Dialektologie und Linguistik, 3/2002)
Can We Find Such Examples in a Corpus?

\[
\begin{align*}
\text{#s:[cat='"S"']} & > \text{HD} \text{ # fin:[pos=finite]} &\& \\
\text{#s} & > @l \text{ #sleftedge} &\&
\text{#s} & > \text{#vf1} &\& \text{#vf1} > @l \text{ #sleftedge} &\& \\
\text{#s} & > \text{#vf2} &\& \text{#vf1} \cdot \text{#vf2} &\& \text{#vf2} \cdot \text{#fin}
\end{align*}
\]

1. Search for a sentential node that immediately dominates the finite Verb.
2. #s has a left edge #sleftedge.
3. #s immediately dominates #vf1
4. and #vf1 has the same left edge #sleftedge, that is, #vf1 is the left-most node under #s.
5. #s immediately dominates another node #vf2.
6. #vf1 borders #vf2
7. and #vf2 borders the finite verb.
Can We Find Such Examples in a Corpus?

\[
\begin{align*}
&s: [\text{cat} = "S"] > \text{HD} & & \text{fin: [pos=finite]} & & \&, \\
&s > @l & & \text{sleftedge} & & \&, \\
&s > & & \text{vf1} & & \text{vf1} > @l & & \text{sleftedge} & & \&, \\
&s > & & \text{vf2} & & \text{vf1} . & & \text{vf2} & & \text{vf2} . & & \text{fin}
\end{align*}
\]

1. Search for a sentential node that immediately dominates the finite Verb.
2. \(s\) has a left edge \(\text{sleftedge}\).
3. \(s\) immediately dominates \(\text{vf1}\)
4. and \(\text{vf1}\) has the same left edge \(\text{sleftedge}\), that is, \(\text{vf1}\) is the left-most node under \(s\).
5. \(s\) immediately dominates another node \(\text{vf2}\).
6. \(\text{vf1}\) borders \(\text{vf2}\)
7. and \(\text{vf2}\) borders the finite verb.
Can We Find Such Examples in a Corpus?

#s:[cat=’”S”’] > HD #fin:[pos=finite] & #s > @l #sleftedge & #s > #vf1 & #vf1 > @l #sleftedge & #s > #vf2 & #vf1 . #vf2 & #vf2 . #fin

1. Search for a sentential node that immediately dominates the finite Verb.
2. #s has a left edge #sleftedge.
3. #s immediately dominates #vf1
4. and #vf1 has the same left edge #sleftedge, that is, #vf1 is the left-most node under #s.
5. #s immediately dominates another node #vf2.
6. #vf1 borders #vf2
7. and #vf2 borders the finite verb.
Can We Find Such Examples in a Corpus?

1. Search for a sentential node that immediately dominates the finite Verb.
2. #s has a left edge #sleftedge.
3. #s immediately dominates #vf1
4. and #vf1 has the same left edge #sleftedge, that is, #vf1 is the left-most node under #s.
5. #s immediately dominates another node #vf2.
6. #vf1 borders #vf2
7. and #vf2 borders the finite verb.

\[
\begin{align*}
#s:\text{[cat="S"]} & > \text{HD } #\text{fin:[pos=finite]} & \& \\
#s & > @l #\text{sleftedge} & \\
#s & > #vf1 & #vf1 & > @l #\text{sleftedge} & \\
#s & > #vf2 & #vf1 & > #vf2 & #vf2 & > #\text{fin} & \\
\end{align*}
\]
Can We Find Such Examples in a Corpus?

1. Search for a sentential node that immediately dominates the finite Verb.
2. #s has a left edge #sleftedge.
3. #s immediately dominates #vf1
4. and #vf1 has the same left edge #sleftedge, that is, #vf1 is the left-most node under #s.
5. #s immediately dominates another node #vf2.
6. #vf1 borders #vf2
7. and #vf2 borders the finite verb.
Can We Find Such Examples in a Corpus?

#s:[cat=""S"]  >HD #fin:[pos=finite] &  
#s >@l #sleftedge &  
#s > #vf1 & #vf1 >@l #sleftedge &  
#s > #vf2 & #vf2 >@l #sleftedge &  
#vf1 > @l #sleftedge &  
#vf2 > @l #sleftedge &  
#vf1 . #vf2 & #vf2 . #fin

1. Search for a sentential node that immediately dominates the finite Verb.
2. #s has a left edge #sleftedge.
3. #s immediately dominates #vf1
4. and #vf1 has the same left edge #sleftedge, that is, #vf1 is the left-most node under #s.
5. #s immediately dominates another node #vf2.
6. #vf1 borders #vf2
7. and #vf2 borders the finite verb.
Can We Find Such Examples in a Corpus?

\[
\text{#s:[cat="S"] > HD fin:[pos=finite] & }
\text{#s > @l sleftedge & }
\text{#s > vf1 & vf1 > @l sleftedge & }
\text{#s > vf2 & vf1 \ . vf2 & vf2 \ . fin}
\]

1. Search for a sentential node that immediately dominates the finite Verb.
2. #s has a left edge #sleftedge.
3. #s immediately dominates #vf1
4. and #vf1 has the same left edge #sleftedge, that is, #vf1 is the left-most node under #s.
5. #s immediately dominates another node #vf2.
6. #vf1 borders #vf2
7. and #vf2 borders the finite verb.
Can We Find Such Examples in a Corpus?

#s:[cat="S"] >HD #fin:[pos=finite] &
#s >@l #sleftedge &
#s > #vf1 & #vf1 >@l #sleftedge &
#s > #vf2 & #vf1 . #vf2 & #vf2 . #fin

1. Search for a sentential node that immediately dominates the finite Verb.
2. #s has a left edge #sleftedge.
3. #s immediately dominates #vf1
4. and #vf1 has the same left edge #sleftedge, that is, #vf1 is the left-most node under #s.
5. #s immediately dominates another node #vf2.
6. #vf1 borders #vf2
7. and #vf2 borders the finite verb.
False Positives

- The query unintentionally returns relative and interrogative clauses and parts of coordinations.
- It also finds examples with adverbs such as *inde*{}_{5}, *jedoch*, *wiederum*:

  (15) Hier *wiederum* mangelt es an
      here again lacks it of
      Opferbereitschaft.
      readiness to make sacrifice
Making the Query More Precise

- Since the class of such adverbs is small, one can rule them out explicitly by listing them:

  \[ #vf2: ![ \text{word=} ("'aber'" | ",also'" | "'auch'" | "'allerdings'" | "'dagegen'" | "'freilich'" | "'hingegen'" | "'jedenfalls'" | "'jedoch'" | "'nämlich'" | "'schließlich'" | "'wiederum'") ] \]

- Once we also exclude the relative clauses and coordination sentences, we get: nothing!
References


http://www.cl.uni-bremen.de/~stefan/Pub/mehr-vf-lb.html. 27.03.2013.


Corpus Linguistics

Syntactic Searching

Searching w/ syntax

Tregex
MaltEval
TIGERSearch

Meurers & Müller
2007

Fronted Particles
Extraposition and Subjacency
Multiple Fronting

References


