Motivation for syntax

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

Query:

\[
\begin{align*}
&\text{[word = "w[e$\mid$u]rde([n$\mid$st])?" $\mid$ word = "wirst" $\mid$ word = "wird" $\mid$ word="geworden"]} \\
&\text{[\]}
\end{align*}
\]

\[
\begin{align*}
&\text{[tpos = "ADJD"]}
\end{align*}
\]

within s

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- TIGERSearch: http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/ (no longer maintained)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

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 (DT this) (NN wine))

(CC and)

(NP (DT these) (NNS snails)))

There were 1 matches in total.

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 (DT this) (NN wine))

(CC and)

(NP (DT these) (NNS snails)))

There were 1 matches in total.
Tregex syntax

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 immediately precedes B)
- A .. B (A precedes B)
- ...

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 be optional in such cases:
  S < VP & < NP

Tregex syntax

Boolean operators

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

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 <- /NN.?/ > (PP <<# (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

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/renameVerb:

VBZ=vbz $ NP
relabel vbz MYVERB

eample call & output:

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

(VP (MYVERB Try) (NP (DT this) (NN wine)) (CC and) (NP (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.

MaltEval

Command-line call

```
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

Graphical search interface:
- allows searching in two-dimensional tree structures
- 2 forms of query design:
  - text mode (logical query language):
    ```
    #n1:[cat="SIMPX"] > "werden" & #n1 > [pos="ADJD"]
    ```
  - graphical mode (construct partial trees)
- searchable relations: direct dominance, dominance (transitive closure), direct linear precedence, linear precedence (transitive closure)
- limited negations

TIGERSearch

(If time ...)

Graphical search interface:
- allows searching in two-dimensional tree structures
- 2 forms of query design:
  - text mode (logical query language):
    ```
    #n1:[cat="SIMPX"] > "werden" & #n1 > [pos="ADJD"]
    ```
  - graphical mode (construct partial trees)
- searchable relations: direct dominance, dominance (transitive closure), direct linear precedence, linear precedence (transitive closure)
- limited negations
Searching in tree structures

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)

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 phenomena 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

Fronted Particles: The Issue

Meurers & Müller 2007

With some knowledge of how queries work, we can look at further syntactic constructions

- Fronted particles
- Extraposition & subjacency
- Multiple fronting

Passive Example

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

Fronted Particles:

Meurers & Müller 2007

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.’
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):

  \[
  \text{[pos=\texttt{PTKVZ}]. [pos=\texttt{finite}]}
  \]

  This searches for a node with part of speech particle immediately preceding a finite verb.

A More Explicit Treebank Query

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

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

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

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

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

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.

Refining the Treebank Query

The query \[
\text{[pos=\texttt{PTKVZ}]. [pos=\texttt{finite}]}
\]

This searches for a node with part of speech particle immediately preceding a finite verb.

A More Explicit Treebank Query

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

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

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

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.

References

- Meurers & Müller (2007)
- Brants (2004)
- TIGERSearch
- Tregex
- MaltEval
A More Explicit Treebank Query

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.

Extraposition and Subjacency: The Issue

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

(5) Karl hat mir [ein Bild [einer Frau _]] gegeben, [die schon lange tot ist].
    long dead is

(6) Karl hat mir [eine Fälschung [des Bildes [einer Frau _]]]
    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
    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 a picture of a woman who has been dead for a long time.'

References

Meurers & Müller 2007

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 [dem Versuch eines Beobachters [der
    I have of the attempt of a proof of the
    assumption that it numbers exist which the following
    erfüllen]],
    conditions satisfy
    such extraposition is equally non-local.
Can We Find Such Examples in a Corpus?

- A basic query using an unannotated corpus:
  Search for nouns that take sentential complements + daβ
  - 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

\[
\text{xp}:\text{[cat\,'NP']}>\text{OC} \land \text{[cat\,'(NP'|'PP')]}>\text{xp} \land \text{discontinuous(xp)}
\]

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.

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 PART
  ‘Maria introduces Max to Peter.’
b. * Maria Max stellt Peter vor.
  Maria Max introduces Peter PART

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

1. If the sentence node #s immediately dominates another node #vf2.
2. #vf1 borders #vf2.
3. 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 > #v1 & #v1 > @l #sleftedge & #s > #v2 & #v1 . #v2 & #v2 . #fin

1. Search for a sentential node that immediately dominates the finite Verb.
2. #s has a left edge #sleftedge.
3. #s immediately dominates #v1
4. and #v1 has the same left edge #sleftedge, that is, #v1 is the left-most node under #s.
5. #s immediately dominates another node #v2.
6. #v1 borders #v2.
7. and #v2 borders the finite verb.

References

Meurers & Mülter 2007
Mülter 2003
Tigges 2005
Tigges 2007
Vogel 2007
Vogel 2007
Vogel 2007
Vogel 2007
Vogel 2007

However, sentences with multiple fronted constituents exist (Müller 2003):

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

13. [Gezielt] [mit] Sprachgeschichte] hat der dritte Beitrag in little with language, history has the third article in this Rubrik zu tun, this group to do

(Zeitschrift für Dialektologie und Linguistik, 3/2002)
Can We Find Such Examples in a Corpus?

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

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?

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

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.

False Positives

- The query unintentionally returns relative and interrogative clauses and parts of coordinations.
- It also finds examples with adverbs such as indes, jedoch, wiederum:

\[(15) \text{Hier wiederum mangelt es an readiness to make sacrifice}\]

Making the Query More Precise

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

\[
\text{#vf2: [\text{pos}=\text{adv}}, (\"\text{"aber}\" | \"\text{"also}\" | \"\text{"auch}\" | \"\text{"allerdings}\" | \"\text{"dagegen}\" | \"\text{"freilich}\" | \"\text{"hingegen}\" | \"\text{"jedenfalls}\" | \"\text{"jedoeh}\" | \"\text{"nämlich}\" | \"\text{"schließlich}\" | \"\text{"wiederum}\")]\}
\]

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