Corpus Linguistics
(L415/L615)
Syntactic Searching

Markus Dickinson

Department of Linguistics, Indiana University
(With thanks to Detmar Meurers & Sandra Kübler)
Motivation for syntax

German example: “search for a form of werden and a predicative adjective in the same sentence”

Query:

[word = "w[e\midu]rde([n\midt])?" \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 two tools:

- MaltEval (for dependencies): http://www.maltparser.org/malteval.html
- TIGERSearch (for general constituency trees, including discontinuities): http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/
  - Slides have notes on TIGERSearch, but we will not have time to go over them

We will focus mainly on Tregex

- Other tools are available, including some of the annotation tools mentioned before
Tregex

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

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

- ! is used to negate a symbol (e.g., !NP)
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: & is optional in such cases:
  \(S < VP & < NP\)
Chain of relations

Note here how the VP is sister to NP-SBJ

```
> ./tregex.sh \
  'VP < (VBZ < is) $ (NP-SBJ <<# (Vinken > NNP))' \
  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.”
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
./run-tregex-gui.command

From file: /Users/md7/research/tools/stanford-tregex-2012-11-11/examples/ws1_0001.mrg

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

Tsurgeon allows one to modify trees, e.g., to delete PPs directly under VPs

Apply delete:

\[
\text{VP} \prec \text{PP}=\text{prep} \\
\text{delete prep}
\]

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

rlabel 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://nlp.stanford.edu/manning/courses/ling289/
TreePatternMatchingPrograms.pdf
MaltEval

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

- Tip: download the -dist file and work the commands with lib/MaltEval.jar
- Tip: you can get some free data to play with at: http://ilk.uvt.nl/conll/free_data.html
MaltEval
Command-line call

java -jar MaltEval.jar -v 1 -g swedish_talbanken05_test.conll

Options include:
- \texttt{-v} allows one to turn the visualization mode on
- \texttt{-g} specify the gold standard file
- \texttt{-s} specify the parsed file
  If visualization is turned on, only one of gold/parsed file needs to be specified
MaltEval
Initial visualization

<ROOT> Individuell beskattning av arbetsinkomster

1 Individuell beskattning av arbetsinkomster
2 Genom skattereformen införs individuell beskattning (särbeskattning) av arbetsinkomster.
3 Det innebär bl. a. att endast en skatteskala kommer att finnas för beräkning av statlig inkomstskatt.
4 Den blir gemensam för alla inkomsttagare oavsett civilstånd.
5 Den gäller även för oskifta dödsbo och familjestiftelser.
6 De inkomster som på detta sätt beskattas individuellt kallas A-inkomster.
7 Dessa inkomster är inkomst av tjänst - lön, pension, livranta, undantagsförmåner och övrig tjänsteinkomst. (Undantag: periodiskt understöd etc...)
8 Makars övriga inkomster är B-inkomster och skall som tidigare sambesiktas.
9 Sådana B-inkomster är t. ex. inkomst av kapital, tillfällig förvärvsverksamhet, periodiskt understöd.
11 Frivillig särbeskattning tillämpas sista gången i samband med 1971 års taxering.
12 Observera att ansökan härom skall göras senast den 1 juli 1971.
13 Kommunaliskatteavdraget (slopas)
14 För fysiska personer, dödsbo och familjestiftelser (slopas) rätten att göra avdrag för kommunalskattefr. o. m. den självdeklaration som skall avlå...
16 Förärsavdragen ändras
17 Förärsavdrag kan medges från A-inkomst, om det finns hemmavaruande barn under 16 år.
18 Sådana avdrag medges på sätt framgår av nedanstående tablå.
19 Barns ålder räknas efter förhållandet den 1 november under inkom斯塔tet.
MaltEval
Searching for a dependency label
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"] >* [word=("werden"|"wird")]
    & \#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 graphical interface
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)
Passive Example

Corpus Linguistics
Syntactic Searching

Searching w/ syntax
Tregex
MaltEval
TiGERSearch
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. "Hm 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