LING/C SC 581:
Advanced Computational Linguistics

Lecture 21

Today's Topic

- Copy the full Penn Treebank (PTB)corpus from the course website
  - instructions given out in Panopto and in class (not on class slides!)
- Homework 9: install tregex
  - we'll be using the software called `tregex` to search the treebanks
  - it's written in Java and requires a Java runtime environment

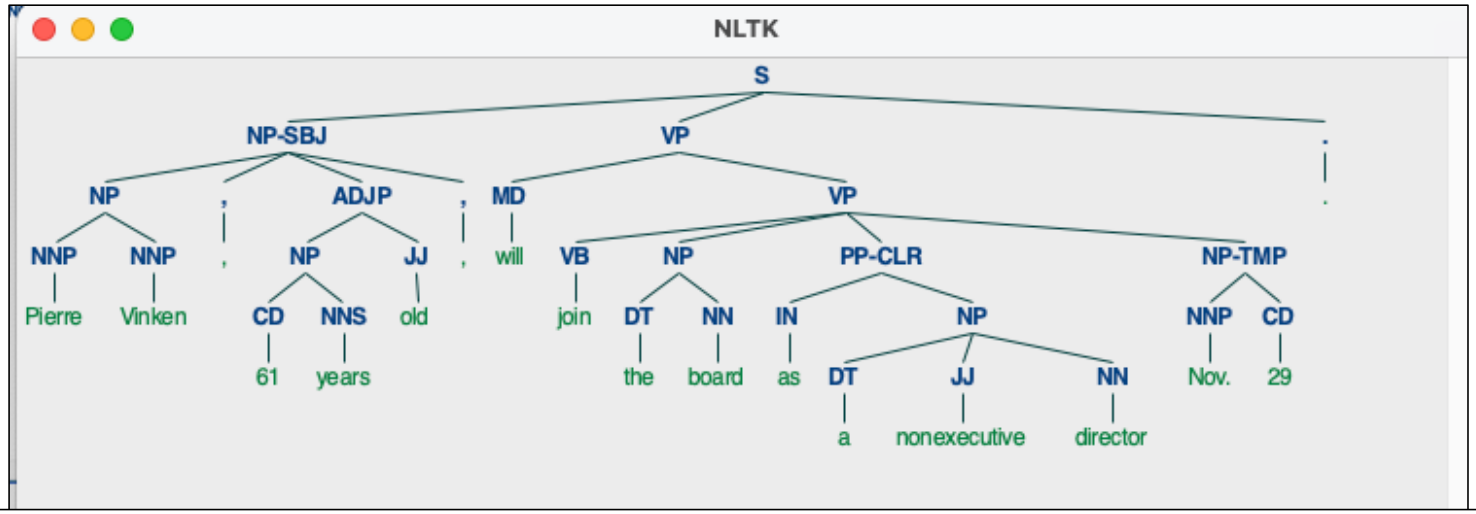
nltk

nltk: corpus treebank

- *a 100,000 word subset of the PTB* (*not the full thing*!)

```
>>> import nltk
>>> from nltk.corpus import treebank
>>> t = treebank.parsed_sents()
>>> t[0]
Tree('S', [Tree('NP-SBJ', [Tree('NP', [Tree('NNP', ['Pierre']), Tree('NNP',
['Vinken'])]), Tree(',', [',']), Tree('ADJP', [Tree('NP', [Tree('CD', ['61']),
Tree('NNS', ['years'])]), Tree('JJ', ['old'])]), Tree(',', [','])]), Tree('VP',
[Tree('MD', ['will']), Tree('VP', [Tree('VB', ['join']), Tree('NP', [Tree('DT',
['the']), Tree('NN', ['board'])]), Tree('PP-CLR', [Tree('IN', ['as']), Tree('NP',
[Tree('DT', ['a']), Tree('JJ', ['nonexecutive']), Tree('NN', ['director'])])]),
Tree('NP-TMP', [Tree('NNP', ['Nov.']), Tree('CD', ['29'])])])]), Tree('.', ['.'])])
>>> t[0].draw()
>>> len(t)
3914
```

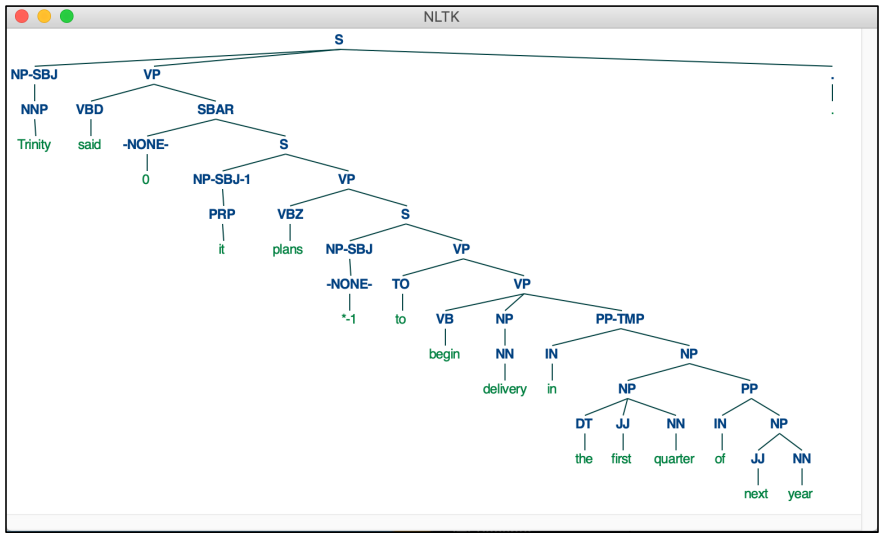
nltk




```
Tree('S', [Tree('NP-SBJ', [Tree('NP', [Tree('NNP', ['Pierre']), Tree('NNP',
['Vinken'])]), Tree(',', [',']), Tree('ADJP', [Tree('NP', [Tree('CD', ['61']),
Tree('NNS', ['years'])]), Tree('JJ', ['old'])]), Tree(',', [','])]), Tree('VP',
[Tree('MD', ['will']), Tree('VP', [Tree('VB', ['join']), Tree('NP', [Tree('DT',
['the']), Tree('NN', ['board'])]), Tree('PP-CLR', [Tree('IN', ['as']), Tree('NP',
[Tree('DT', ['a']), Tree('JJ', ['nonexecutive']), Tree('NN', ['director'])])]),
Tree('NP-TMP', [Tree('NNP', ['Nov.']), Tree('CD', ['29'])])])]), Tree('.', ['.'])])
```

nltk

- `>>> t[-1].draw()`
- The *sample* of the well-known Penn Treebank (PTB) Wall Street Journal (WSJ) corpus includes:
  - 3,914 parsed sentences
  - 49,000+ parsed sentences in the full corpus



nltk

- Words:

```
>>> w = treebank.words()
>>> len(w)
100676
>>> w
['Pierre', 'Vinken', ',', '61', 'years', 'old', ',', ...]
>>> tw = treebank.tagged_words()
>>> tw
[('Pierre', 'NNP'), ('Vinken', 'NNP'), (',', ','), ...]
>>> tw[:10]
[('Pierre', 'NNP'), ('Vinken', 'NNP'), (',', ','), ('61', 'CD'), ('years',
'NNS'), ('old', 'JJ'), (',', ','), ('will', 'MD'), ('join', 'VB'), ('the',
'DT')]
```

nltk: Corpus Readers

- http://www.nltk.org/howto/corpus.html#parsed-corpora
  - The NLTK data package includes a 10% sample of the Penn Treebank (in `treebank`), as well as the Sinica Treebank (in `sinica_treebank`).

- Reading the Penn Treebank (Wall Street Journal sample):

```
>>> from nltk.corpus import treebank
1. treebank.fileids()
2. treebank.words(fileid))
3. treebank.tagged_words(fileid))
4. treebank.parsed_sents(fileid)
```

nltk: Corpus Readers

`treebank.words(`*`fileid`*`)`

```
>>> len(treebank.words('wsj_0003.mrg'))
782
>>> treebank.words('wsj_0003.mrg')
['A', 'form', 'of', 'asbestos', 'once', 'used', '*', ...]
>>> list(treebank.words('wsj_0003.mrg'))[:200]
['A', 'form', 'of', 'asbestos', 'once', 'used', '*', '*', 'to', 'make', 'Kent', 'cigarette', 'filters', 'has', 'caused
', 'a', 'high', 'percentage', 'of', 'cancer', 'deaths', 'among', 'a', 'group', 'of', 'workers', 'exposed', '*', 'to',
'it', 'more', 'than', '30', 'years', 'ago', ',', 'researchers', 'reported', '0', '*T*-1', '.', 'The', 'asbestos', 'fib
er', ',', 'crocidolite', ',', 'is', 'unusually', 'resilient', 'once', 'it', 'enters', 'the', 'lungs', ',', 'with', 'ev
en', 'brief', 'exposures', 'to', 'it', 'causing', 'symptoms', 'that', '*T*-1', 'show', 'up', 'decades', 'later', ',',
'researchers', 'said', '0', '*T*-2', '.', 'Lorillard', 'Inc.', ',', 'the', 'unit', 'of', 'New', 'York-based', 'Loews',
 'Corp.', 'that', '*T*-2', 'makes', 'Kent', 'cigarettes', ',', 'stopped', 'using', 'crocidolite', 'in', 'its', 'Micron
ite', 'cigarette', 'filters', 'in', '1956', '.', 'Although', 'preliminary', 'findings', 'were', 'reported', '*-2', 'mo
re', 'than', 'a', 'year', 'ago', ',', 'the', 'latest', 'results', 'appear', 'in', 'today', "'s", 'New', 'England', 'Jo
urnal', 'of', 'Medicine', ',', 'a', 'forum', 'likely', '*', 'to', 'bring', 'new', 'attention', 'to', 'the', 'problem',
 '.', 'A', 'Lorillard', 'spokewoman', 'said', ',', '``', 'This', 'is', 'an', 'old', 'story', '.', 'We', "'re", 'talkin
g', 'about', 'years', 'ago', 'before', 'anyone', 'heard', 'of', 'asbestos', 'having', 'any', 'questionable', 'properti
es', '.', 'There', 'is', 'no', 'asbestos', 'in', 'our', 'products', 'now', '.', "''", 'Neither', 'Lorillard', 'nor', '
the', 'researchers', 'who', '*T*-3', 'studied', 'the', 'workers', 'were', 'aware', 'of', 'any', 'research', 'on', 'smo
kers', 'of', 'the', 'Kent', 'cigarettes', '.']
>>>
```

nltk: Corpus Readers

```
treebank.fileids()
```

```
>>> treebank.fileids()
['wsj_0001.mrg', 'wsj_0002.mrg', 'wsj_0003.mrg', 'wsj_0004.mrg', 'wsj_0005.mrg', 'wsj_0006.mrg', 'wsj_0007.mrg', 'wsj_0008.mrg',
 'wsj_0009.mrg', 'wsj_0010.mrg', 'wsj_0011.mrg', 'wsj_0012.mrg', 'wsj_0013.mrg', 'wsj_0014.mrg', 'wsj_0015.mrg', 'wsj_0016.mrg',
 'wsj_0017.mrg', 'wsj_0018.mrg', 'wsj_0019.mrg', 'wsj_0020.mrg', 'wsj_0021.mrg', 'wsj_0022.mrg', 'wsj_0023.mrg', 'wsj_0024.mrg',
 'wsj_0025.mrg', 'wsj_0026.mrg', 'wsj_0027.mrg', 'wsj_0028.mrg', 'wsj_0029.mrg', 'wsj_0030.mrg', 'wsj_0031.mrg', 'wsj_0032.mrg',
 'wsj_0033.mrg', 'wsj_0034.mrg', 'wsj_0035.mrg', 'wsj_0036.mrg', 'wsj_0037.mrg', 'wsj_0038.mrg', 'wsj_0039.mrg', 'wsj_0040.mrg',
 'wsj_0041.mrg', 'wsj_0042.mrg', 'wsj_0043.mrg', 'wsj_0044.mrg', 'wsj_0045.mrg', 'wsj_0046.mrg', 'wsj_0047.mrg', 'wsj_0048.mrg',
 'wsj_0049.mrg', 'wsj_0050.mrg', 'wsj_0051.mrg', 'wsj_0052.mrg', 'wsj_0053.mrg', 'wsj_0054.mrg', 'wsj_0055.mrg', 'wsj_0056.mrg',
 'wsj_0057.mrg', 'wsj_0058.mrg', 'wsj_0059.mrg', 'wsj_0060.mrg', 'wsj_0061.mrg', 'wsj_0062.mrg', 'wsj_0063.mrg', 'wsj_0064.mrg',
 'wsj_0065.mrg', 'wsj_0066.mrg', 'wsj_0067.mrg', 'wsj_0068.mrg', 'wsj_0069.mrg', 'wsj_0070.mrg', 'wsj_0071.mrg', 'wsj_0072.mrg',
 'wsj_0073.mrg', 'wsj_0074.mrg', 'wsj_0075.mrg', 'wsj_0076.mrg', 'wsj_0077.mrg', 'wsj_0078.mrg', 'wsj_0079.mrg', 'wsj_0080.mrg',
 'wsj_0081.mrg', 'wsj_0082.mrg', 'wsj_0083.mrg', 'wsj_0084.mrg', 'wsj_0085.mrg', 'wsj_0086.mrg', 'wsj_0087.mrg', 'wsj_0088.mrg',
 'wsj_0089.mrg', 'wsj_0090.mrg', 'wsj_0091.mrg', 'wsj_0092.mrg', 'wsj_0093.mrg', 'wsj_0094.mrg', 'wsj_0095.mrg', 'wsj_0096.mrg',
 'wsj_0097.mrg', 'wsj_0098.mrg', 'wsj_0099.mrg', 'wsj_0100.mrg', 'wsj_0101.mrg', 'wsj_0102.mrg', 'wsj_0103.mrg', 'wsj_0104.mrg',
 'wsj_0105.mrg', 'wsj_0106.mrg', 'wsj_0107.mrg', 'wsj_0108.mrg', 'wsj_0109.mrg', 'wsj_0110.mrg', 'wsj_0111.mrg', 'wsj_0112.mrg',
 'wsj_0113.mrg', 'wsj_0114.mrg', 'wsj_0115.mrg', 'wsj_0116.mrg', 'wsj_0117.mrg', 'wsj_0118.mrg', 'wsj_0119.mrg', 'wsj_0120.mrg',
 'wsj_0121.mrg', 'wsj_0122.mrg', 'wsj_0123.mrg', 'wsj_0124.mrg', 'wsj_0125.mrg', 'wsj_0126.mrg', 'wsj_0127.mrg', 'wsj_0128.mrg',
 'wsj_0129.mrg', 'wsj_0130.mrg', 'wsj_0131.mrg', 'wsj_0132.mrg', 'wsj_0133.mrg', 'wsj_0134.mrg', 'wsj_0135.mrg', 'wsj_0136.mrg',
 'wsj_0137.mrg', 'wsj_0138.mrg', 'wsj_0139.mrg', 'wsj_0140.mrg', 'wsj_0141.mrg', 'wsj_0142.mrg', 'wsj_0143.mrg', 'wsj_0144.mrg',
 'wsj_0145.mrg', 'wsj_0146.mrg', 'wsj_0147.mrg', 'wsj_0148.mrg', 'wsj_0149.mrg', 'wsj_0150.mrg', 'wsj_0151.mrg', 'wsj_0152.mrg',
 'wsj_0153.mrg', 'wsj_0154.mrg', 'wsj_0155.mrg', 'wsj_0156.mrg', 'wsj_0157.mrg', 'wsj_0158.mrg', 'wsj_0159.mrg', 'wsj_0160.mrg',
 'wsj_0161.mrg', 'wsj_0162.mrg', 'wsj_0163.mrg', 'wsj_0164.mrg', 'wsj_0165.mrg', 'wsj_0166.mrg', 'wsj_0167.mrg', 'wsj_0168.mrg',
 'wsj_0169.mrg', 'wsj_0170.mrg', 'wsj_0171.mrg', 'wsj_0172.mrg', 'wsj_0173.mrg', 'wsj_0174.mrg', 'wsj_0175.mrg', 'wsj_0176.mrg',
 'wsj_0177.mrg', 'wsj_0178.mrg', 'wsj_0179.mrg', 'wsj_0180.mrg', 'wsj_0181.mrg', 'wsj_0182.mrg', 'wsj_0183.mrg', 'wsj_0184.mrg',
 'wsj_0185.mrg', 'wsj_0186.mrg', 'wsj_0187.mrg', 'wsj_0188.mrg', 'wsj_0189.mrg', 'wsj_0190.mrg', 'wsj_0191.mrg', 'wsj_0192.mrg',
 'wsj_0193.mrg', 'wsj_0194.mrg', 'wsj_0195.mrg', 'wsj_0196.mrg', 'wsj_0197.mrg', 'wsj_0198.mrg', 'wsj_0199.mrg']
>>>
```

```
>>> len(treebank.fileids())
199
```

nltk: Corpus Readers

`treebank.tagged_words(`*`fileid`*`))`

```
>>> >>> list(treebank.tagged_words('wsj_0003.mrg'))[:100]
[('A', 'DT'), ('form', 'NN'), ('of', 'IN'), ('asbestos', 'NN'), ('once', 'RB'), ('used', 'VBN'), ('*', '-NONE-'), ('*'
, '-NONE-'), ('to', 'TO'), ('make', 'VB'), ('Kent', 'NNP'), ('cigarette', 'NN'), ('filters', 'NNS'), ('has', 'VBZ'), (
'caused', 'VBN'), ('a', 'DT'), ('high', 'JJ'), ('percentage', 'NN'), ('of', 'IN'), ('cancer', 'NN'), ('deaths', 'NNS')
, ('among', 'IN'), ('a', 'DT'), ('group', 'NN'), ('of', 'IN'), ('workers', 'NNS'), ('exposed', 'VBN'), ('*', '-NONE-')
, ('to', 'TO'), ('it', 'PRP'), ('more', 'RBR'), ('than', 'IN'), ('30', 'CD'), ('years', 'NNS'), ('ago', 'IN'), (',', '
,'), ('researchers', 'NNS'), ('reported', 'VBD'), ('0', '-NONE-'), ('*T*-1', '-NONE-'), ('.', '.'), ('The', 'DT'), ('a
sbestos', 'NN'), ('fiber', 'NN'), (',', ','), ('crocidolite', 'NN'), (',', ','), ('is', 'VBZ'), ('unusually', 'RB'), (
'resilient', 'JJ'), ('once', 'IN'), ('it', 'PRP'), ('enters', 'VBZ'), ('the', 'DT'), ('lungs', 'NNS'), (',', ','), ('w
ith', 'IN'), ('even', 'RB'), ('brief', 'JJ'), ('exposures', 'NNS'), ('to', 'TO'), ('it', 'PRP'), ('causing', 'VBG'), (
'symptoms', 'NNS'), ('that', 'WDT'), ('*T*-1', '-NONE-'), ('show', 'VBP'), ('up', 'RP'), ('decades', 'NNS'), ('later',
 'JJ'), (',', ','), ('researchers', 'NNS'), ('said', 'VBD'), ('0', '-NONE-'), ('*T*-2', '-NONE-'), ('.', '.'), ('Loril
lard', 'NNP'), ('Inc.', 'NNP'), (',', ','), ('the', 'DT'), ('unit', 'NN'), ('of', 'IN'), ('New', 'JJ'), ('York-based',
 'JJ'), ('Loews', 'NNP'), ('Corp.', 'NNP'), ('that', 'WDT'), ('*T*-2', '-NONE-'), ('makes', 'VBZ'), ('Kent', 'NNP'), (
'cigarettes', 'NNS'), (',', ','), ('stopped', 'VBD'), ('using', 'VBG'), ('crocidolite', 'NN'), ('in', 'IN'), ('its', '
PRP$'), ('Micronite', 'NN'), ('cigarette', 'NN'), ('filters', 'NNS')]
>>>
```

nltk: Corpus Readers

`treebank.parsed_sents(`*`fileid`*`)`

```
>>> len(treebank.parsed_sents('wsj_0003.mrg'))
30
>>> treebank.parsed_sents('wsj_0003.mrg')[0]
Tree('S', [Tree('S-TPC-1', [Tree('NP-SBJ', [Tree('NP', [Tree('NP', [Tree('DT', ['A']), Tree('NN', ['form'])]), Tree('P
P', [Tree('IN', ['of']), Tree('NP', [Tree('NN', ['asbestos'])])])]), Tree('RRC', [Tree('ADVP-TMP', [Tree('RB', ['once'
])]), Tree('VP', [Tree('VBN', ['used']), Tree('NP', [Tree('-NONE-', ['*'])]), Tree('S-CLR', [Tree('NP-SBJ', [Tree('-NO
NE-', ['*'])]), Tree('VP', [Tree('TO', ['to']), Tree('VP', [Tree('VB', ['make']), Tree('NP', [Tree('NNP', ['Kent']), T
ree('NN', ['cigarette']), Tree('NNS', ['filters'])])])])])])])]), Tree('VP', [Tree('VBZ', ['has']), Tree('VP', [Tree('
VBN', ['caused']), Tree('NP', [Tree('NP', [Tree('DT', ['a']), Tree('JJ', ['high']), Tree('NN', ['percentage'])]), Tree
('PP', [Tree('IN', ['of']), Tree('NP', [Tree('NN', ['cancer']), Tree('NNS', ['deaths'])])]), Tree('PP-LOC', [Tree('IN'
, ['among']), Tree('NP', [Tree('NP', [Tree('DT', ['a']), Tree('NN', ['group'])]), Tree('PP', [Tree('IN', ['of']), Tree
('NP', [Tree('NP', [Tree('NNS', ['workers'])]), Tree('RRC', [Tree('VP', [Tree('VBN', ['exposed']), Tree('NP', [Tree('-
NONE-', ['*'])]), Tree('PP-CLR', [Tree('TO', ['to']), Tree('NP', [Tree('PRP', ['it'])])]), Tree('ADVP-TMP', [Tree('NP'
, [Tree('QP', [Tree('RBR', ['more']), Tree('IN', ['than']), Tree('CD', ['30'])]), Tree('NNS', ['years'])]), Tree('IN',
 ['ago'])])])])])])])])])])]), Tree(',', [',']), Tree('NP-SBJ', [Tree('NNS', ['researchers'])]), Tree('VP', [Tree('V
BD', ['reported']), Tree('SBAR', [Tree('-NONE-', ['0']), Tree('S', [Tree('-NONE-', ['*T*-1'])])])]), Tree('.', ['.'])]
)
>>>
```

Penn POS Tagset

POS Tag	Description	Example
CC	coordinating conjunction	and
CD	cardinal number	1, third
DT	determiner	the
EX	existential there	*there* is
FW	foreign word	d'hoevre
IN	preposition/subordinating conjunction	in, of, like
JJ	adjective	green
JJR	adjective, comparative	greener
JJS	adjective, superlative	greenest
LS	list marker	1)
MD	modal	could, will
NN	noun, singular or mass	table
NNS	noun plural	tables
NNP	proper noun, singular	John
NNPS	proper noun, plural	Vikings

Penn POS Tagset

PDT	predeterminer	*both* the boys
POS	possessive ending	friend*'s*
PRP	personal pronoun	I, he, it
PRP$	possessive pronoun	my, his
RB	adverb	however, usually, naturally, here, good
RBR	adverb, comparative	better
RBS	adverb, superlative	best
RP	particle	give *up*
TO	to	*to* go, *to* him
UH	interjection	uhhuhhuhh
VB	verb, base form	take
VBD	verb, past tense	took
VBG	verb, gerund/present participle	taking
VBN	verb, past participle	taken
VBP	verb, sing. present, non-3d	take
VBZ	verb, 3rd person sing. present	takes

Penn POS Tagset

WDT	wh-determiner	which
WP	wh-pronoun	who, what
WP$	possessive wh-pronoun	whose
WRB	wh-abverb	where, when

Penn Syntax Tagset

- (from The Penn Treebank: An overview, Taylor, Marcus & Santorini)

Table 1.2. The Penn Treebank syntactic tagset

ADJP	Adjective phrase
ADVP	Adverb phrase
NP	Noun phrase
PP	Prepositional phrase
S	Simple declarative clause
SBAR	Subordinate clause
SBARQ	Direct question introduced by *wh*-element
SINV	Declarative sentence with subject-aux inversion
SQ	Yes/no questions and subconstituent of SBARQ excluding *wh*-element
VP	Verb phrase
WHADVP	Wh-adverb phrase
WHNP	Wh-noun phrase
WHPP	Wh-prepositional phrase
X	Constituent of unknown or uncertain category
*	"Understood" subject of infinitive or imperative
0	Zero variant of *that* in subordinate clauses
T	Trace of wh-Constituent

Penn Syntax Tagset

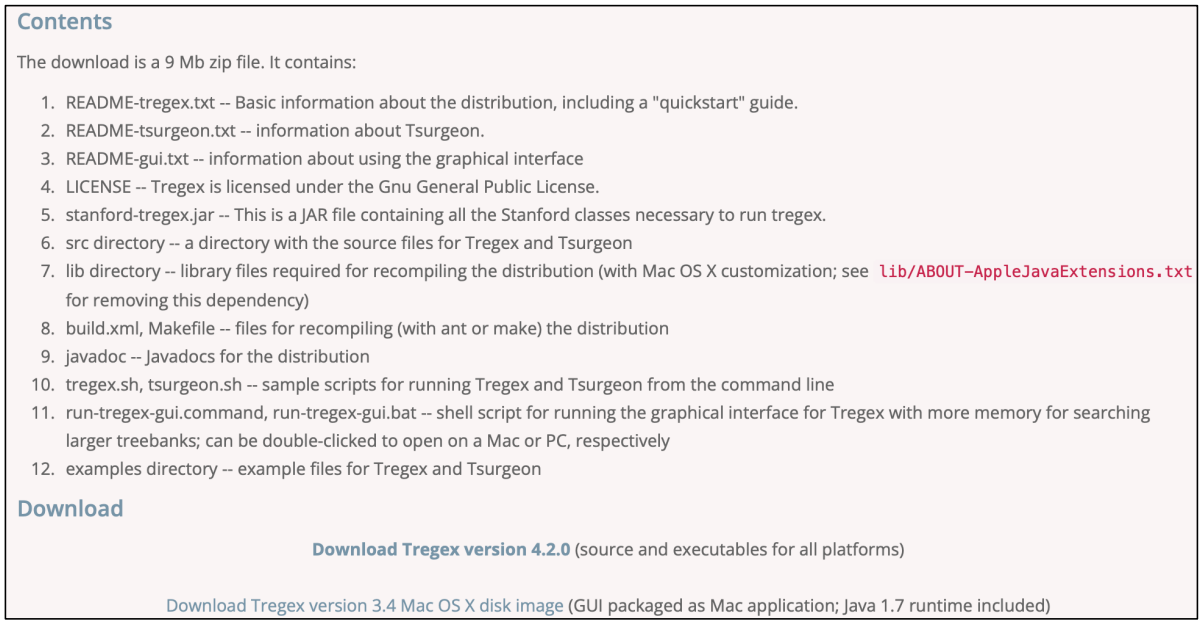
- (from The Penn Treebank: An overview, Taylor, Marcus & Santorini)

Table 1.3. Functional Tags

Text Categories	
-HLN	headlines and datelines
-LST	list markers
-TTL	titles
Grammatical Functions	
-CLF	true clefts
-NOM	non NPs that function as NPs
-ADV	clausal and NP adverbials
-LGS	logical subjects in passives
-PRD	non VP predicates
-SBJ	surface subject
-TPC	topicalized and fronted constituents
-CLR	closely related - see text
Semantic Roles	
-VOC	vocatives
-DIR	direction & trajectory
-LOC	location
-MNR	manner
-PRP	purpose and reason
-TMP	temporal phrases

Tregex

- URL: https://nlp.stanford.edu/software/tregex.shtml



Contents

The download is a 9 Mb zip file. It contains:

1. README-tregex.txt -- Basic information about the distribution, including a "quickstart" guide.
2. README-tsurgeon.txt -- information about Tsurgeon.
3. README-gui.txt -- information about using the graphical interface
4. LICENSE -- Tregex is licensed under the Gnu General Public License.
5. stanford-tregex.jar -- This is a JAR file containing all the Stanford classes necessary to run tregex.
6. src directory -- a directory with the source files for Tregex and Tsurgeon
7. lib directory -- library files required for recompiling the distribution (with Mac OS X customization; see `lib/ABOUT-AppleJavaExtensions.txt` for removing this dependency)
8. build.xml, Makefile -- files for recompiling (with ant or make) the distribution
9. javadoc -- Javadocs for the distribution
10. tregex.sh, tsurgeon.sh -- sample scripts for running Tregex and Tsurgeon from the command line
11. run-tregex-gui.command, run-tregex-gui.bat -- shell script for running the graphical interface for Tregex with more memory for searching larger treebanks; can be double-clicked to open on a Mac or PC, respectively
12. examples directory -- example files for Tregex and Tsurgeon

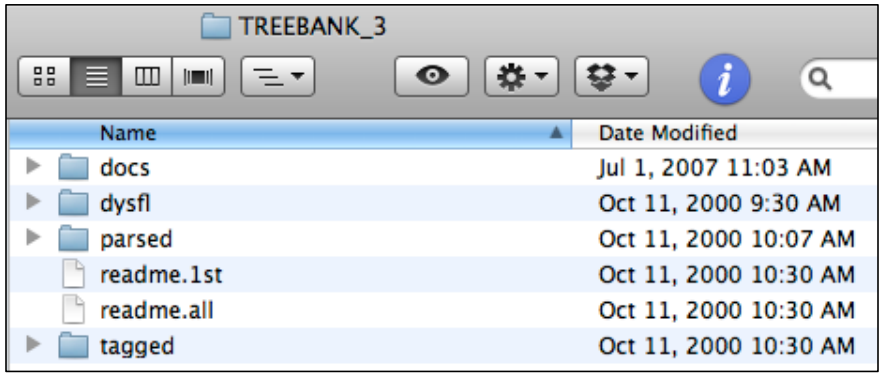
Download

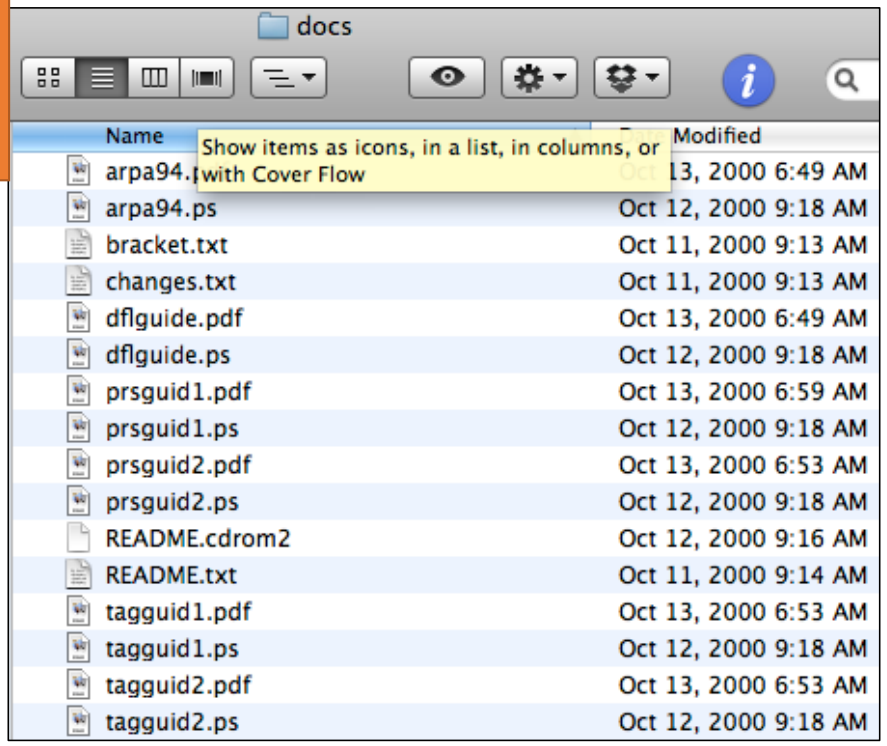
Download Tregex version 4.2.0 (source and executables for all platforms)

Download Tregex version 3.4 Mac OS X disk image (GUI packaged as Mac application; Java 1.7 runtime included)

Treebank Guides

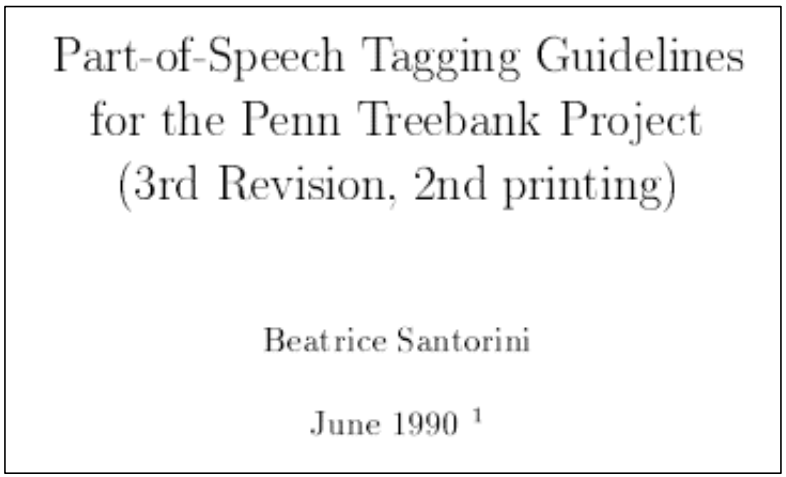
- Tagging Guide
- arpa94 paper
- Parse Guide





Treebank Guides

- Parts-of-speech (POS) Tagging Guide, tagguid1.pdf (34 pages):



Part-of-Speech Tagging Guidelines
for the Penn Treebank Project
(3rd Revision, 2nd printing)

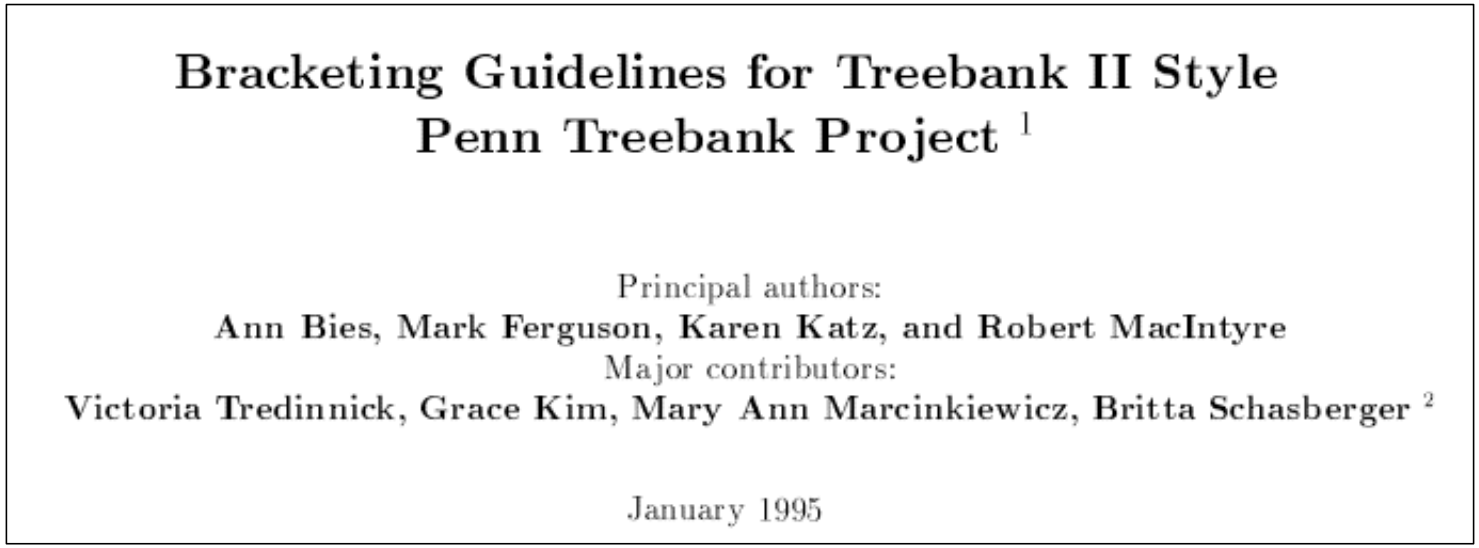
Beatrice Santorini

June 1990 [1]

tagguid2.pdf: addendum, see POS tag ‘TO’

Treebank Guides

- Parsing guide 1, prsguid1.pdf (318 pages):



Bracketing Guidelines for Treebank II Style
Penn Treebank Project [1]

Principal authors:
Ann Bies, Mark Ferguson, Karen Katz, and Robert MacIntyre
Major contributors:
Victoria Tredinnick, Grace Kim, Mary Ann Marcinkiewicz, Britta Schasberger [2]

January 1995

prsguid2.pdf: addendum for the Switchboard corpus

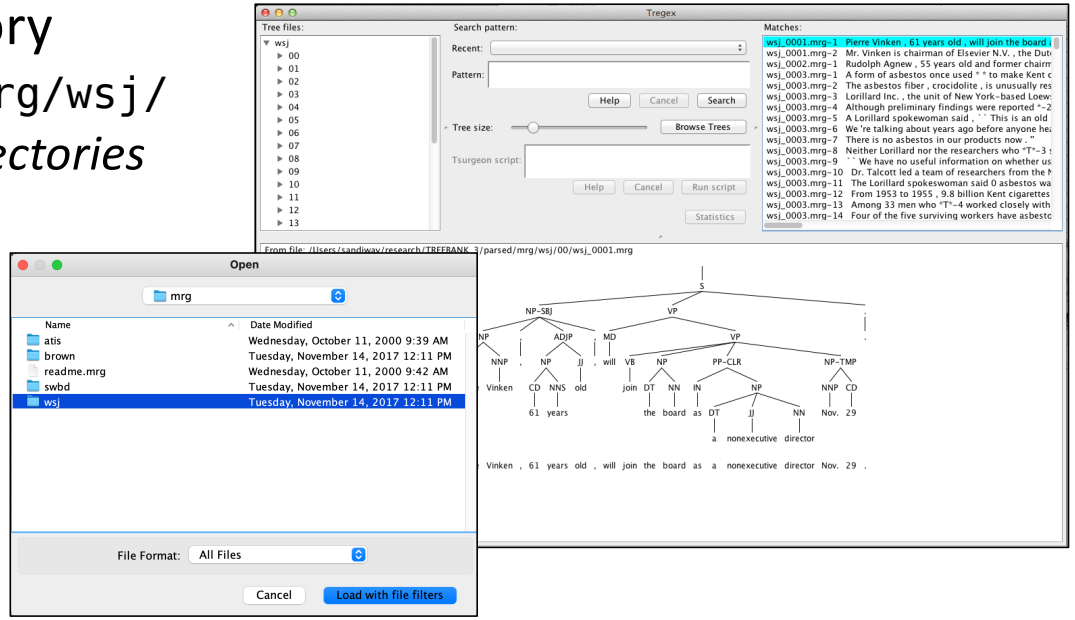
tregex

1. Shell file:

run-tregex-gui.command

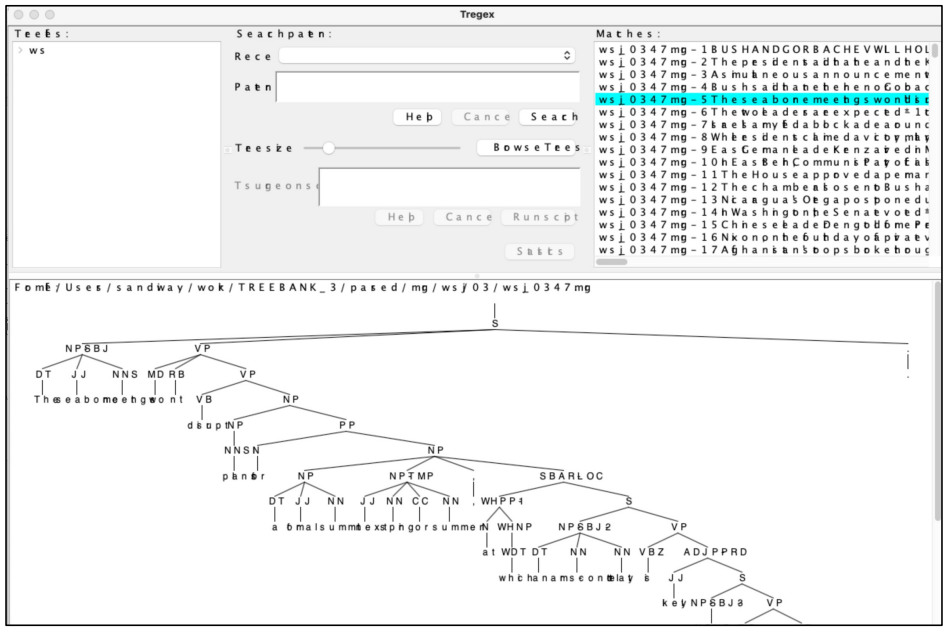
```
#!/bin/sh
java -mx300m -cp `dirname $0`/stanford-tregex.jar edu.stanford.nlp.trees.tregex.gui.TregexGUI
```

2. Select the PTB directory
   - `TREEBANK_3/parsed/mrg/wsj/`
   - *you can select more directories*
3. Browse Trees



Possible macOS Problem

- Disk image version, the Java runtime environment seems to pick wrong fonts. Display is hard to read.

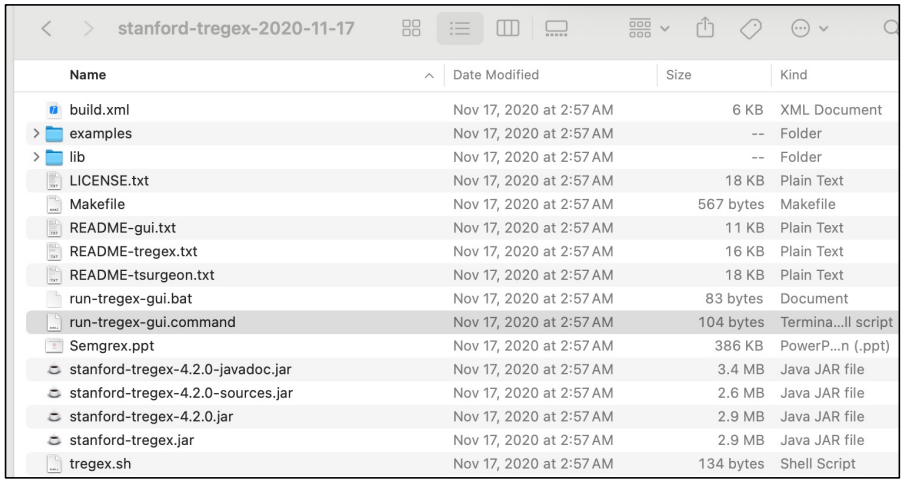


Possible macOS Problem

- If this happens, download the non-image link

Download Tregex version 4.2.0 (source and executables for all platforms)

Download Tregex version 3.4 Mac OS X disk image (GUI packaged as Mac application; Java 1.7 runtime included)



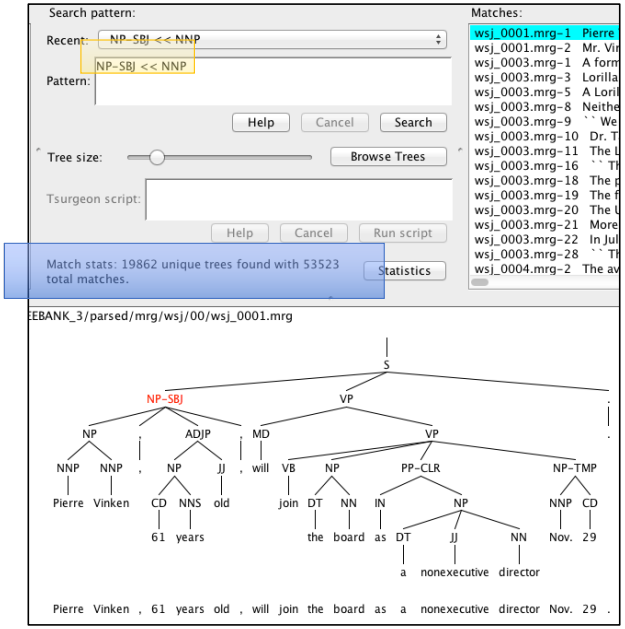
for macOS, run this command

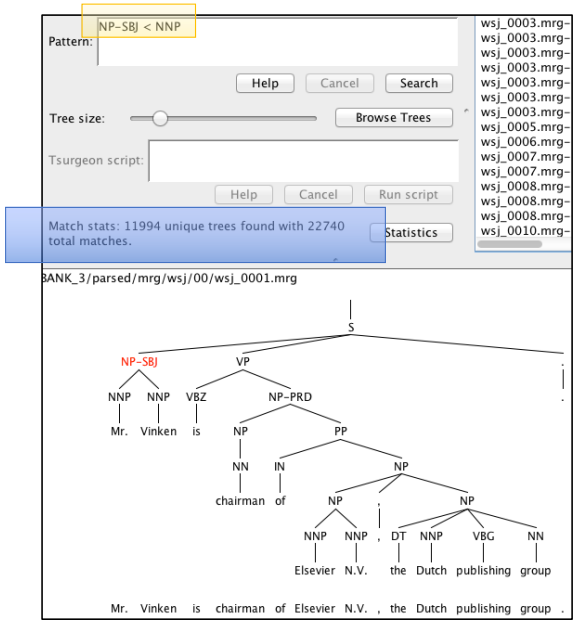
tregex

Statistics History

Pattern	Trees Matched	Total Matches
NP-SBJ << NNP	19862	53523
NP-SBJ < NNP	11994	22740

- Search
  - NP-SBJ << (*dominates*) vs. < (*immediately dominates*) NNP





tregex

- README-tregex.txt

Tregex Pattern Syntax and Uses

Using a Tregex pattern, you can find only those trees that match the pattern you're looking for. The following table shows the symbols that are allowed in the pattern, and below there is more information about using these patterns.

Symbol	Meaning
A << B	A dominates B
A >> B	A is dominated by B
A < B	A immediately dominates B
A > B	A is immediately dominated by 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
A ,, B	A follows B
A , B	A immediately follows B
A <<, B	B is a leftmost descendent of A
A <<- B	B is a rightmost descendent of A
A >>, B	A is a leftmost descendent of B
A >>- B	A is a rightmost descendent of B
A <, B	B is the first child of A
A >, B	A is the first child of B
A <- B	B is the last child of A
A >- B	A is the last child of B
A <` B	B is the last child of A
A >` B	A is the last child of B
A <i B	B is the ith child of A (i > 0)
A >i B	A is the ith child of B (i > 0)
A <-i B	B is the ith-to-last child of A (i > 0)
A >-i B	A is the ith-to-last child of B (i > 0)

A <: B	B is the only child of A
A >: B	A is the only child of B
A <<: B	A dominates B via an unbroken chain (length > 0) of unary local trees.
A >>: B	A is dominated by B via an unbroken chain (length > 0) of unary local trees.
A $++ B	A is a left sister of B (same as $.. for context-free trees)
A $-- B	A is a right sister of B (same as $,, for context-free trees)
A $+ B	A is the immediate left sister of B (same as $. for context-free trees)
A $- B	A is the immediate right sister of B (same as $, for context-free trees)
A $.. B	A is a sister of B and precedes B
A $,, B	A is a sister of B and follows B
A $. B	A is a sister of B and immediately precedes B
A $, B	A is a sister of B and immediately follows B
A <+(C) B	A dominates B via an unbroken chain of (zero or more) nodes matching description C
A >+(C) B	A is dominated by B via an unbroken chain of (zero or more) nodes matching description C
A .+(C) B	A precedes B via an unbroken chain of (zero or more) nodes matching description C
A ,+(C) B	A follows B via an unbroken chain of (zero or more) nodes matching description C
A <<# B	B is a head of phrase A
A >># B	A is a head of phrase B
A <# B	B is the immediate head of phrase A
A >># B	A is the immediate head of phrase B
A == B	A and B are the same node
A : B	[this is a pattern-segmenting operator that places no constraints on the relationship between A and B]