LING/C SC 581:
Advanced Computational Linguistics

Lecture 2

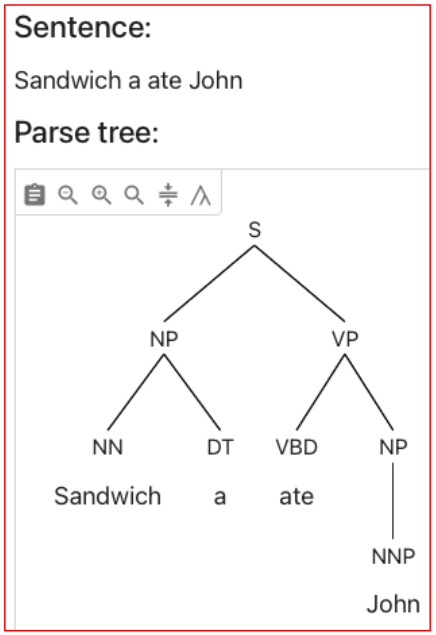
Prof. Sandiway Fong

Today's Topics

- HW 1 Review
- *Penn Treebank* (PTB)
  - I assume everyone has downloaded TREEBANK_3.zip
- installing the full PTB into nltk (Homework 2)
  - `from nltk.corpus import treebank` (3,914 sample)
  - `from nltk.corpus import ptb` (full)
- `tregex`
  - *see* **Appendix** *for macOS TimesRoman FontBook problem*

Homework 1 Review

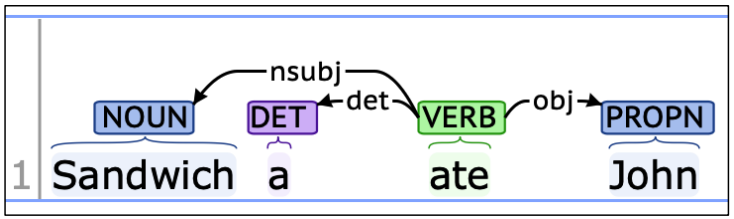
- Q1: what rule of English syntax is violated?

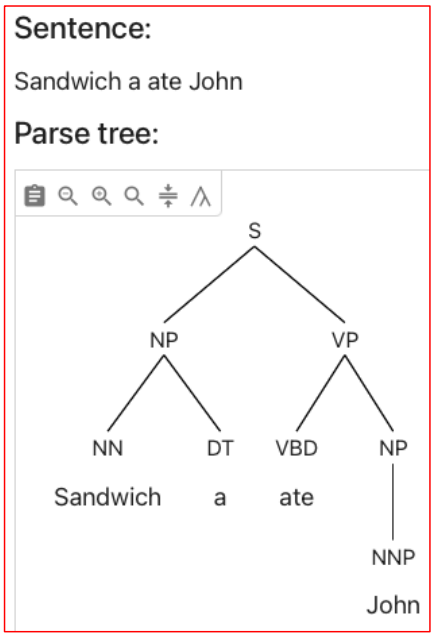


Note:
an utterance can be semantically odd but syntactically fine

Homework 1 Review

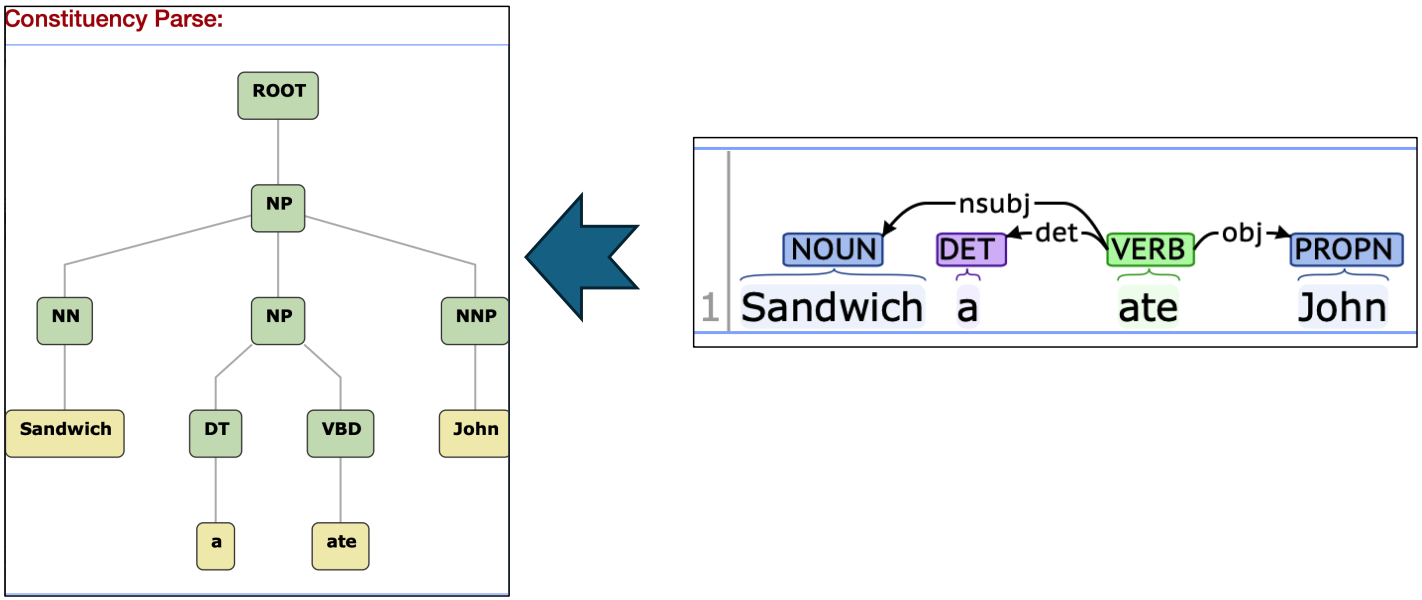
- Q2: what's wrong here? Also compare with Q1.





Homework 1 Review

- Q3: What's wrong with the **phrase** labels built by the conversion?



nltk: **Corpus Readers**

- http://www.nltk.org/howto/corpus.html#parsed-corpora

> If you have access to a full installation of the Penn Treebank, NLTK can be configured to load it as well. Download the `ptb` package, and in the directory `nltk_data/corpora/ptb` place the `BROWN` and `WSJ` directories of the Treebank installation (symlinks work as well). Then use the `ptb` module instead of `treebank`:

  - `nltk.download('ptb')`

```
>>> from nltk.corpus import ptb
>>> print(ptb.fileids()) # doctest: +SKIP ['BROWN/CF/CF01.MRG', 'BROWN/CF/CF02.MRG',
'BROWN/CF/CF03.MRG', 'BROWN/CF/CF04.MRG', ...]
>>> print(ptb.words('WSJ/00/WSJ_0003.MRG')) # doctest: +SKIP ['A', 'form', 'of',
'asbestos', 'once', 'used', '*', ...]
>>> print(ptb.tagged_words('WSJ/00/WSJ_0003.MRG')) # doctest: +SKIP [('A', 'DT'),
('form', 'NN'), ('of', 'IN'), ...]
```

Penn Treebank (PTB) with nltk

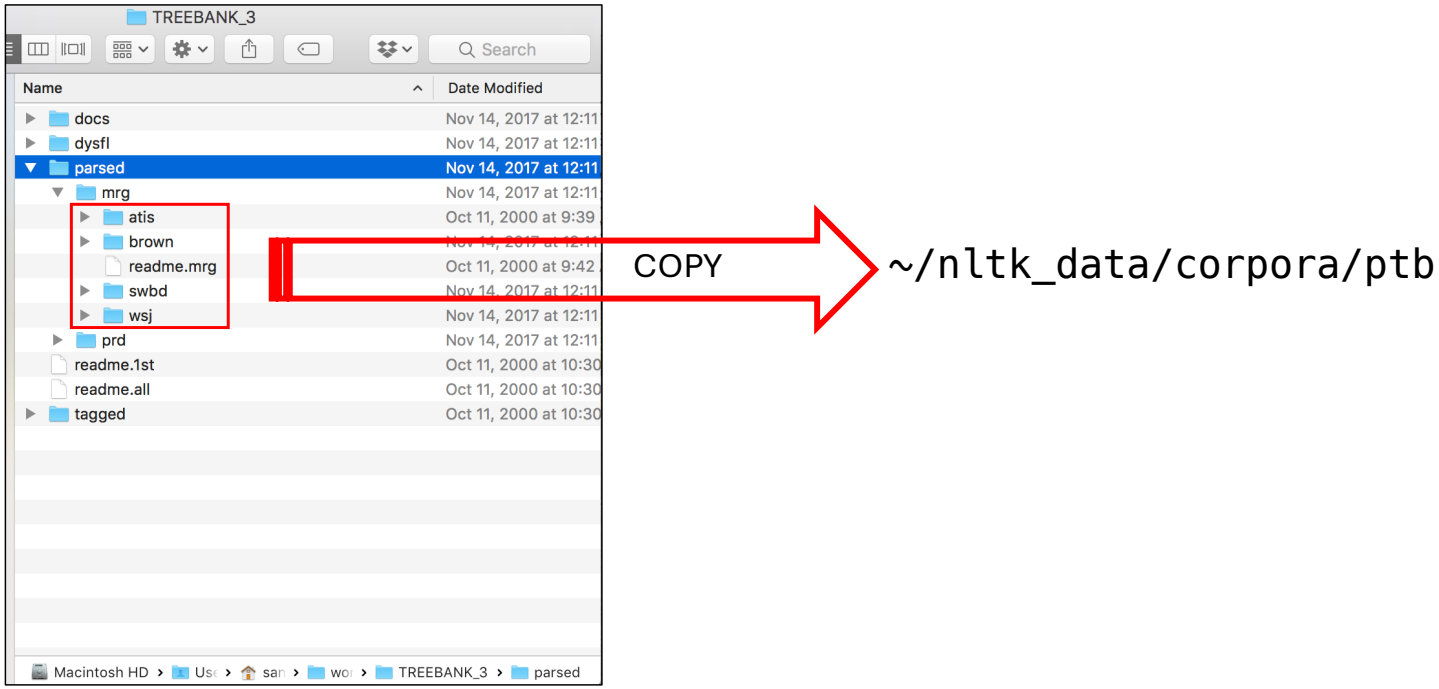
- TREEBANK_3.zip
- Put your wsj directory (from mrg) here `~/nltk_data/corpora/ptb`

```
Sandiways-MacBook:~ sandiway$ python3
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 26 2016, 10:47:25)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import nltk
>>> nltk.download('ptb')
[nltk_data] Downloading package ptb to /Users/sandiway/nltk_data...
[nltk_data]   Unzipping corpora/ptb.zip.
True
>>>
```

```
Sandiways-MacBook:ptb sandiway$ cd wsj/00
Sandiways-MacBook:00 sandiway$ ls
wsj_0001.mrg    wsj_0021.mrg    wsj_0041.mrg    wsj_0061.mrg    wsj_0081.mrg
wsj_0002.mrg    wsj_0022.mrg    wsj_0042.mrg    wsj_0062.mrg    wsj_0082.mrg
wsj_0003.mrg    wsj_0023.mrg    wsj_0043.mrg    wsj_0063.mrg    wsj_0083.mrg
```

Filename case problem!

Penn Treebank (PTB) with nltk



Penn Treebank (PTB) with nltk

- Rename files to uppercase
  - `for f in \`find wsj\`; do mv -v "$f" "\`echo $f | tr '[a-z]' '[A-Z]'\`"; done`
  - (found on stackoverflow.com)
  - *seems to work but not clean*

```
wsj/14/wsj_1493.mrg -> WSJ/14/WSJ_1493.MRG
mv: rename wsj/22 to WSJ/22/22: Invalid argument
wsj/22/wsj_2236.mrg -> WSJ/22/WSJ_2236.MRG
wsj/22/wsj_2222.mrg -> WSJ/22/WSJ_2222.MRG
wsj/22/wsj_2223.mrg -> WSJ/22/WSJ_2223.MRG
```

directory name needs to be uppercased too!

```
Sandiways-MacBook:WSJ sandiway$ cd 00
Sandiways-MacBook:00 sandiway$ ls
WSJ_0001.MRG    WSJ_0021.MRG    WSJ_0041.MRG    WSJ_0061.MRG    WSJ_0081.MRG
WSJ_0002.MRG    WSJ_0022.MRG    WSJ_0042.MRG    WSJ_0062.MRG    WSJ_0082.MRG
WSJ_0003.MRG    WSJ_0023.MRG    WSJ_0043.MRG    WSJ_0063.MRG    WSJ_0083.MRG
WSJ_0004.MRG    WSJ_0024.MRG    WSJ_0044.MRG    WSJ_0064.MRG    WSJ_0084.MRG
WSJ_0005.MRG    WSJ_0025.MRG    WSJ_0045.MRG    WSJ_0065.MRG    WSJ_0085.MRG
```

Penn Treebank (PTB) with nltk

- **Note**: you may run into problems with file permissions when renaming:

```
atis -> ATIS
override r--r--r--  sandiway/staff for ATIS/ATIS3.MRG? (y/n [n]) ^C
```

- Change permissions (recursively):
  - `chmod –R u+w atis`

Penn Treebank (PTB) with nltk

Renaming script courtesy of *Sandeep Suntwal* (from 2018's class):

```
import os
import sys

#Change below path as per your computer
path =  'c:\\Users\sandeep\AppData\Roaming\\nltk_data\corpora\ptb\wsj\\'

for subdir, dirs, files in os.walk(path):
    for filename in files:
        newFileName= filename.upper()
        os.rename(os.path.join(subdir, filename), os.path.join(subdir, newFileName))
```

Penn Treebank (PTB) with nltk

```
>>> from nltk.corpus import ptb
>>> print(ptb.fileids())
['BROWN/CF/CF01.MRG', 'BROWN/CF/CF02.MRG', 'BROWN/CF/CF03.MRG', 'BROWN/CF/CF04.M
RG', 'BROWN/CF/CF05.MRG', 'BROWN/CF/CF06.MRG', 'BROWN/CF/CF07.MRG', 'BROWN/CF/CF
08.MRG', 'BROWN/CF/CF09.MRG', 'BROWN/CF/CF10.MRG', 'BROWN/CF/CF11.MRG', 'BROWN/C
F/CF12.MRG', 'BROWN/CF/CF13.MRG', 'BROWN/CF/CF14.MRG', 'BROWN/CF/CF15.MRG', 'BRO
WN/CF/CF16.MRG', 'BROWN/CF/CF17.MRG', 'BROWN/CF/CF18.MRG', 'BROWN/CF/CF19.MRG',
'BROWN/CF/CF20.MRG', 'BROWN/CF/CF21.MRG', 'BROWN/CF/CF22.MRG', 'BROWN/CF/CF23.MR
G', 'BROWN/CF/CF24.MRG', 'BROWN/CF/CF25.MRG', 'BROWN/CF/CF26.MRG', 'BROWN/CF/CF2
7.MRG', 'BROWN/CF/CF28.MRG', 'BROWN/CF/CF29.MRG', 'BROWN/CF/CF30.MRG', 'BROWN/CF
/CF31.MRG', 'BROWN/CF/CF32.MRG', 'BROWN/CG/CG01.MRG', 'BROWN/CG/CG02.MRG', 'BROW
N/CG/CG03.MRG', 'BROWN/CG/CG04.MRG', 'BROWN/CG/CG05.MRG', 'BROWN/CG/CG06.MRG', '
BROWN/CG/CG07.MRG', 'BROWN/CG/CG08.MRG', 'BROWN/CG/CG09.MRG', 'BROWN/CG/CG10.MRG
', 'BROWN/CG/CG11.MRG', 'BROWN/CG/CG12.MRG', 'BROWN/CG/CG13.MRG', 'BROWN/CG/CG14
```

• • •

```
WSJ_2416.MRG', 'WSJ/24/WSJ_2417.MRG', 'WSJ/24/WSJ_2418.MRG', 'WSJ/24/WSJ_2419.MR
G', 'WSJ/24/WSJ_2420.MRG', 'WSJ/24/WSJ_2421.MRG', 'WSJ/24/WSJ_2422.MRG', 'WSJ/24
/WSJ_2423.MRG', 'WSJ/24/WSJ_2424.MRG', 'WSJ/24/WSJ_2425.MRG', 'WSJ/24/WSJ_2426.M
RG', 'WSJ/24/WSJ_2427.MRG', 'WSJ/24/WSJ_2428.MRG', 'WSJ/24/WSJ_2429.MRG', 'WSJ/2
4/WSJ_2430.MRG', 'WSJ/24/WSJ_2431.MRG', 'WSJ/24/WSJ_2432.MRG', 'WSJ/24/WSJ_2433.
MRG', 'WSJ/24/WSJ_2434.MRG', 'WSJ/24/WSJ_2435.MRG', 'WSJ/24/WSJ_2436.MRG', 'WSJ/
24/WSJ_2437.MRG', 'WSJ/24/WSJ_2438.MRG', 'WSJ/24/WSJ_2439.MRG', 'WSJ/24/WSJ_2440
.MRG', 'WSJ/24/WSJ_2441.MRG', 'WSJ/24/WSJ_2442.MRG', 'WSJ/24/WSJ_2443.MRG', 'WSJ
/24/WSJ_2444.MRG', 'WSJ/24/WSJ_2445.MRG', 'WSJ/24/WSJ_2446.MRG', 'WSJ/24/WSJ_244
7.MRG', 'WSJ/24/WSJ_2448.MRG', 'WSJ/24/WSJ_2449.MRG', 'WSJ/24/WSJ_2450.MRG', 'WS
J/24/WSJ_2451.MRG', 'WSJ/24/WSJ_2452.MRG', 'WSJ/24/WSJ_2453.MRG', 'WSJ/24/WSJ_24
54.MRG']
>>>
```

Checking the install:

class BracketParseCorpusReader

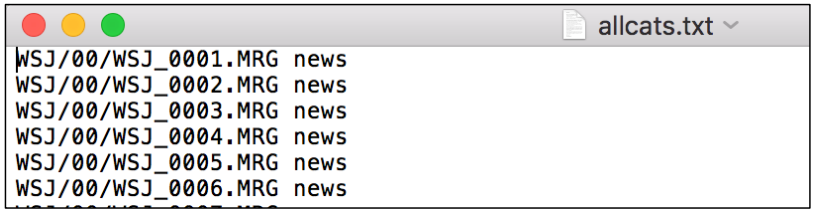
seems to be the Brown corpus + the Wall Street Journal corpus

Penn Treebank (PTB) with nltk

- WSJ only (*news* = WSJ):

```
>>> ptb.categories()
['adventure', 'belles_lettres', 'fiction', 'humor', 'lore', 'mystery', 'news', '
romance', 'science_fiction']
>>> ptb.fileids('news')
['WSJ/00/WSJ_0001.MRG', 'WSJ/00/WSJ_0002.MRG', 'WSJ/00/WSJ_0003.MRG', 'WSJ/00/WS
J_0004.MRG', 'WSJ/00/WSJ_0005.MRG', 'WSJ/00/WSJ_0006.MRG', 'WSJ/00/WSJ_0007.MRG'
, 'WSJ/00/WSJ_0008.MRG', 'WSJ/00/WSJ_0009.MRG', 'WSJ/00/WSJ_0010.MRG', 'WSJ/00/W
SJ_0011.MRG', 'WSJ/00/WSJ_0012.MRG', 'WSJ/00/WSJ_0013.MRG', 'WSJ/00/WSJ_0014.MRG
', 'WSJ/00/WSJ_0015.MRG', 'WSJ/00/WSJ_0016.MRG', 'WSJ/00/WSJ_0017.MRG', 'WSJ/00/
WSJ_0018.MRG', 'WSJ/00/WSJ_0019.MRG', 'WSJ/00/WSJ_0020.MRG', 'WSJ/00/WSJ_0021.MR
G', 'WSJ/00/WSJ_0022.MRG', 'WSJ/00/WSJ_0023.MRG', 'WSJ/00/WSJ_0024.MRG', 'WSJ/00
/WSJ_0025.MRG', 'WSJ/00/WSJ_0026.MRG', 'WSJ/00/WSJ_0027.MRG', 'WSJ/00/WSJ_0028.M
```

- Defined in `~/nltk_data/corpora/ptb/allcats.txt`:



Penn Treebank (PTB) with nltk

- Got it working? Check the numbers below (*proper install*).

```
>>> import nltk
>>> from nltk.corpus import ptb
>>> parses = ptb.parsed_sents()
>>> len(parses)
73451
>>> wsj = ptb.parsed_sents(categories=['news'])
>>> len(wsj)
49208
>>> len(ptb.words())
1740895
>>> len(ptb.words(categories=['news']))
1253013
```

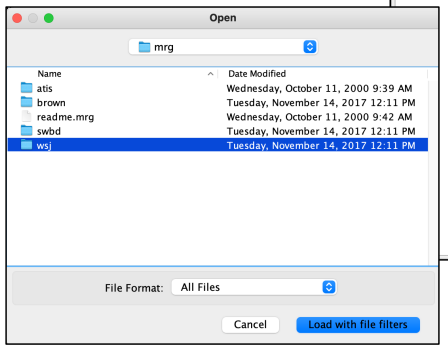
tregex

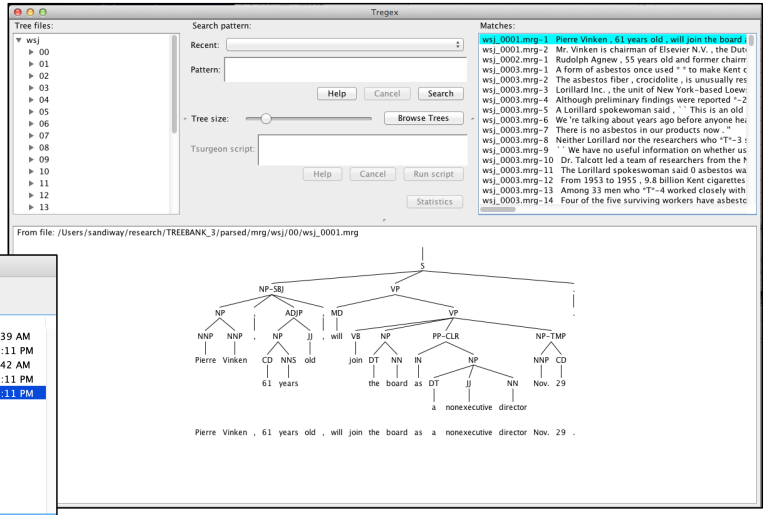
1. Shell file (*edit 300m to give it more memory, e.g. 2000m*):

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



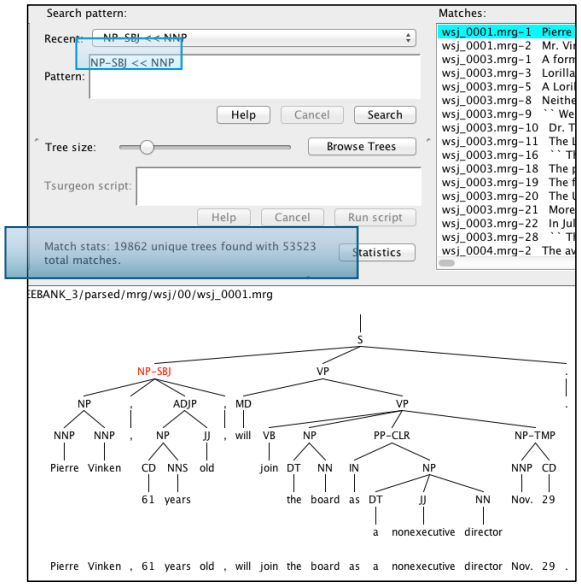


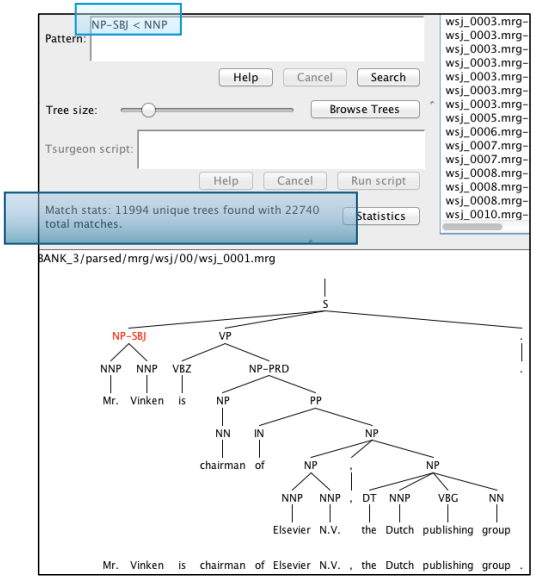
tregex

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)

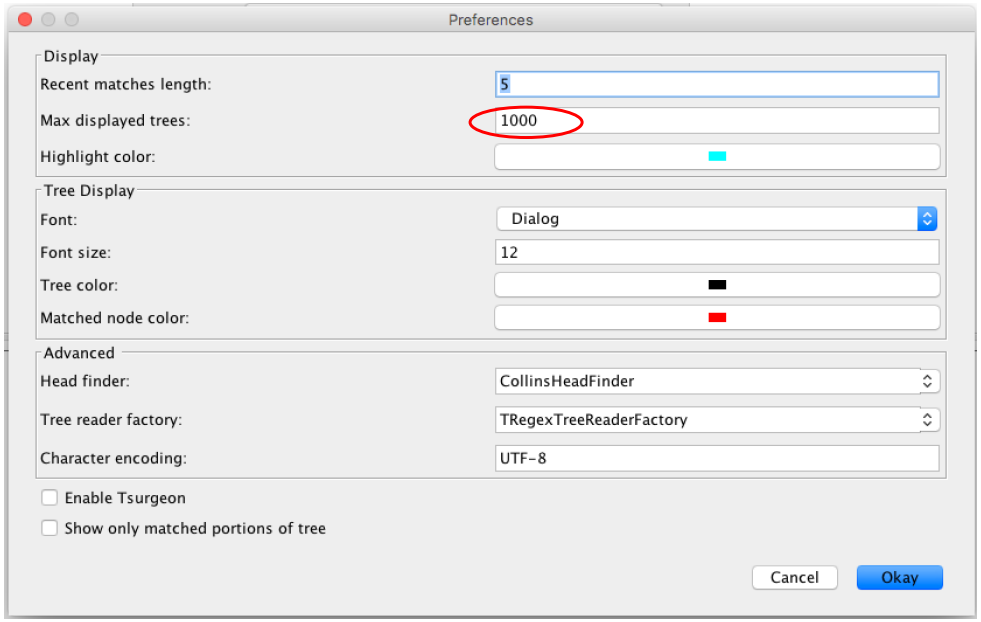
A <: B	B is the only child of A
A >: B	A is the only child of B
A <<: B	A dominates B via an unbroken chain (length > 0) of unary local trees.
A >>: B	A is dominated by B via an unbroken chain (length > 0) of unary local trees.
A $++ B	A is a left sister of B (same as $.. for context-free trees)
A $-- 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]

tregex

- Useful:
  - The (best) introduction to **Tregex** is the brief powerpoint tutorial for **Tregex** by Galen Andrew.
  - https://nlp.stanford.edu/software/tregex/The_Wonderful_World_of_Tregex.ppt

tregex

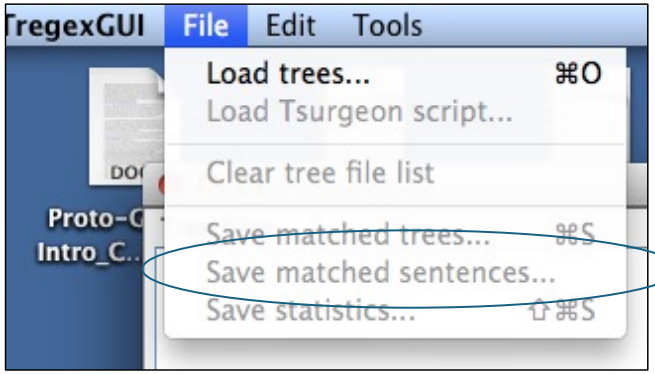
- Adjust Max displayed trees if needed:



tregex

- useful command line tool:
  - **diff <file1> <file2>**

```
dhcp-10-142-182-95:cleft searches sandiway$ diff whclf-0 whclf
4a5,6
> wsj_0415.mrg-5   Who that winner will be *T*-1 is highly uncertain .
> wsj_0415.mrg-22   `` And where we are *T*-1 is bad . ''
```



tregex

- Help: *tregex expression syntax is non-standard wrt bracketing*

Label descriptions can be literal strings, which much match labels exactly, or regular expressions in regular expression bars: /regex/. Literal string matching proceeds as String equality. In order to prevent ambiguity with other Tregex symbols, only standard "identifiers" are allowed as literals, i.e., strings matching [a-zA-Z]([a-zA-Z0-9_])* . If you want to use other symbols, you can do so by using a regular expression instead of a literal string. A disjunctive list of literal strings can be given separated by '|'. The special string '__' (two underscores) can be used to match any node. (WARNING!! Use of the '__' node description may seriously slow down search.) If a label description is preceeded by '@', the label will match any node whose *basicCategory* matches the description. *NB: A single '@' thus scopes over a disjunction specified by '|': @NP|VP means things with basic category NP or VP*. Label description regular expressions are matched as `find()`, as in Perl/tgrep; you need to specify `^` or `$` to constrain matches.

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)`".

Nodes can be grouped using parens '(' and ')' as in `S < (NP $++ VP)` to match an S over an NP, where the NP has a VP as a right sister.

```
S < VP
S < NP
```

tregex

- Help: *tregex boolean syntax is also non-standard*

Boolean relational operators

Relations can be combined using the '&' and '|' operators, negated with the '!' operator, and made optional with the '?' operator. Thus `(NP < NN | < NNS)` will match an NP node dominating either an NN or an NNS. `(NP > S & $++ VP)` matches an NP that is both under an S and has a VP as a right sister.

Relations can be grouped using brackets '[' and ']'. So the expression

```
NP [< NN | < NNS] & > S
```

matches an NP that (1) dominates either an NN or an NNS, and (2) is under an S. Without brackets, & takes precidence over |, and equivalent operators are left-associative. Also note that & is the default combining operator if the operator is omitted in a chain of relations, so that the two patterns are equivalent:

```
(S < VP < NP)
(S < VP & < NP)
```

As another example, `(VP < VV | < NP % NP)` can be written explicitly as `(VP [< VV | [< NP & % NP] ] )`

Relations can be negated with the '!' operator, in which case the expression will match only if there is no node satisfying the relation. For example `(NP ! NNP)` matches only NPs not dominating an NNP. Label descriptions can also be negated with '!': (NP !NNP|NNS) matches NPs dominating some node that is not an NNP or an NNS.

Relations can be made optional with the '?' operator. This way the expression will match even if the optional relation is not satisfied. This is useful when used together with node naming (see below).

tregex

- Help

Basic Categories

In order to consider only the "basic category" of a tree label, i.e. to ignore functional tags or other annotations on the label, prefix that node's description with the @ symbol. For example `(@NP @/NN.?/)` This can only be used for individual nodes; if you want all nodes to use the basic category, it would be more efficient to use a {@link edu.stanford.nlp.trees.TreeNormalizer} to remove functional tags before passing the tree to the TregexPattern.

Segmenting patterns

The ":" operator allows you to segment a pattern into two pieces. This can simplify your pattern writing. For example, the pattern

S : NP

matches only those S nodes in trees that also have an NP node.

tregex

- x <, y, 1st child y; x <– y, last child y;
- x $+ y, x immediate left sister of y

Naming nodes

Nodes can be given names (a.k.a. handles) using '='. A named node will be stored in a map that maps names to nodes so that if a match is found, the node corresponding to the named node can be extracted from the map. For example `(NP < NNP=name)` will match an NP dominating an NNP and after a match is found, the map can be queried with the name to retreived the matched node using `TregexMatcher#getNode(Object o)` with (String) argument "name" (not "=name"). Note that you are not allowed to name a node that is under the scope of a negation operator (the semantics would be unclear, since you can't store a node that never gets matched to). Trying to do so will cause a `ParseException` to be thrown. Named nodes *can* be put within the scope of an optionality operator.

Named nodes that refer back to previous named nodes need not have a node description -- this is known as "backreferencing". In this case, the expression will match only when all instances of the same name get matched to the same tree node. For example: the pattern

```
(@NP <, (@NP $+ (/,/ $+ (@NP $+ /,/=comma))) <- =comma)
```

matches only an NP dominating exactly the sequence `NP , NP ,` -- the mother NP cannot have any other daughters. Multiple backreferences are allowed. If the node w/ no node description does not refer to a previously named node, there will be no error, the expression simply will not match anything.

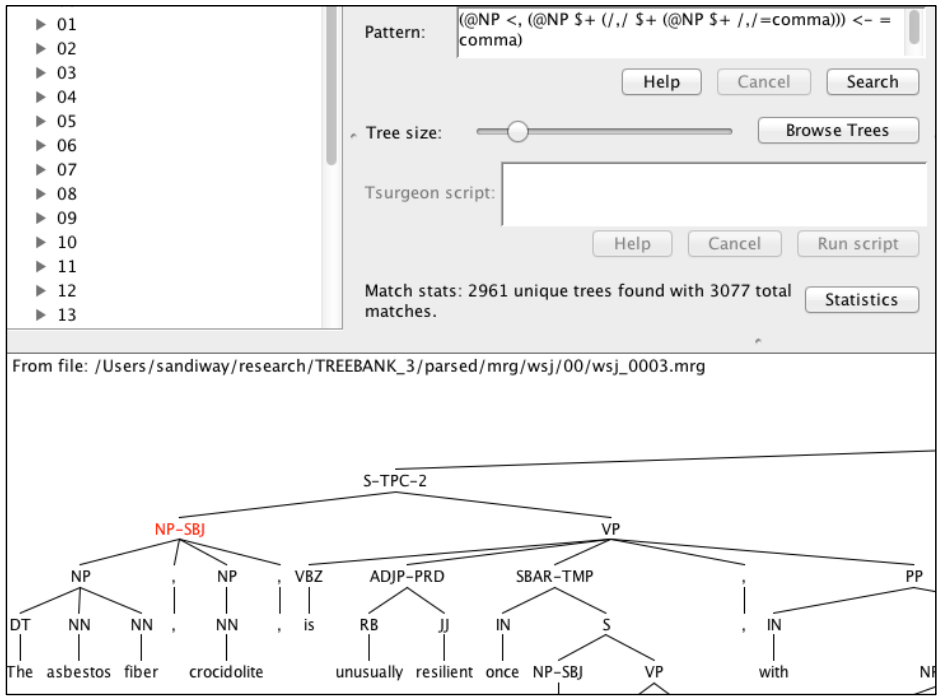
Another way to refer to previously named nodes is with the "link" symbol: '~'. A link is like a backreference, except that instead of having to be *equal to* the referred node, the current node only has to match the label of the referred to node. A link cannot have a node description, i.e. the '~' symbol must immediately follow a relation symbol.

tregex

must be same node

- Pattern:

```
(@NP <, (@NP $+ (/,/ $+ (@NP $+ /,/=comma))) <- =comma)
```



Key:
`<,` first child
`$+` immediate left sister
`<-` last child

tregex

- Help

- Recall regex grouping using parentheses:
  e.g. `(a+)(b+)` defines groups 1 and 2

Variable Groups

If you write a node description using a regular expression, you can assign its matching groups to variable names. If more than one node has a group assigned to the same variable name, then matching will only occur when all such groups capture the same string. This is useful for enforcing coindexation constraints. The syntax is

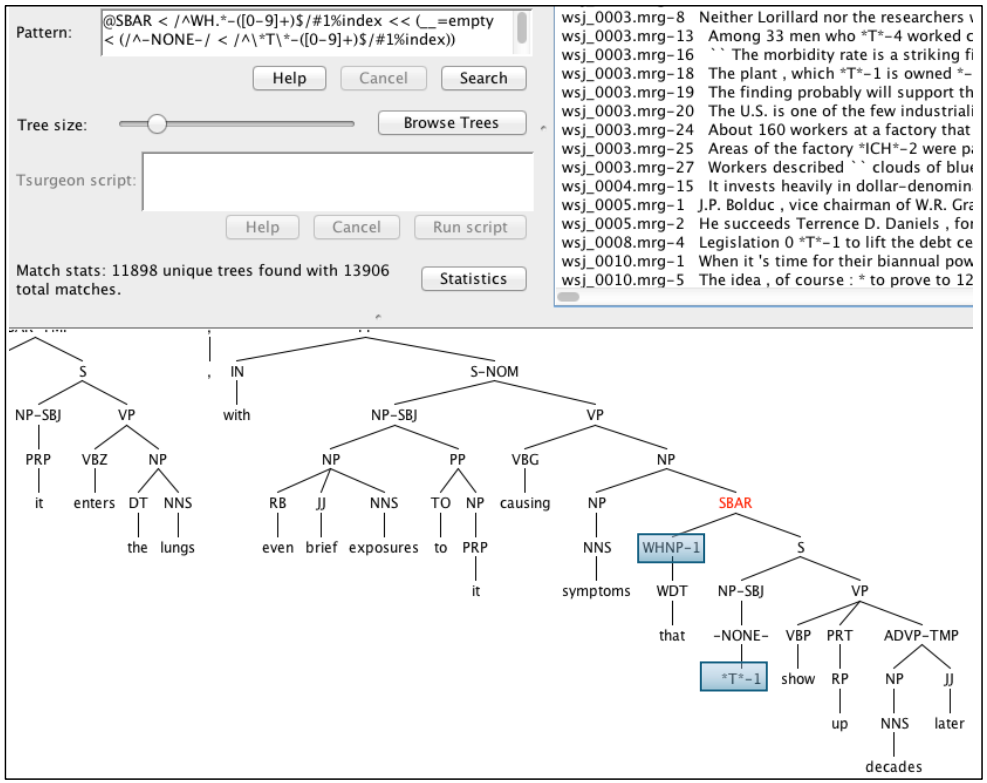
```
/ <regex-stuff> /#<group-number>%<variable-name>
```

For example, the pattern (designed for Penn Treebank trees)

```
@SBAR < /^WH.*-([0-9]+)$/#1%index << (__=empty < (/^-NONE-/ <
/^\*T\*-([0-9]+)$/#1%index))
```

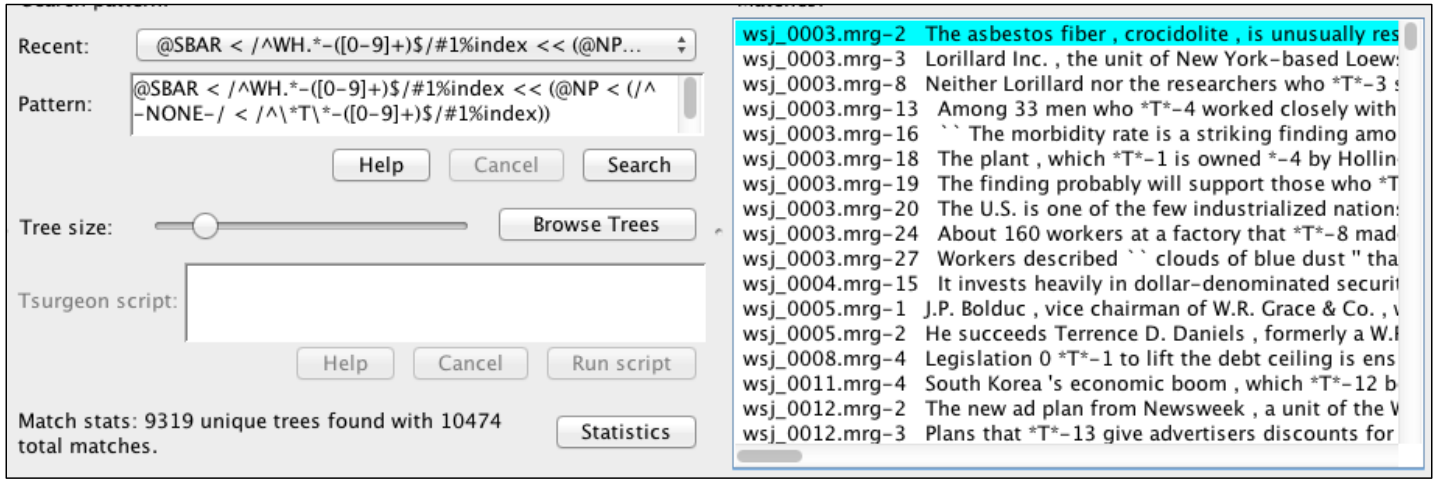
will match only such that the WH- node under the SBAR is coindexed with the trace node that gets the name `empty`.

tregex

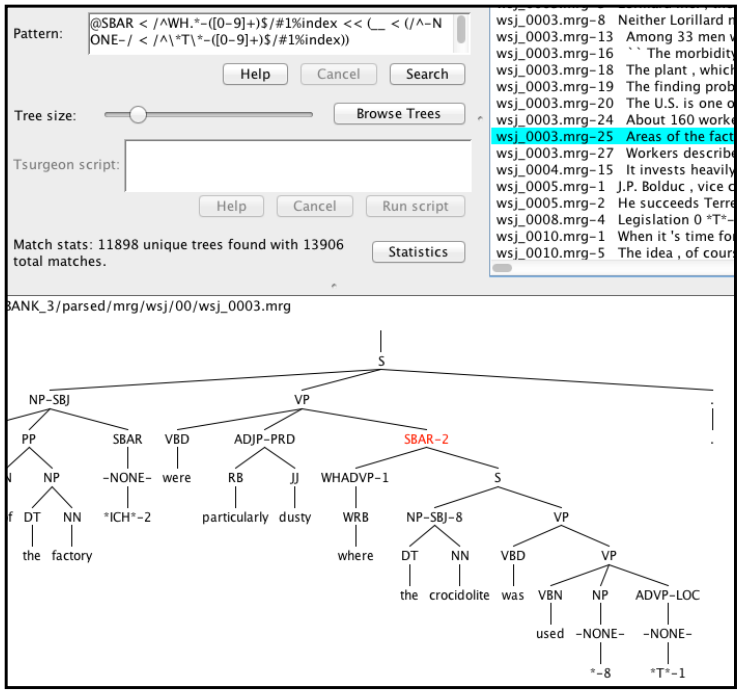

Pattern:
@SBAR < /^WH.*-([0-9]+)$/#1%index << (__=empty
< (/^-NONE-/ < /^*T*-([0-9]+)$/#1%index))
Help
Cancel
Search
Tree size:
Browse Trees
Tsurgeon script:
Help
Cancel
Run script
Match stats: 11898 unique trees found with 13906 total matches.
Statistics
wsj_0003.mrg-8 Neither Lorillard nor the researchers
wsj_0003.mrg-13 Among 33 men who *T*-4 worked c
wsj_0003.mrg-16 `` The morbidity rate is a striking fi
wsj_0003.mrg-18 The plant , which *T*-1 is owned *-
wsj_0003.mrg-19 The finding probably will support th
wsj_0003.mrg-20 The U.S. is one of the few industriali
wsj_0003.mrg-24 About 160 workers at a factory that
wsj_0003.mrg-25 Areas of the factory *ICH*-2 were p
wsj_0003.mrg-27 Workers described `` clouds of blu
wsj_0004.mrg-15 It invests heavily in dollar-denomin
wsj_0005.mrg-1 J.P. Bolduc , vice chairman of W.R. Gra
wsj_0005.mrg-2 He succeeds Terrence D. Daniels , fo
wsj_0008.mrg-4 Legislation 0 *T*-1 to lift the debt ce
wsj_0010.mrg-1 When it 's time for their biannual pow
wsj_0010.mrg-5 The idea , of course : * to prove to 12
S
NP-SBJ
VP
PRP
VBZ
NP
it
enters
DT
NNS
the
lungs
IN
with
S-NOM
NP-SBJ
VP
NP
PP
VBG
NP
RB
JJ
NNS
TO
NP
causing
NP
SBAR
even
brief
exposures
to
PRP
it
NNS
WHNP-1
S
symptoms
WDT
NP-SBJ
VP
that
-NONE-
VBP
PRT
ADVP-TMP
T-1
show
RP
NP
JJ
up
NNS
later
decades

tregex

- Different results from:
  - `@SBAR < /^WH.*-([0-9]+)$/#1%index << (@NP < (/^-NONE-/ < /^\*T\*-([0-9]+)$/#1%index))`



tregex



Reason for difference

Example:

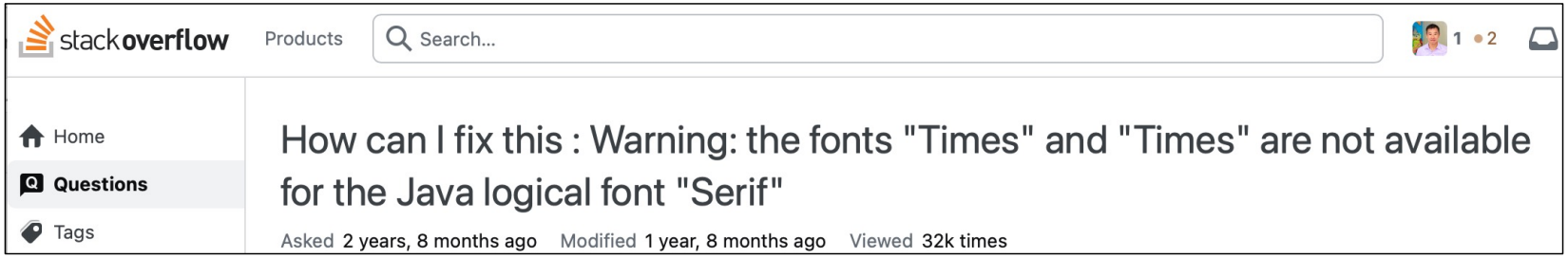
WHADVP also possible (not just WHNP)

Appendix

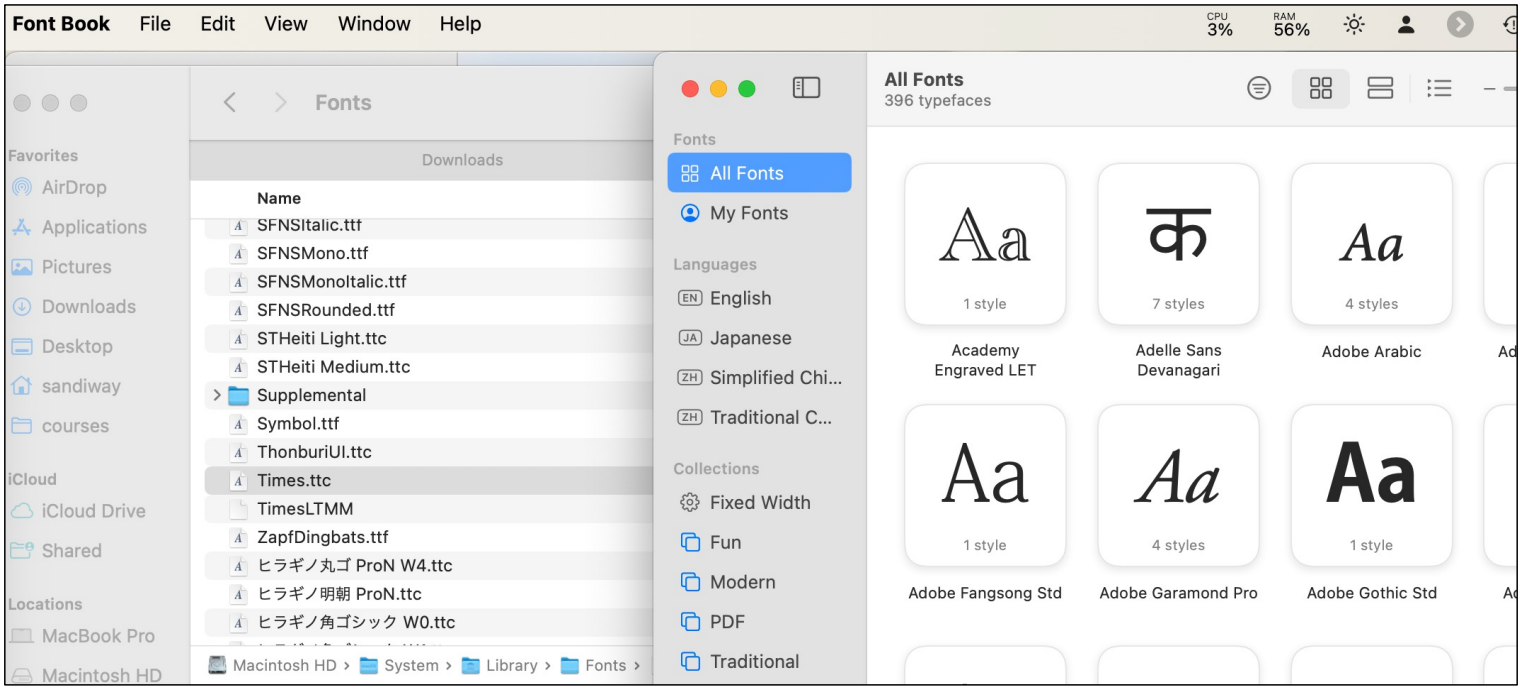
- macOS Times Roman FontBook problem

Times font restore

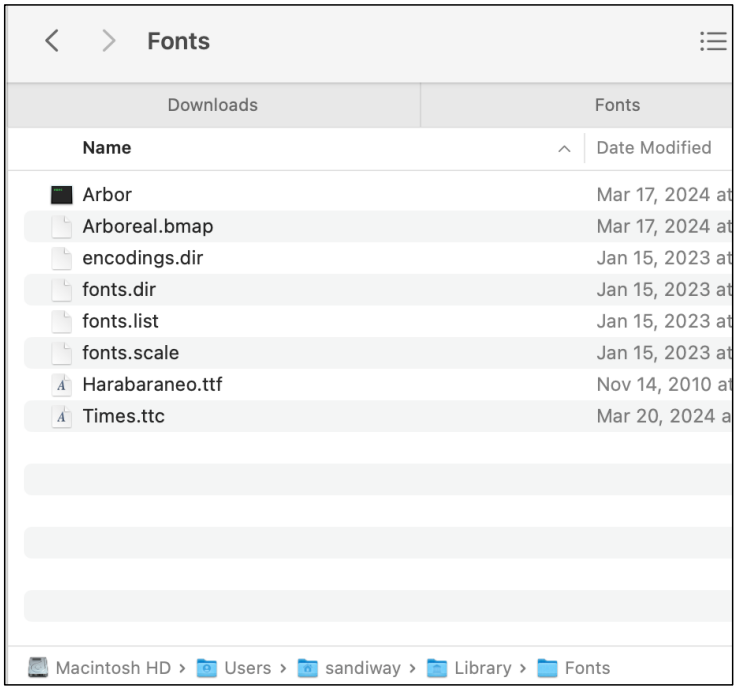
- https://stackoverflow.com/questions/68608157/how-can-i-fix-this-warning-the-fonts-times-and-times-are-not-available-fo



/System/Library/Fonts/Times.ttc



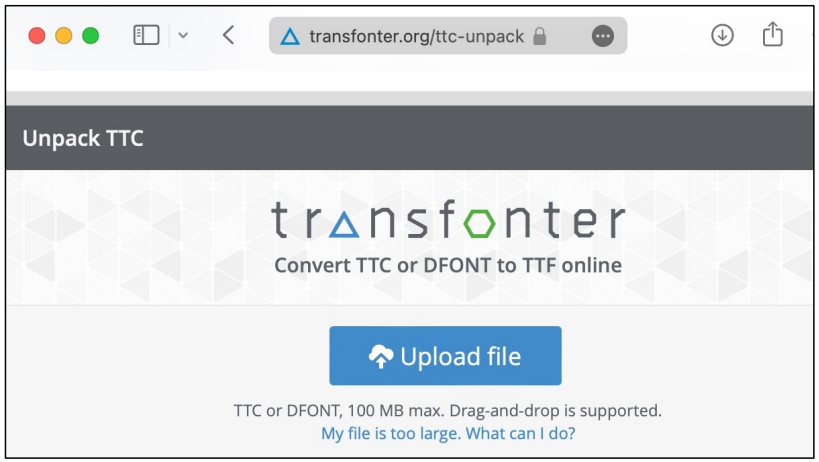
Font Book

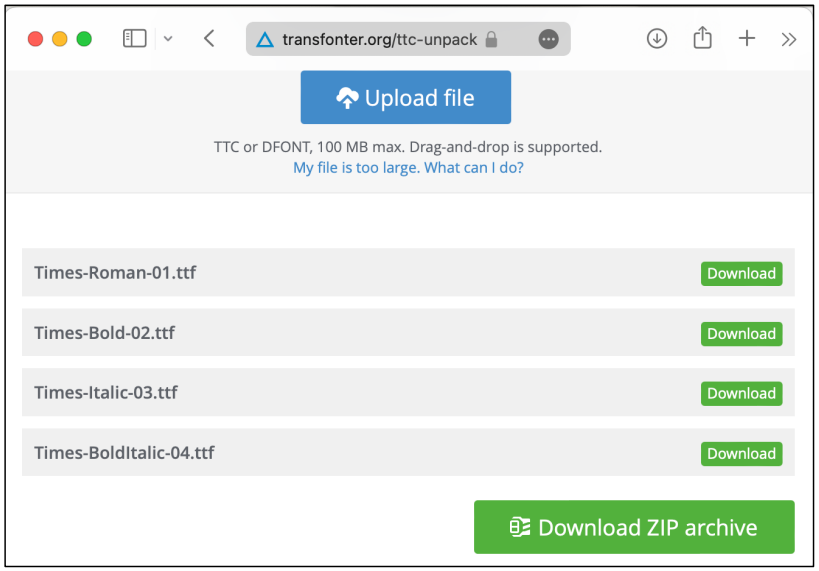


- Drag Times.ttc to Font Book
- Now appears in ~/Library/Fonts

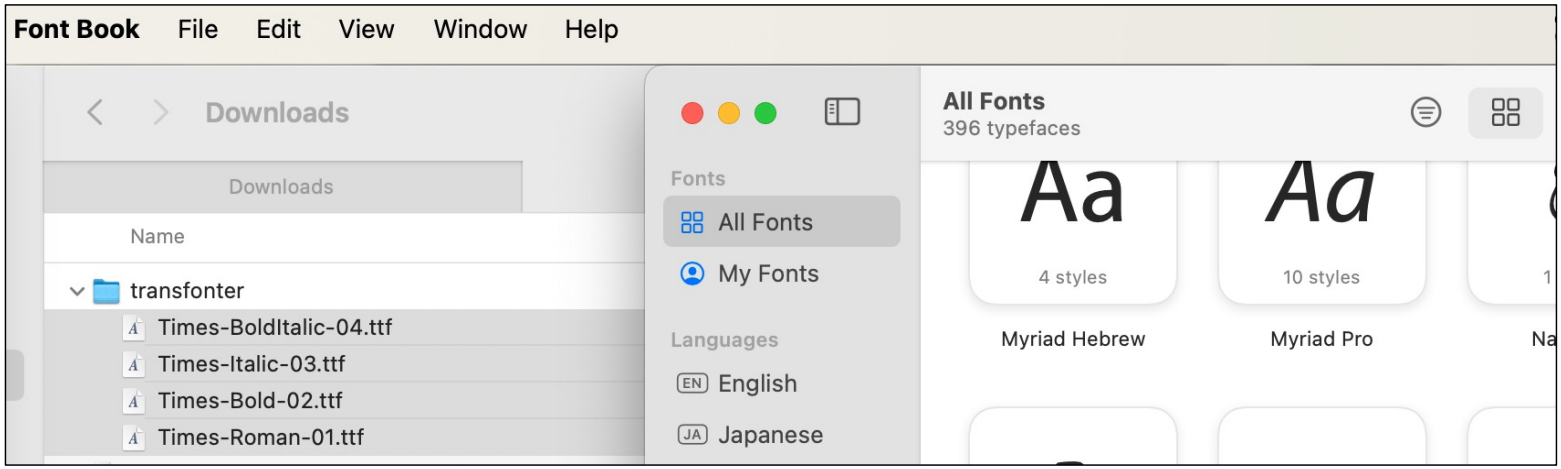
transfonter.org

- upload Times.ttc



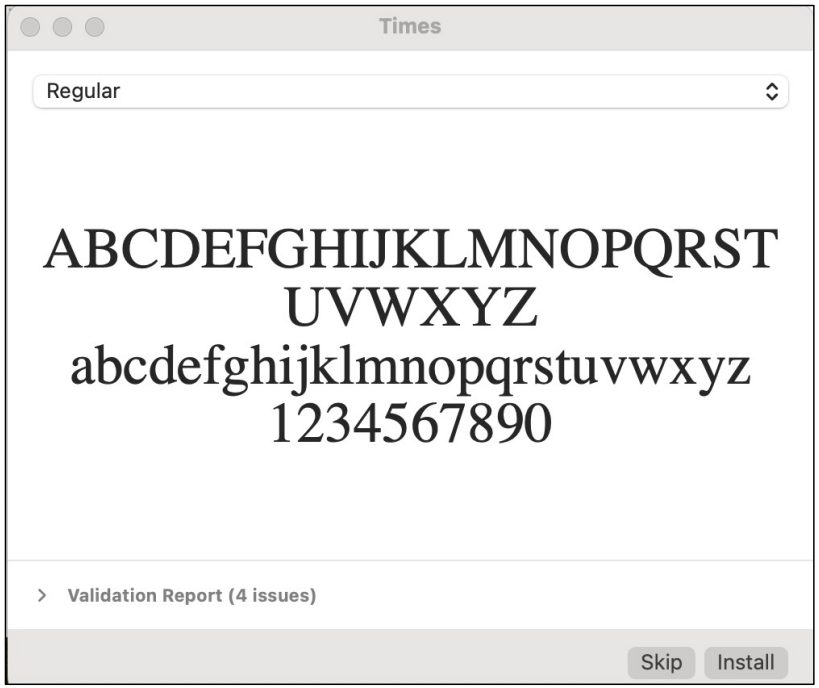


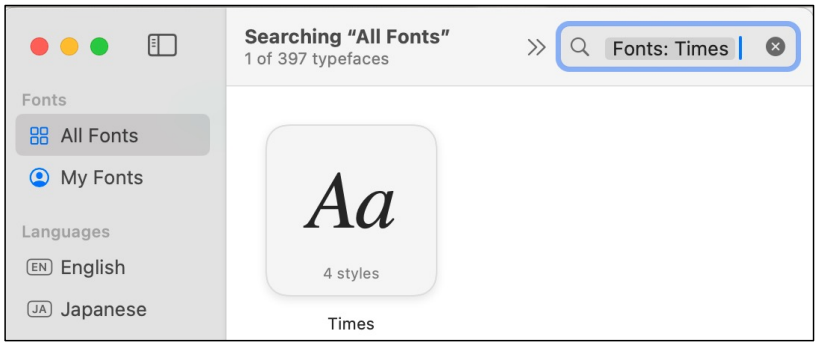
Font Book



- Drag downloaded files to Font Book

Font Book





- install, Times now appears