

Ling 473 Project 1

Due 11:45pm on Thursday, August 2, 2018

For this project you will write a program to count the number of syntactic constituent types that occur in an annotated corpus. Processing all of the files in the following directory, fill out the table below to indicate how many of the syntactic elements of each type are annotated in the entire corpus. Nested constituents of the same type are to be counted equally at any level where they appear.

/corpora/LDC/LDC99T42/RAW/parsed/prd/wsj/14

This is a portion of the Treebank-3 corpus from the Linguistic Data Consortium. You are not permitted to copy this corpus off the computational linguistics cluster.

Constituent	PTB symbol	Count
Sentence	(S ...)	
Noun Phrase	(NP ...)	
Verb Phrase	(VP ...)	
Ditransitive Verb Phrase	(VP <i>verb</i> (NP ...) (NP ...))	
Intransitive Verb Phrase	(VP <i>verb</i>)	

For the last example, we are looking for VPs whose immediate (top-level) constituents include no NPs (or no immediate children at all). It turns out that this will actually give a lot of auxiliary verbs, and also verbs that take a clause as their complement such as “said.”

For the Ditransitive case, we are looking for exactly two immediate constituents of type NP. Do not count NPs that are marked as, for example, NP-SBJ. Dealing with nesting and making sure that you only consider the immediate constituents will be tricky, especially if you are using RegEx, since RegEx does not easily handle matching of balanced parentheses.

You can use Python, Java or C# for this project. If you wish to use a different language and have a good reason for doing so, talk to the instructor before starting the assignment. You do not need to use regular expressions in your program but you are welcome to use this method if you find it convenient. Well-written procedural code is often more self-documenting and maintainable than elaborate regular expressions.

Use absolute paths (path that starts with ‘/corpora/...’) to reference the corpora. Use relative paths (paths that do not start with ‘/’) to reference files you are including with your submission. Do not directly reference your home directory, since I may not have permissions for it when I’m running your program.

Output Format

The output for this assignment should have each constituent on its own line followed by a tab and the count, capitalized and spelled as follows:

Sentence	10
Noun Phrase	23
Verb Phrase	14
Ditransitive Verb Phrase	2
Intransitive Verb Phrase	2

Submission

For this project, you will also create a control file that submits your program as a batch job to the *condor* computing cluster. Include the following files in your submission:

compile.sh	Contains command(s) that compile your program. If you are using python or any other interpreted language that does not require compiling, then this file will be empty, or contain just the single line: #!/bin/sh
run.sh	The command(s) that run your program, emitting required output to the console (stdout). Be sure to include compiled binaries in your submission so that this script will execute without first running compile.sh
condor.cmd	Condor control file, suitable for running your program as follows: condor_submit condor.cmd
output	captured console output (stdout) from running your program, this should be produced by: ./run.sh >output
readme.{pdf, txt}	Your write-up of the project, including a table similar to the one above which reports your results. Describe your approach, any problems or special features, or anything else you'd like me to review. If you could not complete some or all of the project's goals, please explain what you were able to complete.
(source code and binary files)	All source code and binary files (jar, a.out, etc.) required to run and compile your program

Gather together all the required files, making sure that, for example, any PDF or other binary files are transferred from your local machine using a binary transmission format. Then, from within the directory containing your files, issue the following command to package your files for submission.

```
tar -czf hw.tar.gz .
```

Notice that this command packages all files in the current directory; do not include any top-level directories.

Please submit readme.{pdf, txt} separate from your tar ball. Canvas will allow you to upload multiple files at once. Note that if you upload the files in separate submissions, Canvas will overwrite the previous submission, so you should upload the readme file and the tar ball at the same time.

To check that your tar ball contains all of the required files (note that this does not include the readme, because it is expected to be uploaded separately from the output), run **dropbox/18-19/473/project1/check_project1.sh** from the directory containing your tarball. For example, from my project1 folder, I run:

```
~/../../dropbox/18-19/473/project1/check_project1.sh
```

If all files are included, this should run with no errors, and just return "Check complete".

Grading

Correct results	45
All files present, named correctly	10
Clarity and readability of code	15
run.sh runs to completion	15
Write-up	15

Corpus Citation

Mitchell P. Marcus, Beatrice Santorini, Mary Ann Marcinkiewicz and Ann Taylor. 1999. *Treebank-3*.
Linguistic Data Consortium, Philadelphia