Getting SUTime Working

Copy and paste the folder stanford-corenlp-4.0.0 to your computer.

* You *must use* the one from our Github for two reasons:
  + I cannot guarantee we will get the same answers for later versions, and I don’t want to induce tiny/irrelevant changes unnecessarily;
  + This folder contains David’s *altered* SUTime rules file, which we manually edited, and obviously does not come from CoreNLP.

You are going to need the following programmes on your system, and on your system PATH.

* Java.
* Java JDK.
* Maeven.
* Git Bash.

Go to the folder stanford-corenlp-4.0.0.

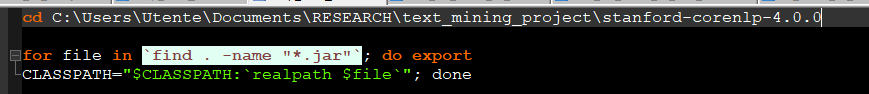
The toolbox calls functions from a Maeven repository. Before you do anything you need to “build” the project by doing this:

* Go to the command line;
* Set your command directory to
* Enter: mvn verify

It should say “BUILD SUCCESS”.

Now run the file setup\_class\_path.sh.

You are going to need to alter this to point to your CoreNLP directory, as you can see below.



Next it would be a good idea to test it is working in a “silly” case.

See here: <https://stanfordnlp.github.io/CoreNLP/download.html> if you need further help.

Note that Chris Manning gives the following “silly” test:

echo "the quick brown fox jumped over the lazy dog" > input.txt

java -mx3g edu.stanford.nlp.pipeline.StanfordCoreNLP -outputFormat json -file input.txt

Personally, I couldn’t get it to work unless I replaced java –mx3g with java –cp “\*” –mx3g. But it should work for you.

Going back to the codes.

You should have the following .java files in your stanford-corenlp-4.0.0 folder.

* run\_sutime\_on\_speeches
* run\_sutime\_on\_SQA
* run\_sutime\_on\_Q
* run\_sutime\_on\_A
* run\_sutime\_on\_FED\_GBK

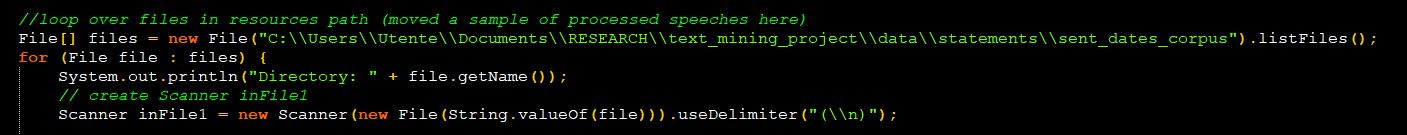
You should also have the files defs2.sutime and english2.sutime in the /sutime folder. These are the ones David altered, and will be called by the code.

Then you want to run the following BASH scripts, which should also be in the stanford-corenlp-4.0.0 folder.

* bash\_run\_sutime\_on\_speeches
* bash\_run\_sutime\_on\_SQA
* bash\_run\_sutime\_on\_Q
* bash\_run\_sutime\_on\_A
* bash\_run\_sutime\_on\_FED\_GBK

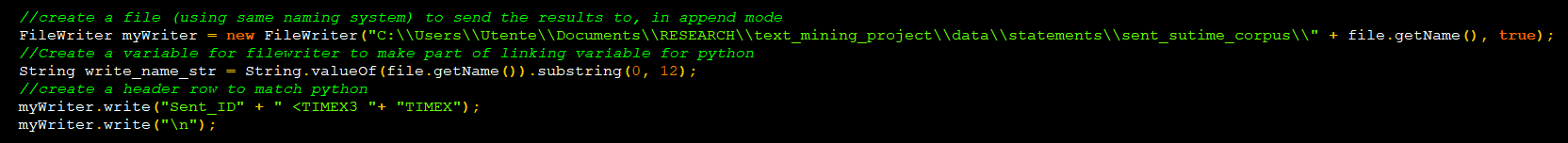
**WARNING!** These will certainly break the first time you try to run them. The reason is that, for example, bash\_run\_sutime\_on\_SQA.sh calls run\_sutime\_on\_SQA.java.

If you open run\_sutime\_on\_SQA.java, you will see that the “input” files come from statements/sent\_dates\_corpus.



However, I have not defined a relative path. So you would need to *manually* change this.

Later down the code you see the output folder.



Again, I have not defined a relative path.

Be careful you set up the input and output correctly, you don’t want to get mixed up here. And it differs depending on the corpus. E.g. the answers corpus takes /answers/sent\_dates\_corpus/ as the input, and /answers/sent\_sutime\_corpus/ as the output.

You have two options here:

* Work out how to pass arguments into the .java codes, so you can re-write the BASH script to “feed in” the correct paths.
* Work out how to use relative paths in Java, and re-code the .java codes to use relative paths appropriately, so they will work on any computer.
* Re-code the .java files so each of them has the correct file structure for your PC.

I tried getting it to use relative paths and it seemed to break. Perhaps I made a silly mistake but I ran out of time. It would be better to use either relative paths, or pass the paths to the functions directly, so we don’t have to re-write functions when we use the codes on different computers. If anyone can fix this be my guest.