

# CS 294-5: Statistical Natural Language Processing

## Accounts and Access

This document is designed to get you up and running with the servers and the code base that this course will be using. If you have questions about an assignment, especially access issues, bugs, and so on, please post to the class newsgroup: `ucb.class.cs294-5`. There is a link to the webnews interface to the newsgroup off of the course homepage.

You will need one of two kinds of accounts to access the course materials: an EECS instructional account, or a Millennium account.

**Getting an account on the instructional system.** An EECS instructional account is the easiest option if you've got access to decent computing resources of your own, or for assignments where lab computers are sufficient. Anyone who can enroll in this class can get one. There's new account information at

<http://www-inst.eecs.berkeley.edu/~iesg/new-users.html>

Machines and labs available for class use are outlined at

<http://www-inst.eecs.berkeley.edu/~iesg/iesglabs.html>

**Getting an account on the NLP cluster.** There is a cluster of 9 dedicated NLP machines on the Millennium network. You can sign up for a Millennium account at

<https://www.millennium.berkeley.edu/account/>

You'll have to list me as your sponsor (Dan Klein, [klein@cs.berkeley.edu](mailto:klein@cs.berkeley.edu)) unless you'll be using the account for more than this class, at which point you should ask your regular research advisor. For the "Description of Research," please enter "cs294-5 class use." If you have an EECS research account already, you can also request under "Description" that your EECS account be exported as your home directory on these machines. The list of machines (and their loads) is available at

<http://www.millennium.berkeley.edu/NLP/index.html>

Please do not use these machines for purposes other than class work, and try not to use more than one CPU – these are primarily research machines and cannot be totally overloaded by class use. Also, please leave `nlp.millennium.berkeley.edu` free of large jobs – it's meant as a front end.

**Getting the course code to run.** I'm going to give directions for running code from a unix shell, and I'm going to assume you know how to make directories, edit files, change your path, etc. under unix. If you want to run code locally, for example on a windows machine, or in an IDE, it's up to you to get that to work, including installing Java 1.5. Note that you may copy the source code and data to local machines, but due to licensing issues, you may **not** distribute either freely.

So, pick your favorite server from the machine list and log in via ssh (info on how to do this can be found at <http://inst.eecs.berkeley.edu/connecting.html>). Make sure you can see the class directory root, which is

```
[instructional] /home/ff/cs294-5/  
[millennium] /work/cs294-5/
```

and which I'll refer to as ROOT. Under this directory, you can find Java in the bin directory. Invoke it to make sure you're getting the right version

```
[instructional] /home/ff/cs294-5/bin/java -version  
[millennium] /work/bin/java -version
```

You should get some subversion of Java 1.5. The java executable in your default paths will not be 1.5, you need this one. When I say "java" from now on, I mean that you should invoke this 1.5 one. I recommend adding the Java executable directory to your path; I'll assume it's in your path from here out.

Under the class root, the source code lives in

```
src/
```

and a pre-compiled set of class files lives in

```
classes/
```

You can run a simple test file by typing

```
java -cp ROOT/classes edu.berkeley.nlp.Test
```

You should get a confirmation message back.

**Getting the course code local and compiling.** You will need to make a local copy of the course code to modify anything. You will only ever need to modify source files in the edu.berkeley.nlp.assignments package, though you're welcome to change anything in your local copy in order to complete the assignments. Be careful if you do – I may modify class source in the event of bugs, and you'll have to do any merging yourself. I'll

assume you've copied the src directory to a local one. You can compile your local copy to a local classes directory with

```
javac -d classes src/**/*.java
```

You're also welcome to use a more sophisticated method of compiling, for example jmake or ant, or an IDE, but that's up to you. Make sure you can run your local compiled code by invoking the Test class locally.