# Using the Stanford Parser

### 1. Download the latest version of the parser.

See the download link on the main page to download the zip file:

<https://stanfordnlp.github.io/CoreNLP/>

This is a large file and may take a few minutes to download. The current version as of this writing is 3.9.1.

Unzip the download wherever you want to keep it. I have a folder called “nlp\_tools” on my disk where I keep software like this. The extracted folder will take up about 500 MB on disk.

### 2. Check your Java version and make sure you have a working IDE.

The current version requres Java 8 (JDK1.8) or later. To check your version of Java, find the Java Control Panel and check the About section on the General tab.

Windows: <https://www.java.com/en/download/help/win_controlpanel.xml>

Mac: <https://www.java.com/en/download/help/mac_controlpanel.xml>

The demos on the next few pages show screen shots from IdeaJ but any IDE will do.

### 3. Test run from the command line.

You can run CoreNLP from the command line in the extracted folder:

java –add-modules java.se.ee –Xmx3g –cp “\*” edu.stanford.pipeline.StanfordCoreNLP –file input.txt

- if your java is > 1.8 use the argument: -add-modules java.se.ee

- if you get a heap memory error, try –Xmx2g and higher

- without the input file argument, it creates an interactive shell; type q to exit

- the default output is xml, for text output add this flag: -outputFormat text

Look in the folder to see input.txt and the input.txt.xml or input.txt.out output files.

The input.txt file consists of two sentences: (of course, you can edit this)

Stanford University is located in California. It is a great university, founded in 1891.

The output file is quite lengthy. First there are annotations for each sentence: the tokens, a dependency parse, NER mentions. Finally, coreference sets for the entire document are listed at the end of the time.

Notice that for each token you get the offsets, POS, lemmas, and named entity.

Tokens:

[Text=Stanford CharacterOffsetBegin=0 CharacterOffsetEnd=8 PartOfSpeech=NNP Lemma=Stanford NamedEntityTag=ORGANIZATION]

### 4. Try out some test programs on the next page.

But first, let’s look at the big picture. CoreNLP contains the following (and other) tools called annotations:

* POS - part of speech tagger
* parser for sentence syntax
* NER – named entity recognition
* coreference resolution
* sentiment analysis

You specify the annotations you want in the “props” as shown below.

Properties props = new Properties();

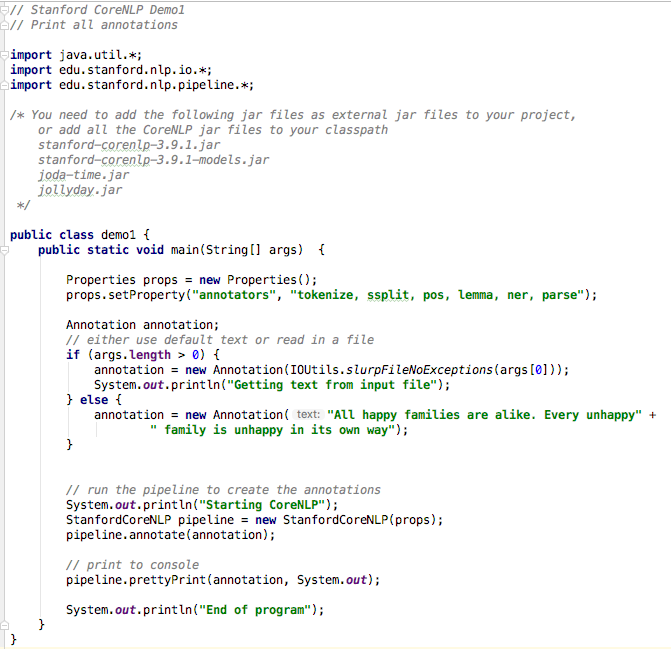
props.setProperty("annotators", "tokenize, ssplit, pos, lemma, ner, parse, dcoref, sentiment");

Be aware that running dcoref may require extra Java heap memory.

Demo 1: Basic Parsing, outputting the complete annotation to the console

* Start a new project in your IDE
* Use Java 8 not higher for this version of CoreNLP.
* This code will use some default text unless you specify some other text in a file and add this file as an argument in your run configuration.
* This code used the default annotations, but they can be changed by adding any of the following to the props object: tokenize, ssplit, pos, lemma, ner, parse,

Try the following code: (available on the github)

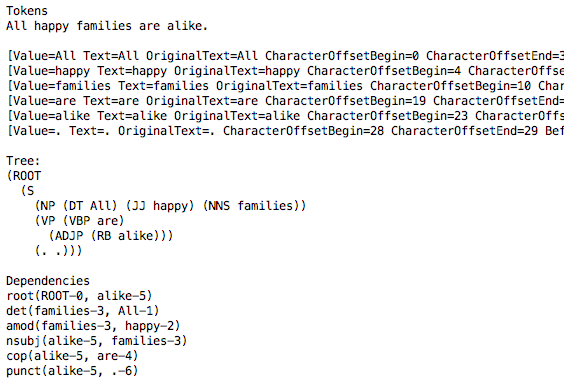


Demo 2: Basic Parsing, selecting annotations to print

The following code, also available on the github, shows how to extract individual components of the annotation. The following screen shot does not show the includes above the class definition but they are the same as in Demo 1.



We asked for token annotations, a parse tree, and the basic dependencies. Let’s look at those for the first sentence:



The screen shot above cuts off the token information, index information for the tokens as well as the lemma, part of speech, and NER info.

The tree is the parse tree, with the indents showing the structure.

The dependencies are of this form:

label(parent-index, token-index)

Troubleshooting:

- To avoid the problem mentioned in the documentation about the SUTime component, set your IDE to use Java 8 if your get errors mentioning “jollyday”.