The Stanford NLP Library

1 Downloading the Library

Using the English library for Stanford NLP is quite simple as they have bundled everything into one downloadable compressed file available from: `http://nlp.stanford.edu/software/stanford-corenlp-2012-01-08.tgz`. Just download and unzip to a convenient location.

2 Using the Library

There are two ways in which the tools from the download can be used:

2.1 In Java Code

You need at least the following `.jar` files on your class path:

- `stanford-corenlp-[version].jar`
- `stanford-corenlp-[version]-models.jar`
- `xom.jar`
- `joda-time.jar`

Once you have the required libraries loaded, the POS tagging or parsing can be done in the following steps:

1. Create the `StanfordCoreNLP` object using the following code:

   ```
   import java.util.Properties;
   import edu.stanford.nlp.pipeline.StanfordCoreNLP;
   ...
   Properties props = new Properties();
   props.put("annotators", "tokenize,ssplit,pos,lemma,parse");
   StanfordCoreNLP snlp = new StanfordCoreNLP(props);
   ```

   What you supply as the second argument to the `props.put` function depends on what you want to do with it. Generally, you would want to be frugal and use only ones that you are going to need, as each one of them tend to make processing slower. Having said that, you would almost always need `tokenize` and `ssplit` (sentence splitting), so be sure to include them. A list of possible values is available as `static` values from the `StanfordCoreNLP` class.

2. Annotate your text using the following code:

   ```
   import edu.stanford.nlp.pipeline.Annotation;
   ...
   String text = "the quick brown fox jumps over the lazy dog";
   Annotation document = new Annotation(text);
   snlp.annotate(document);
   ```

3. Finally, you can use the annotated `document` object to get specific annotations. For example, to get sentences:

   ```
   import edu.stanford.nlp.ling.CoreAnnotations.SentencesAnnotation;
   import edu.stanford.nlp.util.CoreMap;
   ...
   List<CoreMap> sentences = document.get(SentencesAnnotation.class);
   ```

   Each of these `CoreMap` objects can be used to get further details like tokens, POS tags, etc using a similar `.get(SomethingAnnotation.class)` format where `SomethingAnnotation` is one of various annotation classes available (some other useful ones are `TokensAnnotation`, `PartOfSpeechAnnotation`, and `CollapsedCCProcessedDependenciesAnnotation`). Each of these calls returns special objects that can contain further information about what you might ultimately be interested in (for example, the object returned by getting the part-of-speech of a token would return something that would have a label and a POS tag and so on).

I have also written a Java wrapper for the Stanford NLP API that abstracts out some of these low-level details, and makes it easy to just get what you need from the library. I would be happy to share it with anyone interested.

2.2 From the Command Line

If you have the JRE (Java Runtime Environment) installed, you can also run Stanford Core NLP from the command prompt. From the directory where you have unzipped the contents of the download, run the following command:

```
java -cp stanford-corenlp-2012-01-08.jar;stanford-corenlp-2011-12-
27-models.jar;xom.jar;joda-time.jar -Xmx600m edu.stanford.nlp.pipeline.
StanfordCoreNLP -annotators tokenize,ssplit,pos,lemma,parse -file
input.txt
```

This command will take the file input.txt and perform sentence splitting, tokenization, lemmatization, POS-tagging, and parsing. There are other annotators that can also be used such as ner for named-entity recognition and decoref for coreference resolution. This above command will output an XML file called input.txt.xml containing all the information requested. You can inspect the contents of this file to see what to expect from the output.

Most of this information is also available online from: http://nlp.stanford.edu/software/corenlp.shtml.