

The Stanford POS Tagger can be used via the command line or run as a GUI. The GUI interface is for demonstration purposes only; most features of the tagger can only be accessed via the command line. To

run the demonstration GUI, use:

```
java -mx200m -classpath stanford-postagger.jar edu.stanford.nlp.tagger.maxent.MaxentTaggerGUI models/left3words-wsj-0-18.tagger
```

Providing your system gives java enough memory by default, you can also run it by either double-clicking the stanford-postagger.jar file, or giving the command:

```
java -jar stanford-postagger.jar
```

The tagger has three modes: tagging, training, and testing. Tagging allows you to use a pretrained model (two English models are included) to assign part of speech tags to unlabeled text. Training allows you to save a new model based on a set of tagged data that you provide. Testing allows you to see how well a tagger performs by tagging labeled data and evaluating the results against the correct tags

To tag a file using the pre-trained bidirectional model

```
java -mx300m -classpath stanford-postagger.jar edu.stanford.nlp.tagger.maxent.MaxentTagger -model models/bidirectional-wsj-0-18.tagger -textFile sample-input.txt > sample-tagged.txt
```

Tagged output will be printed to standard out, which you can redirect as above.

To train a simple model

```
java -classpath stanford-postagger.jar edu.stanford.nlp.tagger.maxent.MaxentTagger -prop propertiesFile -model modelFile -trainFile trainingFile
```

To test a model

```
java -classpath stanford-postagger.jar edu.stanford.nlp.tagger.maxent.MaxentTagger -prop propertiesFile -model modelFile -testFile testFile
```

Tagging XML with the Stanford Log-linear Part-Of-Speech Tagger

```
java -mx300m -classpath stanford-postagger.jar edu.stanford.nlp.tagger.maxent.MaxentTagger -model models/bidirectional-wsj-0-18.tagger -xmlInput p\ 1 -textFile Wilderness.xml > wilderness-output.xml
```