Using the SENNA SRL Parser

SENNA is a very fast SRL parser, created by Ronan Collobert, et al. SENNA stands for Semantic Extraction using a Neural Network Architecture.

1. Download SENNA.

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

http://ronan.collobert.com/senna/

This download includes binaries for linux, mac, and windows. It also includes source files if you need to compile.

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 250 MB on disk.

2. Sanity Check.

From the console, type the following, replacing “senna” with senna-linux64, senna-win32.exe, or senna-osx.

./senna –posvbs

Then type in a sample sentence and see the results (ctrl-c to quit):

```
All happy families are the same.
           All         DT          B-NP          0          - (S1(S(NP*
         happy         JJ          I-NP          0          - *
      families        NNS          E-NP          0          - *)
           are        VBP          S-VP          0          - (VP*
           the         DT          B-NP          0          - (NP*
          same         JJ          E-NP          0          - *))
             .          .             O          0          - *))
```

The posvbs flag treats verb ‘be’ like any other verb. There are other flags you can read about in the senna link above.

3. Redirect to/from input and output files.

The following example took input sentences from input.txt. It is recommended to put one sentence per line in order to make the output more readable.

To print to console:

./senna –posvbs < input.txt

To redirect the output to a file:

./senna –posvbs < input.txt > output.txt

```
      Happy          JJ          B-NP          0          - (S1(S(NP*
  families         NNS          E-NP          0          - *)
        are         VBP          S-VP          0          - (VP*
        all          DT             O          0          - (ADVP*
      alike          RB       S-ADVP          0          - *))
          .           .             O          0          - *))

      Every          DT          B-NP          0          - (S1(S(NP*
  unhappy          JJ          I-NP          0          - *
     family          NN          E-NP          0          - *)
         is         VBZ          S-VP          0          - (VP*
  unhappy          JJ       S-ADJP          0          - (ADJP*)
         in          IN          S-PP          0          - (PP*
        its       PRP$          B-NP          0          - (NP*
        own          JJ          I-NP          0          - *
        way          NN          E-NP          0          - *)))
          .           .             O          0          - *))
```

Sample usage code at: https://github.com/kjmazidi/senna_parse