**ClearSRL User Guide**

**Setup**

1. Set up the environment variable CLEARSRL\_HOME to the directory of this README file
2. Install WordNet 3.0 (not included, and optionally the WordNet 3.1 data files)
   1. Download WordNet from <http://wordnet.princeton.edu/wordnet/download/current-version/>
3. Configure the properties file (prop/english-srl.properties)
   1. Set srl.wordnet\_dic=${location of WordNet dict directory}

Note: the system will parse environment variables (enclose with ${ENV\_VAR}) in the properties file values.

Ex: if WordNet is installed in /home/username/WordNet-3.0, then we can set:

srl.wordnet\_dic=${HOME}/WordNet-3.0/dict

* 1. Set the constituent parser models (files end with .gr) and SRL models (files end with .model) (parser.grammar and srl.model\_file).

2 parser and SRL model sets are included, one is trained on the official OntoNotes 5.0 training data (models/en.ontonotes5.[gr|model]), the other is trained on OntoNotes 5.0 w/ additional nominal predicate annotations and BOLT data (both verbal and nominal predicates) (models/en.ontonotes5-bolt.[gr|model])

**Running ClearSRL**

1. Set up any additional (optional) property values
   1. parser.threads: indicate the number of threads the Berkeley parser will use. More threads will speed up phrase parsing on multi-core systems (recommend no more than 1 thread per physical CPU cores).
   2. srl.run.regex: acts as a file filter (uses java regular expression convention) when performing SRL on a directory of files. Only filters the actual file name, not the sub-directory path
2. Adjust the java heap space in runsrl.sh (OPTS="-Xmx??g ...”)
   1. The model trained on the official OntoNotes 5.0 corpora requires -Xmx7g or more to run, the model trained OntoNotes 5.0 w/ additional nominal predicate annotations and BOLT data requires –Xmx11g or more to run
3. Invoke ./runsrl.sh –h to familiarize with the options:

-compressOutput : Compress the SRL output

-depIn FILE : dependency input file/directory

-format [TEXT | PROPBANK | PROPBANK\_PR : srl output format: TEXT/PROPBANK

OB | CONLL | CONLL\_DEP] :

-h : help message

-in FILE : input file/directory

-inList FILE : list of files in the input

directory to process (overwrites

regex)

-model VAL : model file to use

-out FILE : output file/directory

-outputParse : output the intermediary parse

-parsed : input is parse trees

-prop FILE : properties file

-usePBCorpus VAL : use specified PropBank corpus to

find predicates

* 1. -compressOutput compresses the SRL output in gzip format
  2. -format outputs SRL in TEXT , PropBank or CoNLL-2005 format

The text format for “John loves Mary” looks like "[ARG0 John] [rel loves] [ARG1 Mary]"

* 1. -inList a text file containing all the filenames (including subdirectory path, one name per line) within a directory to process. This overrides any regular expression based file name filter
  2. -outputParse outputs the intermediary phrase structure parse trees if the input are text files
  3. -parsed indicates the input files are parse trees (instead of sentence segmented and tokenized text files)
  4. -out the name of the output file/directory. If omitted, the system will output to the terminal STDOUT
  5. -prop the name of the property file

1. Perform SRL on the included sample file (run in the CLEARSRL\_HOME directory) to verify everything is working

./runsrl.sh -prop prop/english-srl.properties -in en-sample.txt –out en-sample.prop –outputParse

This should produce en-sample.prop in text format and en-sample.parse (the phrase structure parses of the tokenized input)