The Illinois SRL Manual

Vivek Srikumar

Contents

1 Introduction

The Illinois SRL implements the single-parse Semantic Role Labeler that is described in (Punyakanonk, et. al. 2008). Using a similar approach, it also implements a nominal SRL system for deverbal nouns in Nombank (See (Meyers 2007) for a detailed description of this class.)

This re-implementation is entirely in Java and achieves an equivalent performance on the test set of the Penn Treebank as described in the paper. Using parse trees from the Charniak parser, the original work achieves an average F1 of 76.29%. In comparison, , this re-implementation gets an F1 of 76.47% with beam search (which is comparable to the performance when ILP inference is used). The nominal SRL gets an F1 score of 66.97% with beam search.

Citing this work To come soon.

2 Installation and usage

2.1 Getting started

After downloading the archive containing the SRL system, unpack it and run `srl.sh -v -i`. This will start the verb SRL system in the interactive mode, where you can enter sentences on the command line and get it verb semantic role labels. For nominal semantic role labeling, replace `-v` with `-n`. For the first sentence alone, the system will take a long time to load the model to the memory. Subsequent sentences will be faster. Note that this system requires nearly 10 GB of RAM for verb SRL and about 5 GB for nominals.

If this works you are all set. You can now use the semantic role labeler in one of three modes: as a curator plugin, as a batch annotator and in the interactive mode.

2.2 Configuration

Most of the configuration to the SRL system can be provided via a config file. The configuration file can be specified via the command line option `-c <config-file>`. If this option is not specified, the system looks for the file `srl-config.properties` in the same directory.

Here is a summary of the configuration options:

1. *CuratorHost*: Specifies the host of the curator instance which provides the various inputs to the SRL system.

2. *CuratorPort*: Specifies the port on which the curator is listening on *CuratorHost*.

3. *DefaultParser*: This can either be `Charniak` or `Stanford`. This selects the constituent parser that provides the features for the SRL system. It is assumed that the parser corresponding to the choice here is provided by the Curator. (Note: The SRL system has been trained using the Charniak parser.)

4. *WordNetConfig*: Specifies the xml file that provides the configuration for Java WordNet Library(JWNL). An example configuration file is provided as `jwnl_properties.xml`. The path to the WordNet dictionary should be set in this file.

```
<param name="dictionary_path" value="/path/to/wordnet/dict/here"/>
```

5. *LoadWordNetConfigFromClassPath*: Specifies whether the WordNet config file specified in *WordNetConfig* should be loaded from the classpath. This property can take either `true` or `false` values. If `true`, the system will look for the WordNet configuration file in the classpath. If `false` or if the property is not present, it loads the file from the filesystem.

6. *Inference*: This can either be `BeamSearch` or `ILP` and decides the inference algorithm that is used to make the final prediction. If the choice is `BeamSearch`, in in-built beam search engine is used for inference. If the choice is `ILP`, then the Gurobi ILP solver will be used. (Note: To use ILP inference, the Gurobi engine needs to be configured.)

7. *BeamSize*: Specifies the beam size if beam search inference is chosen. Otherwise, this option is ignored.

8. *TrimLeadingPrepositions*: Should the leading prepositions of arguments be trimmed. If this is set to true, then a sentence like "John bought a car from Mary on Thursday for 2000 dollars." would be analyzed as "bought(A0:John, A1: the car, A2: Mary, A3: 2000 dollars, AM-TMP: Thursday)". If this is set to false (or if the argument is not present), then the leading prepositions are included. This gives "bought(A0:John, A1: the car, A2: from Mary, A3: for 2000 dollars, AM-TMP: on Thursday)" This option applies for both verbs and nouns.

2.3 Modes of use

For all three modes, either `-v` or `-n` argument is required to indicate verb or nominal SRL respectively.

2.3.1 As a Curator plugin

To start the SRL system as a curator plugin, run the following command:

```
./srl.sh [-v |-n ] -s <port-number> [-t <number-of-threads>]
```

The number of threads need not be specified and defaults to using one thread.

After the server starts, the curator instance can be configured to use this to serve SRL outputs. The following XML snippet should be added on to

the curator annotator descriptor file (with appropriate type, host and port entries):

```
<annotator>
  <type>parser</type>
  <field>srl</field>
  <host>srl-host:srlport</host>
  <requirement>sentences</requirement>
  <requirement>tokens</requirement>
  <requirement>pos</requirement>
  <requirement>ner</requirement>
  <requirement>chunk</requirement>
  <requirement>charniak</requirement>
</annotator>
```

2.3.2 As a batch annotator

The SRL system can be used to annotate several sentences as a batch by running it on an input file with a set of sentences. Running the SRL in this form produces a CoNLL style column format with the SRL annotation.

The following command runs the SRL in batch mode:

```
./srl.sh [-v | -n ] -b <input-file> -o <output-file> [-w]
```

Each line in the input file is treated as a separate sentence. The option `-w` indicates that the sentences in the input file are whitespace tokenized. Otherwise, the curator is asked to provide the tokenization.

2.3.3 Interactive mode

The SRL system can be used in an interactive mode by running it with the `-i` option.

3 Papers that used this software

The following papers have used an earlier version of this software:

- G. Kundu and D. Roth, *Adapting Text Instead of the Model: An Open Domain Approach.* In Proc. of the Conference of Computational Natural Language Learning, 2011.

- V. Srikumar and D. Roth, A Joint Model for Extended Semantic Role Labeling. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2011.

If you use this package, please let me know and I will add the reference to this list here.

4 References

1. V. Punyakanok, D. Roth and W. Yih, *The importance of Syntactic Parsing and Inference in Semantic Role Labeling.* Computational Linguistics, 2008.

2. A. Meyers. *Those other nombank dictionaries.* Technical report, Technical report, New York University, 2007.