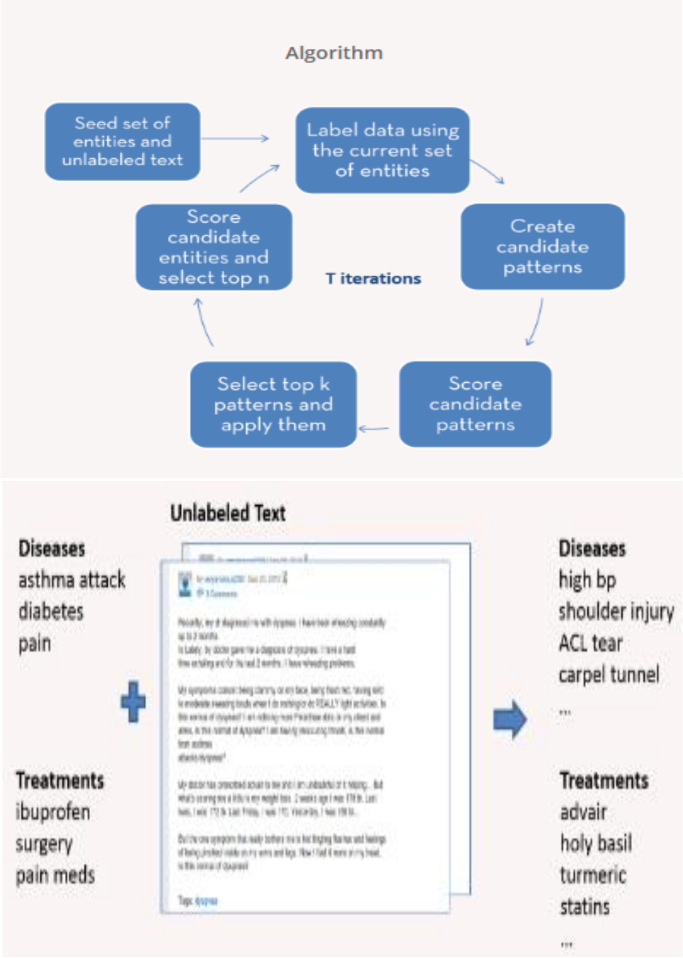
Stanford Pattern-based Information Extraction and Diagnostics (SPIED)



There are two ways of running a demo (both essentially use the same code): (1) See Usage. (2) Download SPIED-viz code from GitHub (the Github code is mainly for visualization after running pattern based entity extraction, but has scripts that download Stanford CoreNLP v3.4.1 and setup the files for running a demo.) See setupWithCoreNLP.sh anddemo.sh files.

Citation

The pattern learning system is described in:
Improved Pattern Learning for Bootstrapped Entity Extraction. Sonal Gupta and Christopher D. Manning. In Proceedings of the Eighteenth Conference on Computational Natural Language Learning (CoNLL). 2014.[pdf; Supplementary; bib]

The code also has implementations of the baseline pattern scoring measures described in the paper.

Licensing

Please refer to the license for Stanford CoreNLP.

Downloads

The pattern-based learning code can be downloaded from the Stanford CoreNLP package (version >=3.4).

Usage

1. **Download Stanford CoreNLP version >= 3.4**

   The main class is edu.stanford.nlp.patterns.GetPatternsFromDataMultiClass. An example properties file is patterns/example.properties and the example data is in the same directory. (If you are using version < 3.5.1, use edu.stanford.nlp.patterns.surface.GetPatternsFromDataMultiClass class.)

2. **Configuration**

   See the example properties file patterns/example.properties from the code distribution as a basis. Change the HOME variable. The *** symbol in the properties file tells you which settings should be adjusted to fit your system; other ones can likely be left alone. For more details on the parameters and more parameters, see the javadoc.

3. **Input**

   The input consists of a file or directory of text and files with seed sets of entities for each label. For an example, see the data in patterns directory -- in this example, we try to learn names of U.S. presidents and vice-presidents, names of their family members, and places they are related to from the text copied from the White House website.

4. **Output**

   The output files are the following, where $v means the value of the variable v in the properties file:
   Inside $outDir/$identifier/$for-each-label , files

learnedwords.txt : learned words, iterations are separated by newlines
learnedpatterns.txt : learned patterns, iterations are separated by newlines
patterns.json : output json file for visualization
words.json : output json file for viusalization
tokensmatchedpatterns.json : output json file for visualization

5. **Running**

   To run with your properties file:

```
java -cp classpath edu.stanford.nlp.patterns.GetPatternsFromDataMultiClass -props yourproperties.properties
```

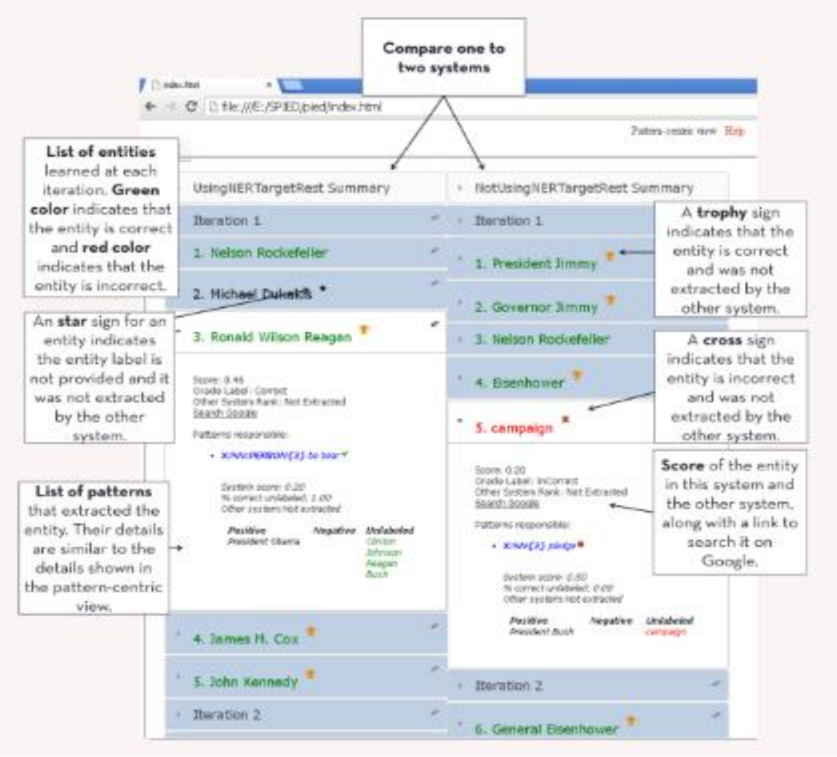
An **example** of how to run using the example data distributed with the code:

```
java -cp stanford-corenlp-3.5.1.jar:stanford-corenlp-3.5.1-models.jar:javax.json.jar:joda-time.jar:jollyday.jar edu.stanford.nlp.patterns.GetPatternsFromDataMultiClass -props patterns/example.properties
```

Visualization

The visualization is aid IE system developers in creating better IE systems efficiently and effectively. Some of the screenshots are below.

Entity centric view



Compare one to two systems

List of entities learned at each iteration. **Green color** indicates that the entity is correct and **red color** indicates that the entity is incorrect.

An **star** sign for an entity indicates the entity label is not provided and it was not extracted by the other system.

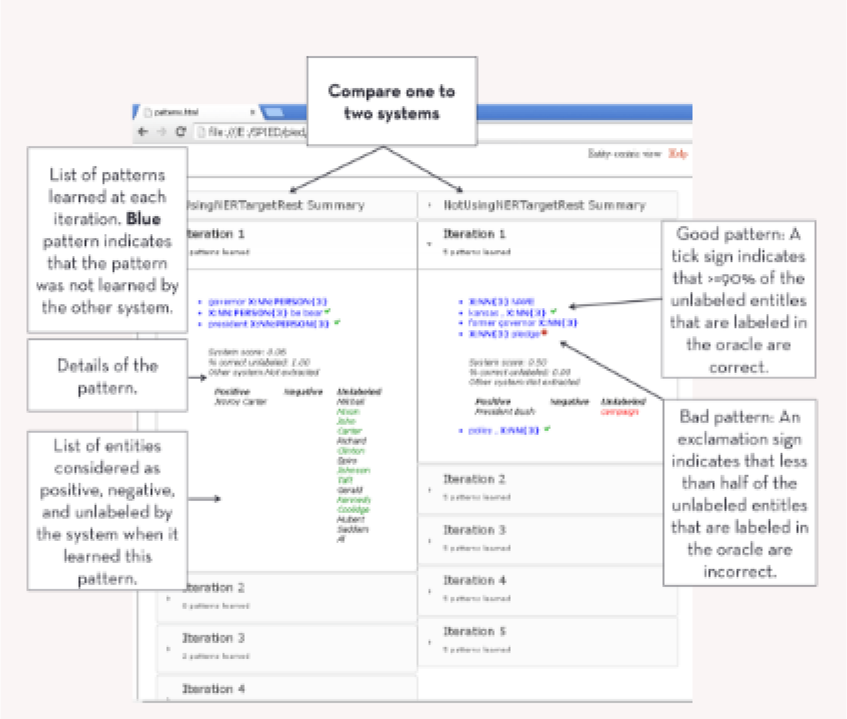
List of patterns that extracted the entity. Their details are similar to the details shown in the pattern-centric view.

A **trophy** sign indicates that the entity is correct and was not extracted by the other system.

A **cross** sign indicates that the entity is incorrect and was not extracted by the other system.

Score of the entity in this system and the other system, along with a link to search it on Google.

Pattern centric view



Compare one to two systems

List of patterns learned at each iteration. **Blue** pattern indicates that the pattern was not learned by the other system.

Details of the pattern.

List of entities considered as positive, negative, and unlabeled by the system when it learned this pattern.

Good pattern: A tick sign indicates that >=90% of the unlabeled entitles that are labeled in the oracle are correct.

Bad pattern: An exclamation sign indicates that less than half of the unlabeled entitles that are labeled in the oracle are incorrect.

An earlier version of the visual interface is described in:

Sonal Gupta and Christopher D. Manning. 2014. SPIED: Stanford Pattern-based Information Extraction and Diagnostics. In *Proceedings of the ACL 2014 Workshop on Interactive Language Learning, Visualization, and Interfaces (ACL-ILLVI)*. [pdf, bib]

SPIED-viz, the visualization part of SPIED, is licensed under the *full* GPL, which allows its use for research purposes, free software projects, software services, etc., but not in distributed proprietary software.