Instruction

Rex Liu
University of Ottawa

April 6, 2015

1 What's in the Package

1.1 src/

This folder contains JAVA source codes.

1.2 src/resources

This folder contains all the resources required to run the package, including:

1. *.ann files. Serialized annotators. city.ann annotates cities; SP.ann annotates states in US and provinces in Canada; country.ann annotates countries.
2. hyer.txt. A Gazetteer list containing information about all countries, SPs and cities.
3. model.ritter_ptb_alldata_fixed.20130723.txt. This file is required to run the Tweet POS tagger.
4. Training and testing datasets. Each dataset contains a folder of texts, and a .labels file which includes all the hand-annotated labels..
    - train/ and train.labels. The totality of all training data, containing 6000 tweets. They have been used to train the *.ann files listed above.
    - small/ and small.labels. A small proportion of all training data, containing around 1400 tweets. They are here for a faster performance evaluation experiment.
    - disam/ and disam.labels. 300 tweets not only annotated with a location type, but also with their actual locations(denoted by an ID). They are used to test the performance of the location disambiguator.

1.3 lib/

All the dependencies required to run this package. They should be added to the CLASSPATH of the application.

2 Apply annotator

To apply saved annotator, simply follow these steps:

1. Call the method **crf.ApplyModel.loadModel** to load an annotator.
2. Put the Tweets ready for annotation in an **ArrayList<String>**.
3. Call the method **crf.ApplyModel.annotate** to annotate those tweets. This method returns an **edu.cmu.minorthird.text.BasicTextLabels** object.
4. Instantiate a **gazetteer.Gazettee**r by calling its constructor with the directory to hyer.txt as parameter.
5. Call the method **crf.ApplyModel.disambiguate** to find true locations for locations annotated.
6. First, consider the Tweets a series of contiguous Spans[1] of tokens.

   The **edu.cmu.minorthird.text.BasicTextLabels** object stores information about labels of those spans. There are two kinds of labels, "type" and "property". In this case, once a span is annotated as a location, it shall have a "type", which is either "city", "SP" or "country", depending on which annotator is loaded beforehand; furthermore, once disambiguation of a span is successful, it shall have a "property", the name of the "property" being "trueLoc", the value of the "property" being the id of the location.

   For instance, the span "London" can have a "type" that is "city", and a "property" named "trueLoc" that is "62035".

   The methods that manipulate **edu.cmu.minorthird.text.BasicTextLabels** object can be found here [2]. But ideally, one only need define
   Iterator<Span> i =labels.instanceIterator("city")
   to retrieve all spans annotated as a "city"; and define
   Iterator<Span> i =labels.getSpansWithProperty("trueLoc")
   to retrieve all spans disambiguated, then

[1] http://teamcohen.github.io/MinorThird/javadoc/edu/cmu/minorthird/text/Span.html
[2] http://teamcohen.github.io/MinorThird/javadoc/edu/cmu/minorthird/text/BasicTextLabels.html

Location loc = gazetter.getByID(labels.getProperty(i.next(), "trueLoc"))
shall return a corresponding Location object.