Morpho-syntactic Analysis with the Stanford CoreNLP

Danilo Croce
croce@info.uniroma2.it

WmIR 2015/2016
Objectives of this tutorial

- Use of a Natural Language Toolkit
  - CoreNLP toolkit

- Morpho-syntactic analysis of short texts
  - Tokenization
  - Part-of-speech Tagging
  - Lemmatization
  - Named Entity Recognition
  - Dependency Parsing

- Use of such body of linguistic evidence for a task
  - Knowledge Acquisition by reading large scale corpora
The Stanford CoreNLP

- The Stanford CoreNLP is a statistical natural language parser from the Stanford Natural Language Processing Group.
  - Used to parse input data written in several languages
    - such as English, German, Arabic and Chinese

- It has been developed and maintained since 2002, from the Stanford University
  - Written in Java
  - The application is licensed under the GNU GPL, but commercial licensing is also available.

- Site: [http://stanfordnlp.github.io/CoreNLP/](http://stanfordnlp.github.io/CoreNLP/)

- A demo is available at: [http://corenlp.run/](http://corenlp.run/)
Stanford CoreNLP

- The list of processors

![Diagram showing the execution flow of processors: Tokenization, Sentence Splitting, Part-of-speech Tagging, Morphological Analysis, Named Entity Recognition, Syntactic Parsing, Coreference Resolution, Other Annotators (gender, sentiment).]
An example

- Let us consider the sentence:
  - “In 1982, Mark drove his car from Los Angeles to Las Vegas.”

Part-of-Speech:

<table>
<thead>
<tr>
<th>1</th>
<th>IN</th>
<th>1982</th>
<th>,</th>
<th>Mark</th>
<th>drove</th>
<th>his</th>
<th>car</th>
<th>from</th>
<th>Los</th>
<th>Angeles</th>
<th>to</th>
<th>Las</th>
<th>Vegas</th>
<th>.</th>
</tr>
</thead>
</table>

Lemmas:

| 1 | in | 1982 | , | Mark | drive | he | car | from | Los | Angeles | to | Las | Vegas | . |
An example

- Let us consider the sentence:
  - “In 1982, Mark drove his car from Los Angeles to Las Vegas.”

Named Entity Recognition:

<table>
<thead>
<tr>
<th></th>
<th>DATE</th>
<th>PERSON</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>In 1982, Mark drove his car from Los Angeles to Las Vegas.</td>
<td></td>
</tr>
</tbody>
</table>

Basic Dependencies:
How to use from Java

- It can be easily integrated in a Java project that support Maven, adding the following dependency to the pom.xml file

```xml
<dependencies>
  <dependency>
    <groupId>edu.stanford.nlp</groupId>
    <artifactId>stanford-corenlp</artifactId>
    <version>3.6.0</version>
  </dependency>
  <dependency>
    <groupId>edu.stanford.nlp</groupId>
    <artifactId>stanford-corenlp</artifactId>
    <version>3.6.0</version>
    <classifier>models</classifier>
  </dependency>
</dependencies>
```
A small example in JAVA

We see here how to parse a sentence and write on

// The Properties object contains the list of processors to be loaded
Properties props = new Properties();
props.put("annotators", "tokenize, ssplit, pos, lemma, ner, parse");

// Initializing the StanfordCoreNLP toolkit
StanfordCoreNLP pipeline = new StanfordCoreNLP(props);
String str = “In 1982, Mark drove his car from Los Angeles to Las Vegas”;
The CONLL format

Annotation document = new Annotation(str);
pipeline.annotate(document);
List<CoreMap> sentences = document.get(SentencesAnnotation.class);
// for each sentence
for (CoreMap sentence : sentences) {
    // Extract the dependency graph
    SemanticGraph dependencies = sentence.get(BasicDependenciesAnnotation.class);
    // For each token
    for (CoreLabel token : sentence.get(TokensAnnotation.class)) {
        IndexedWord sourceNode = dependencies.getNodeByIndex(token.index());
        IndexedWord father = dependencies.getParent(sourceNode);
        // If no father is available, the token is associated to the "root" node
        int fatherId = 0;
        String relName = "root";
        if (father != null) {
            fatherId = father.index();
            SemanticGraphEdge edge = dependencies.getEdge(father, sourceNode);
            relName = edge.getRelation().getShortName();
        }
        System.out.println(token.index() + "\t" + token.originalText() + "\t" + token.lemma() + "\t" + token.tag() + "\t" + token.ner() + "\t" + relName + "\t" + fatherId);
    }
}
System.out.println();
The CONLL tabular format

<table>
<thead>
<tr>
<th>Token ID</th>
<th>Surface</th>
<th>Lemma</th>
<th>POS</th>
<th>Namend Entity</th>
<th>Relation Name</th>
<th>Dependant</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>In</td>
<td>in</td>
<td>IN</td>
<td>O</td>
<td>case</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1982</td>
<td>1982</td>
<td>CD</td>
<td>DATE</td>
<td>nmod</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>,</td>
<td>,</td>
<td>O</td>
<td>punct</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>Mark</td>
<td>Mark</td>
<td>NNP</td>
<td>PERSON</td>
<td>rsubj</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>drove</td>
<td>drive</td>
<td>VBD</td>
<td>O</td>
<td>root</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>his</td>
<td>he</td>
<td>PRP$</td>
<td>O</td>
<td>rmod:poss</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>car</td>
<td>car</td>
<td>NN</td>
<td>O</td>
<td>obj</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>from</td>
<td>from</td>
<td>IN</td>
<td>O</td>
<td>case</td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>Los</td>
<td>Los</td>
<td>NNP</td>
<td>LOCATION</td>
<td>compound</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>Angeles</td>
<td>Angeles</td>
<td>NNP</td>
<td>LOCATION</td>
<td>rmod</td>
<td>5</td>
</tr>
<tr>
<td>11</td>
<td>to</td>
<td>to</td>
<td>TO</td>
<td>O</td>
<td>case</td>
<td>13</td>
</tr>
<tr>
<td>12</td>
<td>Las</td>
<td>Las</td>
<td>NNP</td>
<td>LOCATION</td>
<td>compound</td>
<td>13</td>
</tr>
<tr>
<td>13</td>
<td>Vegas</td>
<td>Vegas</td>
<td>NNP</td>
<td>LOCATION</td>
<td>rmod</td>
<td>5</td>
</tr>
<tr>
<td>14</td>
<td>.</td>
<td>.</td>
<td>O</td>
<td>punct</td>
<td></td>
<td>5</td>
</tr>
</tbody>
</table>

Named Entity Recognition:

In 1982, Mark drove his car from Los Angeles to Las Vegas.
### The CONLL tabular format

<table>
<thead>
<tr>
<th>Token ID</th>
<th>Surface</th>
<th>Lemma</th>
<th>POS</th>
<th>Namend Entity</th>
<th>Relation Name</th>
<th>Dependant</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>In</td>
<td>in</td>
<td>IN</td>
<td>O</td>
<td>case</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1982</td>
<td>1982</td>
<td>CD</td>
<td>DATE</td>
<td>nmod</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>,</td>
<td>,</td>
<td>O</td>
<td>O</td>
<td>punct</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>Mark</td>
<td>Mark</td>
<td>NNP</td>
<td>PERSON</td>
<td>nsubj</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>drove</td>
<td>drive</td>
<td>VBD</td>
<td>O</td>
<td>root</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>his</td>
<td>he</td>
<td>PRP$</td>
<td>O</td>
<td>nmod:poss</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>car</td>
<td>car</td>
<td>NN</td>
<td>O</td>
<td>dobj</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>from</td>
<td>from</td>
<td>IN</td>
<td>O</td>
<td>case</td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>Los</td>
<td>Los</td>
<td>NNP</td>
<td>LOCATION</td>
<td>compound</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>Angeles</td>
<td>Angeles</td>
<td>NNP</td>
<td>LOCATION</td>
<td>nmod</td>
<td>5</td>
</tr>
<tr>
<td>11</td>
<td>to</td>
<td>to</td>
<td>TO</td>
<td>O</td>
<td>case</td>
<td>13</td>
</tr>
<tr>
<td>12</td>
<td>Las</td>
<td>Las</td>
<td>NNP</td>
<td>LOCATION</td>
<td>compound</td>
<td>13</td>
</tr>
<tr>
<td>13</td>
<td>Vegas</td>
<td>Vegas</td>
<td>NNP</td>
<td>LOCATION</td>
<td>nmod</td>
<td>5</td>
</tr>
<tr>
<td>14</td>
<td>.</td>
<td>.</td>
<td>O</td>
<td>O</td>
<td>punct</td>
<td>5</td>
</tr>
</tbody>
</table>

**Basic Dependencies:**

```
1  In
  1982, Mark
  drove
  his
  car from
  Los Angeles to
  Las Vegas.
```
Final objective of this exercitation

- Let us try to navigate the parse tree
- It will provide a first form of Ontology Learning by exploiting Syntactic information
  - Acquiring simple facts from the automatic analysis of large scale corpora, e.g.
    
    ```
    PERSON - marry - PERSON
    ```

- We will extract morpho-syntactic parser in the form
  
  ```
  Subject - VERB - Direct Object
  ```

- We need a large-scale corpus already processed with CoreNLP
You will be provided with...

- A “small” selection of abstracts from Wikipedia processed with CoreNLP, already in CONLL format
  - 375K sentences
  - 9.6M of tokens

- A python script (CoreNLP_simple_reader.py) to
  - load the processed sentences stored as textual files in the CONLL format
  - navigate the resulting dependency graph
  - extract simple information
Exercise

1. Extract and count the occurrences of Nouns in the provided corpus
   - The list of Part-of-speech tags is provided here:
     https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

2. Extract and count the occurrences of mentions to Named Entities in the provided corpus
   - Focus on the PERSON category
Exercise (2)

3. Extract and count all patterns in the form
   Noun – nsubj – Verb – dobj – Noun
   keeping the lemma of the involved nouns / verbs, e.g.,
   Mark – drive – car

4. Extract and count all patterns in the form extracted at step 3, but replace the Named Entities with their types
   PERSON – drive – car

The complete list of existing dependency can be found here:
Useful Links

- CoreNLP Web Page:
  - http://stanfordnlp.github.io/CoreNLP/

- How to Download the last version of CoreNLP:

- Demo
  - http://corenlp.run/

- A description of CoreNLP:
### Alphabetical list of POS tags used in the Penn Treebank Project

<table>
<thead>
<tr>
<th>Tag</th>
<th>Description</th>
<th>Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC</td>
<td>Coordinating conjunction</td>
<td>PRP$</td>
<td>Possessive pronoun</td>
</tr>
<tr>
<td>CD</td>
<td>Cardinal number</td>
<td>RB</td>
<td>Adverb</td>
</tr>
<tr>
<td>DT</td>
<td>Determiner</td>
<td>RBR</td>
<td>Adverb, comparative</td>
</tr>
<tr>
<td>EX</td>
<td>Existential there</td>
<td>RBS</td>
<td>Adverb, superlative</td>
</tr>
<tr>
<td>FW</td>
<td>Foreign word</td>
<td>RP</td>
<td>Particle</td>
</tr>
<tr>
<td>IN</td>
<td>Preposition or subordinating conjunction</td>
<td>SYM</td>
<td>Symbol</td>
</tr>
<tr>
<td>JJ</td>
<td>Adjective</td>
<td>TO</td>
<td>to</td>
</tr>
<tr>
<td>JJR</td>
<td>Adjective, comparative</td>
<td>UH</td>
<td>Interjection</td>
</tr>
<tr>
<td>JJS</td>
<td>Adjective, superlative</td>
<td>VB</td>
<td>Verb, base form</td>
</tr>
<tr>
<td>LS</td>
<td>List item marker</td>
<td>VBD</td>
<td>Verb, past tense</td>
</tr>
<tr>
<td>MD</td>
<td>Modal</td>
<td>VBG</td>
<td>Verb, gerund or present participle</td>
</tr>
<tr>
<td>NN</td>
<td>Noun, singular or mass</td>
<td>VBN</td>
<td>Verb, past participle</td>
</tr>
<tr>
<td>NNS</td>
<td>Noun, plural</td>
<td>VBP</td>
<td>Verb, non-3rd person singular present</td>
</tr>
<tr>
<td>NNP</td>
<td>Proper noun, singular</td>
<td>VBZ</td>
<td>Verb, 3rd person singular present</td>
</tr>
<tr>
<td>NNPS</td>
<td>Proper noun, plural</td>
<td>WDT</td>
<td>Wh-determiner</td>
</tr>
<tr>
<td>PDT</td>
<td>Predeterminer</td>
<td>WP</td>
<td>Wh-pronoun</td>
</tr>
<tr>
<td>POS</td>
<td>Possessive ending</td>
<td>WP$</td>
<td>Possessive wh-pronoun</td>
</tr>
<tr>
<td>PRP</td>
<td>Personal pronoun</td>
<td>WRB</td>
<td>Wh-adverb</td>
</tr>
</tbody>
</table>