Assignment_2

March 11, 2018

1 Information Extraction - Assignment

This assignment is based on the Information Extraction lecture and the lab.

```
In [7]: import nltk
        import re
        from statistics import mode
```

```
In [2]: inputfile='../football_players.txt' #Location of the file
        buf=open(inputfile)
        list_of_doc=buf.read().split('\n')
```

2 Task 1 (10 Marks)

Write a function that takes each document and performs: 1) sentence segmentation 2) tokenization 3) part-of-speech tagging

Please keep in mind that the expected output is a list within a list as shown below.

```
In [3]: def ie_preprocess(document):
            #code goes here
            return pos_sentences
```

Run the following code to check your result for the first document (Ronaldo).

```
In [ ]: first_doc=list_of_doc[0]
        pos_sent=ie_preprocess(first_doc)
        pos_sent
```

Expected output [...[(’He’, ’PRP’), (’is’, ’VBZ’), (’a’, ’DT’), (’forward’, ’NN’), (’and’, ’CC’), (’serves’, ’NNS’), (’as’, ’IN’), (’captain’, ’NN’), (’for’, ’IN’), (’Portugal’, ’NNP’), (’.’, ’.’)], ...]

3 Task 2 (20 Marks)

Write a function that will take the list of tokens with POS tags for each sentence and returns the named entities (NE).

Hint: Use binary=True while calling NE chunk function

```
In [4]: def named_entity_finding(pos_sent):
            #Code goes here

            return named_entities

        pos_sents=ie_preprocess(list_of_doc)
        named_entity_finding(pos_sents[0])
```

Expected output ['Cristiano Ronaldo', 'Santos Aveiro', 'ComM', 'GOIH', 'Portuguese', 'Portuguese', 'Spanish', 'Real Madrid', 'Portugal']

4 Task 3 (10 Marks)

Now use the named_entity_finding() function to extract all NEs for each document.

Hint: pos_sents holds the list of lists of tokens with POS tags

```
In [ ]: def NE_flat_list_fn(pos_sents):
            NE=[]
            for pos_sent in pos_sents:
                #Single line code here. Call the funtion named_entity_finding(pos_sent) and
                        #append the result to the NE list
            #Single line code here. Flatten the list of lists to the single list NE_flat_list
            return NE_flat_list
```

5 Task 4 (40 Marks)

Write functions to extract the name of the player, country of origin and date of birth as well as the following relations: team(s) of the player and position(s) of the player.

Hint: Use the re.compile() function to create the extraction patterns

Reference: https://docs.python.org/3/howto/regex.html

```
In [ ]: def name_of_the_player(doc):
            #code goes here
            # Hint: Use the named_entity_finding() function
            return name

        def country_of_origin(doc):
            #code goes here
            return country

        def date_of_birth(doc):
            #code goes here
            return date

        def team_of_the_player(doc):
            #code goes here
            return team
```

```
    def position_of_the_player(doc):
        #code goes here
        return position
```

Execute the below command to check your fuction

```
In [ ]: date_of_birth(list_of_doc[2])
```

Expected output '5 February 1992'

6 Task 5 (10 Marks)

Write a function using the outputs from the previous functions to generate JSON-LD output as follows.

Reference: https://json-ld.org/primer/latest/

{ "@id": "http://my-soccer-ontology.com/footballer/name_of_the_player",

```
    "name": "",
    "born": "",
    "country": "",
    "position": [
        { "@id": "http://my-soccer-ontology.com/position",
            "type": ""
        }
     ]
     "team": [
        { "@id": "http://my-soccer-ontology.com/team",
            "name": ""
        }
     ]
```

}

```
In [ ]: def generate_jsonld([arg1,arg2,...]):
            #Code goes here
            #Hint: arg1,arg2,..... are the arguments you will be passing to the function
            return
```

7 Task 6 (10 Marks)

Identify one other relation (besides team and player) and write a function to extract this. Also extend the JSON-LD output accordingly.