NLP_Information Extraction_Assignment_2 FINAL VERSION

September 27, 2021

Information Extraction

[1]:
```python
import nltk
import re
nltk.download('all')
```

```
[nltk_data] Downloading collection 'all'
[nltk_data]    |
[nltk_data]    | Downloading package abc to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package abc is already up-to-date!
[nltk_data]    | Downloading package alpino to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package alpino is already up-to-date!
[nltk_data]    | Downloading package biocreative_ppi to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package biocreative_ppi is already up-to-date!
[nltk_data]    | Downloading package brown to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package brown is already up-to-date!
[nltk_data]    | Downloading package brown_tei to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package brown_tei is already up-to-date!
[nltk_data]    | Downloading package cess_cat to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package cess_cat is already up-to-date!
[nltk_data]    | Downloading package cess_esp to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package cess_esp is already up-to-date!
[nltk_data]    | Downloading package chat80 to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package chat80 is already up-to-date!
[nltk_data]    | Downloading package city_database to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package city_database is already up-to-date!
[nltk_data]    | Downloading package cmudict to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package cmudict is already up-to-date!
[nltk_data]    | Downloading package comparative_sentences to
```

```
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package comparative_sentences is already up-to-
[nltk_data]    |       date!
[nltk_data]    | Downloading package comtrans to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package comtrans is already up-to-date!
[nltk_data]    | Downloading package conll2000 to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package conll2000 is already up-to-date!
[nltk_data]    | Downloading package conll2002 to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package conll2002 is already up-to-date!
[nltk_data]    | Downloading package conll2007 to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package conll2007 is already up-to-date!
[nltk_data]    | Downloading package crubadan to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package crubadan is already up-to-date!
[nltk_data]    | Downloading package dependency_treebank to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package dependency_treebank is already up-to-date!
[nltk_data]    | Downloading package dolch to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package dolch is already up-to-date!
[nltk_data]    | Downloading package europarl_raw to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package europarl_raw is already up-to-date!
[nltk_data]    | Downloading package floresta to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package floresta is already up-to-date!
[nltk_data]    | Downloading package framenet_v15 to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package framenet_v15 is already up-to-date!
[nltk_data]    | Downloading package framenet_v17 to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package framenet_v17 is already up-to-date!
[nltk_data]    | Downloading package gazetteers to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package gazetteers is already up-to-date!
[nltk_data]    | Downloading package genesis to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package genesis is already up-to-date!
[nltk_data]    | Downloading package gutenberg to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package gutenberg is already up-to-date!
[nltk_data]    | Downloading package ieer to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package ieer is already up-to-date!
```

```
[nltk_data]    | Downloading package inaugural to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package inaugural is already up-to-date!
[nltk_data]    | Downloading package indian to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package indian is already up-to-date!
[nltk_data]    | Downloading package jeita to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package jeita is already up-to-date!
[nltk_data]    | Downloading package kimmo to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package kimmo is already up-to-date!
[nltk_data]    | Downloading package knbc to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package knbc is already up-to-date!
[nltk_data]    | Downloading package lin_thesaurus to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package lin_thesaurus is already up-to-date!
[nltk_data]    | Downloading package mac_morpho to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package mac_morpho is already up-to-date!
[nltk_data]    | Downloading package machado to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package machado is already up-to-date!
[nltk_data]    | Downloading package masc_tagged to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package masc_tagged is already up-to-date!
[nltk_data]    | Downloading package moses_sample to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package moses_sample is already up-to-date!
[nltk_data]    | Downloading package movie_reviews to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package movie_reviews is already up-to-date!
[nltk_data]    | Downloading package names to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package names is already up-to-date!
[nltk_data]    | Downloading package nombank.1.0 to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package nombank.1.0 is already up-to-date!
[nltk_data]    | Downloading package nps_chat to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package nps_chat is already up-to-date!
[nltk_data]    | Downloading package omw to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package omw is already up-to-date!
[nltk_data]    | Downloading package opinion_lexicon to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package opinion_lexicon is already up-to-date!
```

```
[nltk_data]    | Downloading package paradigms to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package paradigms is already up-to-date!
[nltk_data]    | Downloading package pil to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package pil is already up-to-date!
[nltk_data]    | Downloading package pl196x to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package pl196x is already up-to-date!
[nltk_data]    | Downloading package ppattach to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package ppattach is already up-to-date!
[nltk_data]    | Downloading package problem_reports to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package problem_reports is already up-to-date!
[nltk_data]    | Downloading package propbank to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package propbank is already up-to-date!
[nltk_data]    | Downloading package ptb to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package ptb is already up-to-date!
[nltk_data]    | Downloading package product_reviews_1 to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package product_reviews_1 is already up-to-date!
[nltk_data]    | Downloading package product_reviews_2 to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package product_reviews_2 is already up-to-date!
[nltk_data]    | Downloading package pros_cons to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package pros_cons is already up-to-date!
[nltk_data]    | Downloading package qc to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package qc is already up-to-date!
[nltk_data]    | Downloading package reuters to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package reuters is already up-to-date!
[nltk_data]    | Downloading package rte to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package rte is already up-to-date!
[nltk_data]    | Downloading package semcor to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package semcor is already up-to-date!
[nltk_data]    | Downloading package senseval to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package senseval is already up-to-date!
[nltk_data]    | Downloading package sentiwordnet to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package sentiwordnet is already up-to-date!
```

```
[nltk_data]    | Downloading package sentence_polarity to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package sentence_polarity is already up-to-date!
[nltk_data]    | Downloading package shakespeare to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package shakespeare is already up-to-date!
[nltk_data]    | Downloading package sinica_treebank to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package sinica_treebank is already up-to-date!
[nltk_data]    | Downloading package smultron to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package smultron is already up-to-date!
[nltk_data]    | Downloading package state_union to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package state_union is already up-to-date!
[nltk_data]    | Downloading package stopwords to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package stopwords is already up-to-date!
[nltk_data]    | Downloading package subjectivity to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package subjectivity is already up-to-date!
[nltk_data]    | Downloading package swadesh to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package swadesh is already up-to-date!
[nltk_data]    | Downloading package switchboard to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package switchboard is already up-to-date!
[nltk_data]    | Downloading package timit to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package timit is already up-to-date!
[nltk_data]    | Downloading package toolbox to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package toolbox is already up-to-date!
[nltk_data]    | Downloading package treebank to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package treebank is already up-to-date!
[nltk_data]    | Downloading package twitter_samples to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package twitter_samples is already up-to-date!
[nltk_data]    | Downloading package udhr to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package udhr is already up-to-date!
[nltk_data]    | Downloading package udhr2 to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package udhr2 is already up-to-date!
[nltk_data]    | Downloading package unicode_samples to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package unicode_samples is already up-to-date!
```

```
[nltk_data]    | Downloading package universal_treebanks_v20 to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package universal_treebanks_v20 is already up-to-
[nltk_data]    |       date!
[nltk_data]    | Downloading package verbnet to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package verbnet is already up-to-date!
[nltk_data]    | Downloading package verbnet3 to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package verbnet3 is already up-to-date!
[nltk_data]    | Downloading package webtext to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package webtext is already up-to-date!
[nltk_data]    | Downloading package wordnet to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package wordnet is already up-to-date!
[nltk_data]    | Downloading package wordnet_ic to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package wordnet_ic is already up-to-date!
[nltk_data]    | Downloading package words to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package words is already up-to-date!
[nltk_data]    | Downloading package ycoe to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package ycoe is already up-to-date!
[nltk_data]    | Downloading package rslp to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package rslp is already up-to-date!
[nltk_data]    | Downloading package maxent_treebank_pos_tagger to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package maxent_treebank_pos_tagger is already up-
[nltk_data]    |       to-date!
[nltk_data]    | Downloading package universal_tagset to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package universal_tagset is already up-to-date!
[nltk_data]    | Downloading package maxent_ne_chunker to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package maxent_ne_chunker is already up-to-date!
[nltk_data]    | Downloading package punkt to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package punkt is already up-to-date!
[nltk_data]    | Downloading package book_grammars to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package book_grammars is already up-to-date!
[nltk_data]    | Downloading package sample_grammars to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package sample_grammars is already up-to-date!
[nltk_data]    | Downloading package spanish_grammars to
```

```
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package spanish_grammars is already up-to-date!
[nltk_data]    | Downloading package basque_grammars to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package basque_grammars is already up-to-date!
[nltk_data]    | Downloading package large_grammars to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package large_grammars is already up-to-date!
[nltk_data]    | Downloading package tagsets to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package tagsets is already up-to-date!
[nltk_data]    | Downloading package snowball_data to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package snowball_data is already up-to-date!
[nltk_data]    | Downloading package bllip_wsj_no_aux to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package bllip_wsj_no_aux is already up-to-date!
[nltk_data]    | Downloading package word2vec_sample to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package word2vec_sample is already up-to-date!
[nltk_data]    | Downloading package panlex_swadesh to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package panlex_swadesh is already up-to-date!
[nltk_data]    | Downloading package mte_teip5 to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package mte_teip5 is already up-to-date!
[nltk_data]    | Downloading package averaged_perceptron_tagger to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package averaged_perceptron_tagger is already up-
[nltk_data]    |       to-date!
[nltk_data]    | Downloading package averaged_perceptron_tagger_ru to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package averaged_perceptron_tagger_ru is already
[nltk_data]    |       up-to-date!
[nltk_data]    | Downloading package perluniprops to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package perluniprops is already up-to-date!
[nltk_data]    | Downloading package nonbreaking_prefixes to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package nonbreaking_prefixes is already up-to-date!
[nltk_data]    | Downloading package vader_lexicon to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package vader_lexicon is already up-to-date!
[nltk_data]    | Downloading package porter_test to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package porter_test is already up-to-date!
[nltk_data]    | Downloading package wmt15_eval to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
```

```
[nltk_data]    |   Package wmt15_eval is already up-to-date!
[nltk_data]    | Downloading package mwa_ppdb to
[nltk_data]    |     C:\Users\shefa\AppData\Roaming\nltk_data…
[nltk_data]    |   Package mwa_ppdb is already up-to-date!
[nltk_data]    |
[nltk_data]  Done downloading collection all
```

[1]:
```
True
```

0.1 Task 1: Named Entity Annotation (10 Marks)

Using the IOB tagging scheme annotate all of the named entities (PERson, LOCation, ORGanisation, TIME) in the following sentence:

Wayne Rooney is a professional footballer from England who last played for Major League Soccer club D.C. United and will join Derby County in January 2020.

Edit this cell and write your annotation below the line. (Note that you don't have to write code for this task, you have to annotate it manually)

Wayne NNP B-PER Rooney NNP I-PER is VBZ O a DT O professional JJ O footballer NN O from IN O England NNP B-LOC who WP O last JJ O played VBD O for IN O Major NNP B-ORG League NNP I-ORG Soccer NNP I-ORG club NN O D.C. NNP B-ORG United NNP I-ORG and CC O will MD O join VB O Derby NNP B-LOC County NNP I-LOC in IN O January NNP B-TIME 2020 CD I-TIME . . O

0.1.1 For subsequent tasks in this assignment, you will work with the documents in `football_players.txt` to perform various information extraction tasks.

[2]:
```
# Download the text file (uncomment the line below in this cell, if not already␣
↪downloaded from Blackboard)
!curl "https://ideone.com/plain/OvwDXZ" > football_players.txt
```

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed

  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 24172  100 24172    0     0  49942      0 --:--:-- --:--:-- --:--:-- 49942
```

Read all the documents from `football_players.txt` into a list called `docs`.

[25]:
```
docs = []
docs = open("football_players.txt",encoding="utf8").readlines()
```

0.2 Task 2 (10 Marks)

Write a function that takes a document and returns a list of sentences with part-of-speech tags.

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

Hint: For this task you need to perform three steps: 1. Sentence Segmentation 1. Word Tokenization 1. Part-of-Speech Tagging

```
[26]: # The below code will take a document and returns a list of sentences with␣
       ↪part-of-speech tags
      def ie_preprocess(document):
          sentences = nltk.sent_tokenize(document) # Sentence Segmentation
          tokenized_sentences = [nltk.word_tokenize(sent) for sent in sentences] #␣
       ↪Word Tokenization
          pos_sentences = [nltk.pos_tag(sent) for sent in tokenized_sentences] #␣
       ↪Part-of-Speech Tagging
          return pos_sentences
```

Run the cell below to verify your result for the second sentence in the first document. Expected output: `[('He', 'PRP'), ('is', 'VBZ'), ('a', 'DT'), ('forward', 'NN'), ('and', 'CC'), ('serves', 'NNS'), ('as', 'IN'), ('captain', 'NN'), ('for', 'IN'), ('Portugal', 'NNP'), ('.', '.')]`

```
[27]: first_doc = docs[0]
      tagged_sentences = ie_preprocess(first_doc)
      tagged_sentences[1]
```

```
[27]: [('He', 'PRP'),
       ('is', 'VBZ'),
       ('a', 'DT'),
       ('forward', 'NN'),
       ('and', 'CC'),
       ('serves', 'NNS'),
       ('as', 'IN'),
       ('captain', 'NN'),
       ('for', 'IN'),
       ('Portugal', 'NNP'),
       ('.', '.')]
```

0.3 Task 3 (20 Marks)

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

Hint: Use `binary = True` while calling NE chunk function

```
[28]: # find_named_entities(sent) will take a list of tokens with POS tags for a␣
       ↪sentence and returns a list of named entities (NE)
      # as named_entities
      def find_named_entities(sent):
          named_entities = []
          tree = nltk.ne_chunk(sent,binary=True)
          #print(tree)
          for subtree in tree.subtrees():
```

```
        if subtree.label()=='NE':
            data=""
            for leaf in subtree.leaves():
                data = data + leaf[0] + " "
            named_entities.append(data.strip())
    return named_entities
```

Run the cell below to verify your result for the first sentence in the first document. Expected output: `['Cristiano Ronaldo', 'Santos Aveiro', 'ComM', 'GOIH', 'Portuguese', 'Portuguese', 'Spanish', 'Real Madrid', 'Portugal']`

[29]:

```
tagged_sentences = ie_preprocess(docs[0])
find_named_entities(tagged_sentences[0])
```

[29]:

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

0.4 Task 4 (5 Marks)

Implement the `find_all_named_entities` function below to find **all** NEs in a given document.

Hint: Use `find_named_entities` implemented above for this task.

[30]:

```
# This would fetch all the named the named entities in a given document.
def find_all_named_entities(doc):
    named_entities = []
    tagged_sentences = ie_preprocess(doc)
    for i in tagged_sentences:
        for j in find_named_entities(i):
            named_entities.append(j)
    return named_entities   # return a flat list and not a list of lists
```

How many named entities did you find in the first document?

[31]:

```
# number of named entity in the first document
number_named_entities = len(find_all_named_entities(docs[0]))
print("Number of named_entities in the first document",number_named_entities)
```

```
Number of named_entities in the first document 56
```

0.5 Task 5 (5 Marks)

Find named entities across **all** documents in `football_players.txt`, and save the result into a single flat list.

[34]:

```python
# Iterated over the find_all_named_entities with docs as the input to fetch all␣
 ↪the named entities across all the documents.
all_named_entities = []
for z in docs:
    all_named_entities.extend(find_all_named_entities(z))
print(all_named_entities)
```

```
['Cristiano Ronaldo', 'Santos Aveiro', 'ComM', 'GOIH', 'Portuguese',
'Portuguese', 'Spanish', 'Real Madrid', 'Portugal', 'Portugal', 'Ballon',
'FIFA', 'FIFA Ballon', 'Ronaldo', 'Ronaldo', 'Portuguese', 'Portuguese Football
Federation', 'European Golden Shoe', 'ESPN', 'Ronaldo', 'Manchester United',
'England', 'United', 'UEFA Champions League', 'FIFA Club', 'Ballon', 'FIFA',
'Manchester United', 'Madrid', 'Spain', 'Ronaldo', 'UEFA Champions League',
'Ronaldo', 'La Liga', 'Ronaldo', 'UEFA Champions League', 'Real Madrid', 'La
Liga', 'Lionel Messi', 'Ronaldo', 'Portugal', 'Portugal', 'European', 'FIFA
World Cups', 'Portuguese', 'Portugal', 'Portugal', 'Portugal', 'Ronaldo', 'UEFA
European', 'European', 'Michel Platini', 'Ronaldo', 'Portugal', 'France',
'Silver Boot', 'Lionel Andrés', 'Spanish', 'Spanish', 'Argentina', 'Messi',
'FIFA Ballons', 'European Golden Shoes', 'Messi', 'La Liga', 'La Liga', 'Copa
América', 'Born', 'Argentina', 'Messi', 'Spain', 'Barcelona', 'Barcelona',
'Messi', 'Ballon', 'FIFA', 'Barcelona', 'Spanish', 'Messi', 'Ballon', 'FIFA',
'Messi', 'FIFA Ballons', 'European', 'Barcelona', 'Ballon', 'Cristiano Ronaldo',
'Messi', 'Champions League', 'Barcelona', 'European', 'Messi', 'FIFA World Youth
Championship', 'Golden Ball', 'Golden Shoe', 'Diego Maradona', 'Messi', 'FIFA',
'Argentina', 'Golden Ball', 'Neymar', 'Silva Santos Júnior', 'Portuguese',
'Neymar', 'Neymar Jr.', 'Brazilian', 'Spanish', 'Brazil', 'Neymar', 'Santos',
'Campeonato Paulista', 'Brasil', 'Santos', 'Neymar', 'South American
Footballer', 'Europe', 'Barcelona', 'Barça', 'Lionel Messi', 'Luis Suárez',
'UEFA Champions League', 'FIFA Ballon', 'Brazil', 'Neymar', 'Brazil', 'South
American Youth Championship', 'FIFA Confederations Cup', 'Golden Ball', 'FIFA',
'Brazil', 'Neymar', 'Brazil', 'Santos', 'SportsPro', 'ESPN', 'Ronaldo',
'Ronaldinho', 'Brazilian', 'Portuguese', 'Ronaldinho Gaúcho', 'Brazilian',
'Spanish', 'FIFA', 'Ronaldinho', 'Brazilian', 'Ronaldo', 'Brazil', 'Ronaldo',
'Europe', 'Ronaldinho', 'Ronaldinho', 'Brazil', 'FIFA World', 'Korea', 'Japan',
'Ronaldo', 'Rivaldo', 'FIFA', 'Ronaldinho', 'Brazilian', 'European', 'Paris',
'Barcelona', 'Milan', 'Brazil', 'Flamengo', 'Atlético Mineiro', 'Mexico',
'Querétaro', 'Barcelona', 'UEFA Champions League', 'FIFA', 'Ballon',
'Ronaldinho', 'Pelé', 'FIFPro World', 'Wayne Mark Rooney', 'England', 'Rooney',
'Everton', 'Merseyside', 'Manchester United', 'Rooney', 'United', 'UEFA
Champions League', 'FIFA Club', 'Football League', 'Rooney', 'Rooney',
'England', 'England', 'UEFA Euro', 'European Championship', 'Rooney', 'England
Player', 'Rooney', 'England', 'David Beckham', 'Rooney', 'England', 'Rooney',
'PFA Players', 'FWA Footballer', 'Steven Gerrard', 'FIFA Ballon', 'FIFPro',
'Rooney', 'Season', 'BBC', 'Zlatan Ibrahimović', 'Swedish', 'Bosnian',
'Swedish', 'Sweden', 'Ibrahimović', 'Malmö FF', 'Ajax', 'Juventus', 'Serie',
'David Trezeguet', 'UEFA Team', 'Ibrahimović', 'Barcelona', 'Serie', 'Milan',
'Paris', 'PSG', 'Ibrahimović', 'France', 'PSG', 'PSG', 'Ibrahimović', 'Swedish',
'Sweden', 'FIFA World Cups', 'Guldbollen', 'Golden Ball', 'Swedish', 'Marco',
'Basten', 'Ibrahimović', 'Sweden', 'England', 'FIFA Puskás Award', 'Goal',
'Ibrahimović', 'Guardian', 'Lionel Messi', 'Cristiano Ronaldo', 'Swedish',
'Dagens Nyheter', 'Swedish', 'Björn Borg', 'David Robert Joseph Beckham', 'OBE',
'Preston North End', 'Real Madrid', 'Milan', 'LA Galaxy', 'Paris', 'England',
'Wayne Rooney', 'England', 'Spain', 'United States', 'France', 'Beckham',
'FIFA', 'Beckham', 'Manchester United', 'United', 'UEFA Champions League', 'Real
Madrid', 'Beckham', 'Major League Soccer', 'Galaxy', 'Italy', 'Milan',
'British', 'Beckham', 'England', 'FIFA', 'European', 'Beckham', 'Victoria
Beckham', 'UNICEF UK', 'David Beckham', 'UNICEF Fund', 'MLS', 'Beckham',
'Miami', 'Mesut Özil', 'German', 'Turkish', 'German', 'English', 'German',
'Özil', 'German', 'FIFA', 'Golden Ball Award', 'Özil', 'Werder Bremen', 'Real
Madrid', 'FIFA', 'Germany', 'Arsenal', 'German', 'Özil', 'José Mourinho', 'Real
Madrid', 'Zinedine Zidane', 'Özil', 'European', 'La Liga', 'Özil', 'FIFA', 'UEFA
Euro', 'Gareth Frank Bale', 'Spanish', 'Real Madrid', 'Wales', 'Bale',
'Southampton', 'Bale', 'Tottenham Hotspur', 'Tottenham', 'Harry Redknapp',
'Bale', 'PFA Players', 'UEFA Team', 'PFA Young Player', 'FWA Footballer',
'Bale', 'Madrid', 'Cristiano Ronaldo', 'Bale', 'Real Madrid', 'UEFA Champions
League', 'UEFA Super Cup', 'FIFA Club', 'UEFA Squad', 'Season', 'ESPN', 'Bale',
'Bale', 'Wales', 'Ian Rush', 'Wales', 'UEFA Euro', 'Andrés Iniesta Luján',
'Spanish', 'Spanish', 'FC Barcelona', 'Spain', 'Barcelona', 'Iniesta',
'Barcelona', 'Iniesta', 'Barcelona', 'Spanish', 'Iniesta', 'Spain', 'Spain',
'Iniesta', 'Spanish', 'Netherlands', 'Match', 'Iniesta', 'Spain', 'Italy',
'Iniesta', 'UEFA Team', 'FIFA World XI', 'Iniesta', 'UEFA Best Player',
'Europe', 'IFFHS World', 'Lionel Messi', 'FIFA Ballon']
```

How many named entities did you find across all documents?

[35]:

```
len_all_named_entities = len(all_named_entities)
print("Length of all the named entites ",len_all_named_entities)
```

```
Length of all the named entites  380
```

0.6 Task 6 (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

[51]:

```
import re
# using re.findall to find the pattern that matches the given regex.

# name_of_the_player function will fecth the name of the player
```

```
'''
    AIM: is to find the name_of_the_player from a document.
    Approach Used: In each document the name of the player is the string from␣
 ↪starting of the document until open bracket.
    Regex used: .+? will fetch all characters and stops the serach untill it␣
 ↪finds an open bracket(, used .+?(?=\()
    The required output is in the first element of the list, so will fetch the␣
 ↪index 0 of the list name
Output: This is succesfully fetching the full name of the players in each␣
 ↪document.
'''
def name_of_the_player(doc):
    name_list = re.findall(r'.+?(?=\()',doc)
    name = name_list[0]
    return name

# country_of_origin will fetch us country of origin of the players
'''
    Aim: is to find the country_of_origin from a document.
    Approach Used: The name of the country can be fetched by fetching the word␣
 ↪before national team, as each player plays for
    the national team.
    Method used: used re.findall to find the regex [a-zA-Z]+\snational\steam␣
 ↪which will fetch any alphabatic(one or more than
    characters) followed by the national team. This will fetch a list with the␣
 ↪country name and national team as the elements.
    To get the country name alone, I have split the list and fetched the index␣
 ↪0 which contains the country.
Output: This is successfully giving me the country of the origin as the output.
'''
def country_of_origin(doc):
    country_list = re.findall(r'[a-zA-Z]+\snational\steam',doc)
    country = country_list[0].split()[0]
    return country

# date_of_birth will fetch us date of birth of the players
''' Aim: is to find the date of birth of the player.
    Approach used: Since date of birth will have 1 or 2 digits followed with␣
 ↪month and year(YYYY). I have used regex to find the
    pattern.
    Method used : To fecth the date of bith I have used the regex
        [\d]{1,2} - any digit 0-9 which can be a single digit or double digit
        [ADFJMNOS]\w* - to fectch the month of the year
        [\d]{4} - it will fecth the year in (YYYY) format
        It will fetch all the patterns that matches with the regex in a␣
 ↪document, but the date of birth is in index[0].
```

```
        Hence, used date_list[0].
Output: Successfully fetching the date of the birth of the documents.
'''
def date_of_birth(doc):
  # your code goes here
    date_list = re.findall(r'[\d]{1,2} [ADFJMNOS]\w* [\d]{4}',doc)
    date = date_list[0]
    return date

# team_of_the_player will fetch us the teams of the players
'''
Aim: is to find the team_of_the_player.
Approach used  : There are one or more teams and atleast one national team␣
 ↪associated with each of the players.
                The teams apart form the national team are being fetched by␣
 ↪using Positive lookahead and positive lookbehind
                assertion. All the team name are starting from both or for and␣
 ↪ending at and|after|. I have written a regex that
                will capture the strings between these starting and ending␣
 ↪words.

Positive lookahead assertion d(?=r)..matches a d only if is followed by r, but␣
 ↪r will not be part of the overall regex match.
Positive lookbehind asseertion (?<=r)d matches a d that immediately precedes␣
 ↪the current position in the string is r.

(?<=both|for\s)- will fetch the string that precedes the word both|for and
(.*?)(?=and|after|\.) - will fetch all the strings until it reaches a and␣
 ↪|after | .  and stores in team_list_1.

now, For the national team, I have called the country_of_origin function and␣
 ↪addaed a string after that as the pattern for national
team is name_of_the_country followed by national team and stored in team_list_2.

For fecting all the teams, I have fetched the first index of  team_list_1 and␣
 ↪concatinated using an and with the team_list_2.

Output: Able to fetch team name for all the documnets except for one document␣
 ↪d[4], where I am getting only half of the result.
'''
def team_of_the_player(doc):
    team_list_1 = re.findall(r'(?<=both|for\s)(.*?)(?=and|after|\.)',doc)
    team_list_2 = country_of_origin(doc) + " "+ "national team"
    team = team_list_1[0] + "and "+ team_list_2
    return team
```

```
# position_of_the_player will fetch the position of the players.
'''
    Approach Used: All the documents have position of the player from these␣
 ↪forward,striker,central midfielder,attacking midfielder,
    winger,right winger. There are only limited positions available in the␣
 ↪whole document.
    so I have made a regex and used re.findall which will have fetch the␣
 ↪position of the players from the above list.
    Regex: 'forward|striker|central midfielder|attacking␣
 ↪midfielder|winger|right winger'
    It can fetch more than one position also, if there are.
    Since, few documents can have more than positions or the same position name␣
 ↪for more than once, I have stored the answer in
    a set and then returned the string(using join).
Output: This is fetching the position(s) of the player for all of the documents.
 ↪
'''
def position_of_the_player(doc):
    position = ""
    position_list = re.findall(r'forward|striker|central midfielder|attacking␣
 ↪midfielder|winger|right winger',doc)
    position  = ",".join(set(position_list))
    return position
```

Execute the cell below to verify the `date_of_birth` function for the third player. Expected output `5 February 1992`

[46]:

```
date_of_birth(docs[2])
```

[46]:

```
'5 February 1992'
```

0.7 Task 6 (10 Marks)

Identify one other relation (besides team and player) and write a function to extract it.

[48]:

```
'''
Aim : Each player has bagged some achievement in his carrer; like winning a␣
 ↪trophey, best player etc.
      Im trying to find out the achievement which have been win/won or gathered␣
 ↪by the players in each document.

Approach: In order to extract the right information I have used (?
 ↪<=win\s|won\s)(.*?)(?=\.) which will fetch string starting
from either win, won and goes until full stop(?=\.). On callilng the function␣
 ↪it will print the achievement one by one with index.
For those players who have no major achievement the regex will return an empty␣
 ↪string and the function will return a string saying
```

```
that it has No major achievement were won by the [name_of_the_player].

Output: This is fetching us all the achievement bagged by a player in an order.
        There is one player docs[7], who hasnt won any awards hence the␣
 ↪function will return No major achievement bagged by
        followed by name of the player.
'''
def players_achievement(doc):
    Win = re.compile(r'(?<=win\s|won\s)(.*?)(?=\.)')
    list_1 = re.findall(Win, doc)
    if list_1 ==[]:
        return "No major achievement bagged by " + name_of_the_player(doc)
    else:
        for i in range(len(list_1)):
             print(i,list_1[i])
```