import java.util.\*;

import edu.stanford.nlp.hcoref.CorefCoreAnnotations;

import edu.stanford.nlp.hcoref.data.CorefChain;

import edu.stanford.nlp.ling.CoreAnnotations;

import edu.stanford.nlp.ling.CoreLabel;

import edu.stanford.nlp.pipeline.Annotation;

import edu.stanford.nlp.pipeline.StanfordCoreNLP;

import edu.stanford.nlp.semgraph.SemanticGraph;

import edu.stanford.nlp.semgraph.SemanticGraphCoreAnnotations;

import edu.stanford.nlp.trees.Tree;

import edu.stanford.nlp.trees.TreeCoreAnnotations;

import edu.stanford.nlp.util.CoreMap;

import java.util.List;

import java.util.Map;

import java.util.Properties;

import java.io.\*;

import org.apache.commons.io.FileUtils;

public class que\_ans{

public static void main(String args[]) {

// creates a StanfordCoreNLP object, with POS tagging, lemmatization, NER, parsing, and coreference resolution

Properties props = new Properties();

props.setProperty("annotators", "tokenize, ssplit, pos, lemma, ner, parse, dcoref");

StanfordCoreNLP pipeline = new StanfordCoreNLP(props);

// read some text in the text variable

// String text = "This is a sample text"; // Add your text here!

String nextLine = null;

try {

nextLine = FileUtils.readFileToString(new File("data\\data.txt"));

} catch (Exception e) {

e.printStackTrace();

}

System.out.println(nextLine);

// create an empty Annotation just with the given text

Annotation document = new Annotation(nextLine);

// run all Annotators on this text

pipeline.annotate(document);

// these are all the sentences in this document

// a CoreMap is essentially a Map that uses class objects as keys and has values with custom types

List<CoreMap> sentences = document.get(CoreAnnotations.SentencesAnnotation.class);

ArrayList list1 = new ArrayList();

ArrayList list2 = new ArrayList();

ArrayList list3 = new ArrayList();

for (CoreMap sentence : sentences) {

// traversing the words in the current sentence

// a CoreLabel is a CoreMap with additional token-specific methods

for (CoreLabel token : sentence.get(CoreAnnotations.TokensAnnotation.class)) {

// System.out.println("\n" + token);

// this is the text of the token

String word = token.get(CoreAnnotations.TextAnnotation.class);

// System.out.println("-----------------Text Annotation-----------------");

System.out.println(token + ":" + word);

// this is the POS tag of the token

// this is the NER label of the token

String ne = token.get(CoreAnnotations.NamedEntityTagAnnotation.class);

//System.out.println("-----------------NER-----------------");

System.out.println(token + ":" + ne);

System.out.println("\n\n");

if (ne.contentEquals("MISC")) {

list1.add(token);

} else if (ne.contentEquals("PERSON")){

list2.add(token);

} else if (ne.contentEquals("LOCATION")) {

list3.add(token);

} else {

}

}

}

BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

System.out.println("enter Question");

String que = null;

try {

que = reader.readLine();

} catch (IOException e) {

e.printStackTrace();

}

//System.out.println("enter Question is " + que);

Properties props1 = new Properties();

props1.setProperty("annotators", "tokenize, ssplit, pos, lemma, ner, parse, dcoref");

StanfordCoreNLP pipeline1 = new StanfordCoreNLP(props1);

Annotation document1 = new Annotation(que);

// run all Annotators on this text

pipeline.annotate(document1);

List<CoreMap> sentences1 = document1.get(CoreAnnotations.SentencesAnnotation.class);

ArrayList who = new ArrayList();

ArrayList where = new ArrayList();

ArrayList noun = new ArrayList();

for (CoreMap sentence : sentences1) {

{

for (CoreLabel token1 : sentence.get(CoreAnnotations.TokensAnnotation.class)) {

//System.out.println(token1);

String pos = token1.get(CoreAnnotations.PartOfSpeechAnnotation.class);

//System.out.println("--------------POS------------------");

// System.out.println(token1 + ":" + pos);

//System.out.println("\n\n");

if (pos == "WP" )//who what

{

who.add("WP");

who.add(token1);

}

else if (pos.contentEquals( "WRB") )//where

{

where.add("WRB");

where.add(token1);

}

else if (pos == "NN" )//where

{

noun.add("WRB");

noun.add(token1);

}

}

}

}

if (who.size()==noun.size()&&who.size()!=0){

System.out.println("the main charecters are "+ list2.get(1)+ ";"+list2.get(2));

}

else if (where.size()>0){

System.out.println("It happened at "+ list3.get(1));

}

}

}