CS 6320 – Natural Language Processing
Spring 2020
Dr. Mithun Balakrishna
Course Project

A. Project Steps and Deadlines:

- **Project Group Formation**:
    - Due by **Friday, March 13th 2020, 11:59pm**
    - A maximum of two (2) students per project group
    - The group should decide on an appropriate group name
    - One group member should submit a document containing the group name and the group member information i.e. Group name and Group member names, via eLearning
        - Please name the document following the convention "ProjectGroupInfo-GROUPNAME.pdf", where GROUPNAME is your project group's name.
        - Submit the document to the "Group Information Submission" assignment inside the "Final Project" folder listed in the course home page on eLearning.
        - Students that want to work on the project individually should also submit this document
    - Students that need help to form a group should meet the Instructor on **Friday, March 13th 2020** at **6:00pm** in the class room (ECSS 2.206)
        - Students that want to work on the project individually do NOT need to do this
- **Project Demo**:
    - Due date: **TBA**
    - Demo sign-up details: **TBA**
    - Submit your project source code and report via eLearning before your group's allocated demo session:
        - One group member should submit a single zip file containing the following via eLearning:
            - Project source code/script file(s)
            - A ReadMe file with instructions on how to access the project demo
            - Project report in PDF or MS Word document format.
        - Please name the zip archive document following the convention "ProjectFinalSubmission-GROUPNAME.zip", where GROUPNAME is your project group's name.
        - Submit the document to the "Project Final Submission" assignment inside the "Final Project" folder listed in the course home page on eLearning.

- Please hand over a hard copy of the project report before the start of your group’s demo session with the TA

B. Project Report

Please write a project report (5 to 10 pages) with the following details:

- Problem description
- Proposed solution
- Full implementation details
  - Programming tools (including third party software tools used)
  - Architectural diagram
  - Results and error analysis (with appropriate examples)
  - A summary of the problems encountered during the project and how these issues were resolved
  - Pending issues
  - Potential improvements

C. Project Description:

For the project, you need to implement an Information Extraction application using NLP features and techniques:

Training

Input:

- 30 text articles:
  - 10 articles related to Organizations
  - 10 articles related to Persons
  - 10 articles related to Locations
- Set of information templates
  - Template #1:

    BUY(Buyer, Item, Price, Quantity, Source)
  - Template #2:

    WORK(Person, Organization, Position, Location)
  - Template #3:

    PART(Location, Location)

Testing/Runtime

Input:

- Text article

Output:

- All instances (i.e. filled) of the above three templates found in the input text article
  - Example #1:

    Document: *Amazon_com.txt*

    Sentence(s)*: In 2017, Amazon acquired Whole Foods Market for US$13.4 billion, which vastly increased Amazon's presence as a brick-and-mortar retailer.*

    Extracted Template*: BUY(“Amazon”, “Whole Foods Market”, “US$13.7 billion”, “”, “”)*

- Example #2:

  Document: *AppleInc.txt*

  Sentence(s)*: Steven Paul Jobs (; February 24, 1955 – October 5, 2011) was an American business magnate and investor. He was the chairman, chief executive officer (CEO), and co-founder of Apple Inc.; chairman and majority shareholder of Pixar; a member of The Walt Disney Company's board of directors following its acquisition of Pixar; and the founder, chairman, and CEO of NeXT.*

  Extracted Template*: WORK("Steven Paul Jobs", "Apple Inc.", "chairman ; chief executive officer (CEO); co-founder", "")*

  Extracted Template*: WORK("Steven Paul Jobs", "Pixar", "chairman", "")*

  Extracted Template*: WORK("Steven Paul Jobs", "The Walt Disney Company", "board member", "")*

  Extracted Template*: WORK("Steven Paul Jobs", "NeXT", "founder; chairman ; CEO", "")*

- Example #3:

  Document: *Richardson_Texas.txt*

  Sentence(s)*: Richardson is a principal city in Dallas and Collin counties in the U.S. state of Texas.*

  Extracted Template*: PART("Richardson", "Dallas")*

  Extracted Template*: PART("Richardson", "Collin counties")*

  Extracted Template*: PART("Richardson", "U.S. state of Texas | Texas")*

  Extracted Template*: PART("Texas", "U.S.")*

- Output Format (JSON): Please see the sample.json file in the "Projects" folder in eLearning.

The following are the tasks that need to be performed:

Task 1: Implement a deep NLP pipeline to extract the following NLP based features from the text articles/documents:

- Split the document into sentences
- Tokenize the sentences into words
- Lemmatize the words to extract lemmas as features
- Part-of-speech (POS) tag the words to extract POS tag features
- Perform dependency parsing or full-syntactic parsing to get parse-tree based patterns as features
- Using WordNet, extract hypernymns, hyponyms, meronyms, AND holonyms as features
- Some additional features that you can think of, which may make your representation better

Note: you are free to implement or use a third-party tool. Some useful resources are provided at the end of this document.

Task 2: Implement a machine-learning, statistical, or heuristic (or a combination) based approach to extract filled information templates from the corpus of text articles:

- Run the above described deeper NLP on the corpus of text articles and extract NLP features
- Implement a machine-learning, statistical, or heuristic (or a combination) based approach to extract filled information templates from the corpus of text articles

Task 3: Implement a program that will accept an input text document and:

- Run the above described deep NLP on the input text document
- Extract information templates from the input text document using your information extraction approach implemented in Task 2
- Output a JSON file with extracted/filled information templates from the input text document

Performance Evaluation: The performance of your NLP and Information Extraction system will evaluated on an **unseen** test corpus of text articles.

D. Project Point Distribution

1. Max points available: 100 points
2. Division of points:

a. Group information: 2 points
b. Project implementation and demo: 90 points
   i. Task 1: 30 points
   ii. Task 2: 30 points
   iii. Task 3: 5 points
   iv. Performance Evaluation: 25 points
      1. Max Score: 100
      2. Teams with score >= 95: 25 points
      3. Teams with score between 90 and 94: 22.5 points
      4. Teams with score between 85 and 89: 20 points
      5. Teams with score between 80 and 84: 17.5 points
      6. Teams with score between 75 and 79: 15 points
      7. Teams with score between 70 and 74: 12.5 points
      8. Rest: 10 points
c. Project Report: 8 points

E. Useful resources

Some resources that you may find useful for this project are listed below:

- TextBlob: Python API for common NLP tasks
- spaCy: Python API commonly used in the industry
- NLTK: Python API for common NLP tasks
- PyTorch: Python library for deep learning
- TensorFlow: Another more common Python library for deep learning
- Stanford NLP: Java tool for common NLP tasks
- OpenNLP: Java tool that provides machine learning libraries for NLP tasks
- MIT-IE toolkit: C, C++ and Python tools for Information Extraction
- Charniak Parser: C++ implementation of the Charniak parser