CS 6320 – Natural Language Processing
Spring 2019
Dr. Mithun Balakrishna
Course Project
Updated On: 04/25/2019

A. Project Steps and Deadlines:

- **Project Group Formation**:
    - Due by ~~**Friday, March 15th 2019, 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 "Project 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 15th 2019** at **6pm** in the class room (ECSS 2.306)

- **Project Submission**:
    - Project Submission Deadline: **Wednesday, May 8th 11:59pm**
    - Submit your project source code and report via eLearning:
        - 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 configure, build and run the project
            - 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.

- **Project Demo**:
    - Demo dates: **May 9th 2019 and May 10th 2019**
    - 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 a Question Answering (QA) system using NLP features and techniques for the following Question Types:

1. WHO questions:
   a. Examples:
      i. Who founded Apple Inc.?
      ii. Who supported Apple in creating a new computing platform?
2. WHEN questions:
   a. Examples:
      i. When was Apple Inc. founded?
      ii. When did Apple go public?
3. WHERE questions:
   a. Examples:
      i. Where is Apple's headquarters?
      ii. Where did Apple open its first retail store?

The following data will be provided to the students:

1. 30 Wikipedia articles:
   a. 10 Wikipedia articles related to Organizations
   b. 10 Wikipedia articles related to Persons
   c. 10 Wikipedia articles related to Locations

   The QA system will process and answer questions on this data
2. 20 question and answer pairs for QA system development process

QA system requirements:

- Input: natural language question
- Output:
  a. Exact answer phrase(s)
  b. Supporting sentence(s) in Wikipedia document
  c. Supporting Wikipedia document name(s)

The following are the tasks that need to be performed:

1. **Task 1**: Implement a deeper NLP pipeline to extract **at least** the following NLP based features from the Wikipedia documents and natural language questions:

   - Tokenize text into sentences and 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 parse-tree based patterns as features

- Using WordNet, extract hypernymns, hyponyms, meronyms, AND holonyms as features

Note: you are free to implement or use a third-party tool such as:

1. NLTK: http://www.nltk.org/
2. Stanford NLP: http://nlp.stanford.edu/software/corenlp.shtml
3. Apache OpenNLP: http://opennlp.apache.org/

2. **Task 2**: Implement a QA system to extract relevant sentence(s) and exact answer(s) for a natural language question from the processed Wikipedia documents:

   - Run the above described deeper NLP on the Wikipedia documents and extract NLP features

   - Run the above described deeper NLP on the natural language question and extract NLP features

   - Implement a machine-learning, template, statistical, heuristic/rule, etc. (or a combination) based approach to extract relevant sentence(s) and exact answer(s) for a natural language question from the processed Wikipedia documents

3. **Task 3**: Provide an executable program that will accept input and produce output as specified below:

   - Input: File containing a list of natural language questions (one per line)
   - Output a JSON file with the following information:
     a. Input question
     b. Exact answer phrase(s)
     c. Supporting sentence(s) in Wikipedia document
     d. Supporting Wikipedia document name(s)

Use the following format for results returned as JSON string:

NOTE: please do remember that elements between square brackets (most outer brackets in the format below) should be stored inside a JSON array since JSON arrays maintain order.

```
[
  {
    "Question": "question 1 string",
    "answers”:{//answers to question 1 here like below
      "1": "first answer",
      "2": "second answer",
      …
    },
    "sentences”:{//supporting sentences containing answers to question
1 like below
      "1": "sentence containing first answer",
      "2": "sentence containing second answer",
      …
    },
    "documents”:{//supporting Wikipedia documents containing answers
to question 1 like below
      "1": "Wikipedia article name containing first answer",
      "2": "Wikipedia article name containing second answer",
      …
    }
  },
  {
    "Question": "question 2 string",
    "answers”:{
      //answers to question 2 here
    },
    "sentences”:{
      //supporting sentences containing answers to question 2 here
    },
    "documents”:{
      //supporting Wikipedia documents containing answers to question
2 here
    }
  },
  ...
  //repeat the same format as above for other questions
]
```

4. **Performance Evaluation:** The performance of the system will be tested on an unseen benchmark set of 20 questions.

D. Project Point Distribution

1. Max points available: 100 points
2. Division of points:
    a. Project implementation and demo: 90 points
        i. Task 1: 30 points
        ii. Task 2: 35 points
        iii. Task 3: 10 points
        iv. Evaluation Results: 20 points
    b. Project Report: 5 points