# CS 6320 – Natural Language Processing
# Fall 2021
# Dr. Mithun Balakrishna
# Course Project Description (Version 1.1)

## A. Project Steps and Deadlines:

- **Project Group Formation**:
    - Due by **Thursday, October 7th 2021, 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 "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 **Thursday, October 7th 2021** at **8:15pm** in the class room
        - 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 "Project-FinalSubmission-GROUPNAME.zip", where GROUPNAME is your project group's name.
        - Submit the document to the "Project Submission" assignment inside the "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 a Question Answering (QA) system using NLP features and techniques for the following Question Types:

1. WHAT questions:
    a. Examples:
        i. What act was repulsive to Romans?
        ii. What company did Ray own?
2. WHEN questions:
    a. Examples:
        i. When was the invasion of Gaul by Rome?
        ii. When did Apple go public?
3. WHO questions:
    a. Examples:
        i. Who founded Apple Inc.?
        ii. To whom was John married?

The data is extracted from **S**tanford **Qu**estion **A**nswering **D**ataset (SQuAD), a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or *span*, from the corresponding reading passage, or the question might be unanswerable.

**Data**:

**1)** A set of 30 articles from which your Question Answering system should provide an answer for the input natural language question

**2)** Sample data with the following format is provided for training, development, and testing:
[[*article id*, [(*question 1*, *answer 1*) , … , (*question n*, *answer n*)]],
[*article id*, [(*question 1*, *answer 1*) , … , (*question n*, *answer n*)]],
…,
[*article id*, [(*question 1*, *answer 1*) , … , (*question n*, *answer n*)]]]

**Question Answering system requirements**: You QA system is required to **return the sentence** containing the answer for the **input question** from the **given 30 articles**.

- Input: A file containing natural language questions one per line
- Output: supporting sentence which contains the answer for each question, and the article id which contains the supporting answer sentence

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 articles in the dataset 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) for a natural language question from the processed SQuAD dataset:

   - Run the above described deeper NLP on the dataset and extract NLP features
   - Run the above described deeper NLP on the natural language question and extract NLP features
   - Implement a NLP knowledge driven (i.e. template, statistical, heuristic/rule, or a combination) approach to extract the relevant answer sentence for a natural language question from the given 30 articles dataset

     - **Notes:**
       - You are **NOT** allowed to use a Machine Learning (either traditional or Neural Network) technique for extracting the relevant answer sentence
       - You are allowed to use word embeddings to perform word similarity matching or other tasks

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 CSV file with the following columns:
     a. Input question
     b. Supporting sentence containing the answer of that question
     c. Supporting article's id

Use the following format for results returned in a CSV file:

*Question_string_1, article_id_1, answer_sentence_1*
*Question_string_2, article_id_2, answer_sentence_2*
……
*Question_string_N, article_id_N, answer_sentence_N*

4. **Performance Evaluation:** The performance of the system will be tested on an unseen test question set. (TBD)

## 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: 35 points
        iii. Task 3: 10 points
        iv. Evaluation Results: 20 points
    c. Project Report: 8 points