AIT726: Natural Language Processing with Deep Learning

Term Project Guide

This is a **self-defined** **teamwork** project designed for the students who are required to prepare an NLP system that solves a well-defined NLP problem. The technologies are NOT limited to those studied during the class. Students are encouraged to apply other technologies beyond the course content.

Note that each student in a team will be evaluated by other members in the same team. Every student in one team may have different credit based on their own overall contributions and performances.

General Instructions

You are free to pick your own topic. You must have your topic approved by the instructor. Turn in a paragraph long description of your proposed topics via the Blackboard before you start working on your proposal.

Some project ideas are as follows:

- Extend your implementation of a solution to one of the class assignments. You must have some set of specific functionality extensions in mind and be ready to articulate them, why they are important, what they will add to your existing solution, etc.
- You can browse the web for publicly available NLP corpora, e.g., https://github.com/niderhoff/nlp-datasets, and identify a data set that would serve as a testbed for you. Example data sets in the above link include document classification corpora, spam detection corpora, sentiment analysis corpora, etc. Linguistic Data Consortium is also a great resource for NLP corpora.
- Any other NLP topic that you find interesting.

For your topic, you must know what a typical baseline solution looks like and how well it performs. You must also know the state of the art and its performance level. You must run and evaluate your system on your data, then present us your solution and its performance, along with a comparison to the baseline and the state of the art.

Your system must run on the SW/HW platform as described in your README file and final report. It must be clearly documented and implemented in such a way that the instructor and/or GTA can easily understand and run your system.

Objectives

The main objectives of designing this term project are to help students:

- prepare an NLP with deep learning system that solves a well-defined NLP problem using deep learning

- design NLP algorithms based on the requirements
- apply tools and methods of NLP and/or data analytics technologies
- develop skills in applying NLP techniques to particular tasks
- write scientific or technical documentations
- prepare a conference-style paper for submission
- conduct scientific technical presentations
- learn project management skills

Deliverables

The NLP Project is a course-long teamwork effort consisting of several separate deliverables. You are strongly encouraged to work ahead on these deliverables. You may submit them early, if you would like, but they will not be assessed until after their respective due dates.

This project **(100%)** consists of **seven (7)** deliverables:

- Deliverable 1 **(2%)** – **Project Topic**
- Deliverable 2 **(11%)** – **Project Checkpoint 1**
- Deliverable 3 **(11%)** – **Project Checkpoint 2**
- Deliverable 4 **(11%)** – **Project Checkpoint 3**
- Deliverable 5 **(20%)** – **Project Presentation** (15-20 minutes presentation and Q&A)
- Deliverable 6 **(20%)** – **Project Final Paper** (Conference-style**,** 8-10 pages)
- Deliverable 7 (**25%)** – **Project Working System**

Due dates: check the latest class schedule on blackboard.

Submission

Please submit all your files with the file names "**AIT726_Team#_NameInitials_*.docx/pptx/zip**" (e.g., * could be ***topic.docx***, ***cp1/2/3.docx***, ***PPT.pptx***, ***paper.docx***, ***sys.zip***). For example, **AIT590_Team3_NJ_KB_cp1.docx**.

Please submit to Blackboard under your team group.

Note that your submission may not be graded if you do not follow the required file naming convention.

Grading

The whole project will be graded based upon scope, complexity, quality, and deliverable.

Specific Requirements and Templates for Deliverables

Read the sections below for the specific requirements of each deliverable. **Use the following structures as the templates for your reports and submissions.**

1. Project Topic (*submitted in **one 1-page Word file***)

- **Header:** Course # & Name, Professor, University
- **Title, Team# and/or Name(s)**
- Free to pick your own topic but need to get approved by the instructor.
- Turn one or two paragraph description of your proposed topic before you start to work on your proposal.

2. Project Checkpoint 1 (*submitted in **one Word file***)

- **Header:** Course # & Name, Professor, University
- **Tile, Team#, and/or Name(s)**
- **Introduction**
  - Briefly explain the project and its scope
    - Describe the problem that you are trying to solve
    - Explain why it should be interesting (to the audience), why it is challenging (technically)
    - Clearly and concisely describe the major contribution of your proposed solutions.
- **Related Work**
  - Discuss some of the previous attempts (by others in the field) to the problem you are trying to solve.
    - It should highlight the best solutions in the field, why they work, where they fall short, and how these approaches relate to your proposed solution to the problem.
- **Objectives**
  - Briefly describe/list the objectives of the overall project you proposed
- **Selected Dataset**
  - Briefly describe the selected dataset
    - Have your dataset on hand and start any manual annotation that you may need to do?
- **Proposed Solution**
  - Describe in a paragraph some of the potential approaches you will explore for your solution to the problem.
  - How are your solutions different from what has already been done?
    - Innovation or
    - significant improvement
- **Proposed development platforms**
  - Briefly describe the software and hardware development platforms
- **References**
- *Other contents could be added if necessary*
  - *show a piece of data as an appendix*

3. Project Checkpoint 2 (*submitted in **one Word file***)

- **Header:** Course # & Name, Professor, University
- **Tile, Team#, and/or Name(s)**
- **Selected Dataset(s)**
  - ▪ Briefly describe the overall selected dataset
  - ▪ Briefly describe the features you would like to select for your solution
- **Baseline Solution**
  - Implement and evaluate a baseline solution to your problem on your data.
- **Extend Checkpoint 1 write up**
  - Describe your baseline solution and present its results.
  - Present an analysis and discussion of the errors of the baseline solution.
  - What errors should be addressed by your (next iteration of the) system?

4. Project Checkpoint 3 (*submitted in **one Word file***)

- **Header**: Course # & Name, Professor, University
- **Tile, Team #, and/or Name(s)**
- **Continue the work from Checkpoint 2**
  - Given the errors of the baseline solution, implement your proposed solutions and present preliminary results.
  - Analyze the preliminary results of your solution and comment on whether or not it addressed the shortcomings of the baseline solution.
  - If it did not, what modifications should you make to the solution in order to improve the results?
- **Extend your write-up for Checkpoint 2 with the preliminary results**
  - Analyze the results
  - Explain for how your proposed solution succeeded (or not) to address the previously identified errors.
  - What errors remain open and what should be done about those?

5. Project Presentation (*submitted in one **PPT file***)

- **Make Power Point slides**
  - Title, Team#, and/or Name(s)
  - Course # & Name, Professor, University
  - Introduction
  - Related Work
  - Dataset(s)
  - Framework
  - Algorithms/Approaches
  - Results and Analysis
  - Conclusions
- **In-Class Presentation**
  - ▪ Presentation of the final report
  - ▪ Demonstrations of the working system

6. Project Final Report – A Conference Paper

NOTE: The final report should build on your check point writeups that you have been submitting throughout the term. The length of the paper is **8-10 pages**, **two-column**, **single-spaced**, including references.

Please follow the attached **AIT726 Paper Format Template** to complete the following sections as required*:*

- **Header:** Course # & Name, Professor, University
- **Paper Title and Name(s)**
- **Abstract** (200-300 words)
    - The abstract should be a concise summary of the general thesis and conclusions of the paper.
    - Key words (5-8)
- **Introduction**
- **Related Work**
    - Discuss some of the previous attempts (by others in the field) to the problem you are trying to solve.
    - It should highlight the best solutions in the field, why they work, where they fall short, and how these approaches relate to your proposed solution to the problem.
    - Briefly describe what's **difference** (i.e. innovation or significant improvement) of your proposed solution(s) from other's work.
- **The Dataset(s)**
- **The Approaches/Methods**
    - **Framework**
    - Draw an overview diagram of the approach or method framework
    - Briefly describe each part
    - **Algorithms/Approaches**
    - Describe the major algorithm(s)/approaches.
        - Existing algorithm(s)
            - Briefly describe the existing algorithm(s) you used and why
            - Briefly describe how to use these algorithms for your data analysis
        - New **algorithm**(s) developed by your own
            - Explain/describe the algorithm(s) in detail
            - Describe how to use these algorithms for your data analysis
    - **SW/HW Development platforms**
- **Experimental Results and Analysis**
    - Explore and present analysis of the dataset using relevant approaches
    - Prepare relevant analysis and visualizations for selected data items
    - Analyze and interpret the results
    - Discuss errors and open questions
    - Explain for how your proposed solution succeeded (or not)
- **Conclusions**
    - Draw conclusions for overall project
    - Lessons learned from this project
    - Future work

- **Acknowledgements**
- **References**
  - Provide appropriate citations and references
  - Be sure to include a citation and link(s) for the dataset(s)
  - see http://infoguides.gmu.edu/citingdata
- *Other contents could be added if necessary.*
- ***Note**: if your final report is NOT a **conference-paper style** as required, it will **deduct 50 points**.*

7. Project Code - A Working System (100%) (*submitted in one zipped file)*

- **(25 points) Introductory Comments (README)**
  Write a **README.txt** file. It should
  - (15 pts) clearly introduce to problem, point by point outline of solution, and examples of actual program input and output;
  - (10 pts) concisely describe how to set up and run your system, dataset link(s), and any other info you need to tell others to re-compile and re-run your system.
- **(25 points) Detailed Comments in source code**
  - The comments should be clear and easy to follow and understand.
    - 25 – detailed comments throughout body of code, where authors are clearly identified
    - 11-24 – partial comments throughout code, or no identification of authors
    - 1-10 – limited comments or comments that are difficult to understand and follow
    - 0 – no comments, or completely incomprehensible comments
- **(50 points) Project Functionality**
  - The project should implement **all** required functionality, runs on **all** evaluation data and achieves a score **greater than** the task baseline.
    - 50 – greater than the task baseline
    - 40 – equal to the task baseline
    - 30 – less than the task baseline
    - 20 – NOT get score due to kind of system errors
    - 10 – ONLY partial functionality on all or a partial subset of evaluation data
    - 0 – NOT function
- **Submissions**
  - All the source code, datasets, **README**, and related files should be submitted in **one zipped file: AIT726_Team#_NameInitials_sys.zip**.
  - If the original dataset is too big, please provide the link(s) for downloading in the **README** file.
  - If the zipped file is too big to submit into Blackboard, please contact Instructor.
- NOTE: the system may be re-compiled and re-run on instructor's computer to double check for grading. **Please clearly list your code running requirements and steps in the README file.**