Assignment 2

NLP application mini project: information retrieval and question answering system

Word count/length limit for the report: report approx. 12 pages
Weighting: 30% of the total course marks.

The purpose of this assignment is to provide you with the opportunity to implement a real-world solution for an NLP problem.

This assignment will support you in achieving the following Course Learning Outcomes:

2. Use existing natural language processing tools to conduct basic natural language processing, such as text normalization, named entity extraction, or syntactic parsing.
3. Use machine learning tools to build solutions for natural language processing problems.
4. Decompose a real-world problem into subproblems in natural language processing and identify potential solutions.

Project choices

Project option 1: Question answering system (one person work).

Design a question answering system. The data subset will be a set of news articles. Questions should be in natural language and answers will be just snippets of text extracted from the article. The questions and answers are short, limited to one sentence. The user submits an article and asks a question. The question may contain different phrases than those used in the article. For given question, retrieve text snippets. Then use these snippets to produce the answer. If the system cannot find a high confidence answer (select the confidence parameter), just say that there is no answer. The system has a simple prompt loop to continue until instructed to stop.

Assessable Tasks:

1. Read and pre-process dataset.
    a. Download and read the dataset *xxxxx* and use the code template. The dataset contains a very large number of articles in many different formats. You may use a random sample for your testing >= 1000 articles. If you do, describe how you sample the data and how many articles are in the sample.
    b. Use necessary text pre-processing. The type of text pre-processing depends on method you are going to use in the assignment. Examples: Bag of Words (BOW), sentence phrases.
2. NER. Perform Named Entity Recognition (NER). This will help to answer questions where different entity names are used.
3. Article indexing. Develop an indexing method that would make is faster to answer queries. The index may be thematic (by topic) or by Named Entity (which article talks about which entity) or more than one index.
4. Text matching utility.
    a. Create text matching utility that retrieves articles based on a question. The utility should output the article number and the matching score with the question.
    b. Make a set of example questions to test your code. Questions required for this assignment are simple, e.g., “How many states are in Australia?”,
5. Test utility.
    a. Develop a test utility that accepts a set of test questions with answers and outputs a metric e.g. MRR or MAP. Choose the metric which better for your purpose.

b. Test your application with at least 10 test questions and show the results using your chosen metric. You can create some of the test questions from Kaggle/target_tables included in the dataset. You can also use Stanford SQuAD (https://rajpurkar.github.io/SQuAD-explorer/)

Reading:

- Jurafsky textbook ch. 14.4 Knowledge-based Question Answering
- Jurafsky textbook ch. 14.5 Using Language Models to do QA

Project option 2: Basic dialog system (for a team of two)

The dialog system will be an extension of Option 1 with the following additional features: 1) The user does not specify the article, the system must find relevant articles in given dataset, and answer the question from the top ranked article. 2) The user may ask a related question referring to previous questions and answers.

Assessable Tasks:

TBD

Project option 3: Advanced dialog system (for a team of three)

The dialog system will be an extension of Option 2 with the following additional features: 1) All responses are in natural language, 4) questions may relate to more than one article, 4) The system may decide to ask the user for clarification to improve the response relevance and quality.

Assessable Tasks:

TBD

Project report structure

(choose sections that are relevant to the project option)

Front page: project title, assignment and group members
Abstract:
Briefly summarize the objectives, methodologies, and key findings of the project.
1. Introduction:
  - Introduce the importance of question answering dialog systems in the context of news articles.
  - Provide an overview of the tasks assigned to each student.
2. Data Preprocessing (Student 1):
  - Describe the data preprocessing steps, including cleaning, tokenization, and data augmentation.
3. Model Selection and Training (Student 1):
  - Discuss the choice of the NLP model for question answering.
  - Outline the fine-tuning process on the prepared dataset.
  - Present the evaluation results on the testing set.
4. System Architecture (Student 2):
  - Detail the overall architecture of the question answering dialog system.
  - Explain how the trained model from Student 1 is integrated into the system.
5. User Interface (Student 2):
  - Showcase the design of the user interface for interacting with the system.
  - Highlight user input and output mechanisms.
6. Dialog Flow (Student 2):

- Explain the implementation of a natural dialog flow, including handling ambiguous queries.
- Describe how the system transitions between user inputs and responses.

7. Evaluation (Both Students):
   - Present the test cases used to evaluate the system's performance.
   - Provide metrics such as accuracy, precision, recall, and F1 score.
   - Include user feedback on system usability and effectiveness.
8. Discussion:
   - Discuss the challenges faced during the development process.
   - Compare the achieved results with initial expectations.
   - Reflect on the collaboration between Student 1 and Student 2.
9. Conclusion:
   - Summarize the key findings and contributions of the project.
   - Discuss potential areas for future improvements.
10. References:
    - Cite relevant literature, resources, and tools used during the project.
11. Appendices (if any):
    - Include any supplementary materials such as code snippets, additional charts, or user feedback forms.

Submission

There will be ONE submission of the code and ONE submission of Report for each group.

Code: Python Notebook or Python code one file: **<group_number>_assign2.ipynb (or .py)**

Report: PDF file named **<group_number>_assign2.pdf**

Do not include dataset. Do not zip files.

Late submission rules:

If you hand in your work late, your mark will be capped, based on the number of late days. A part of the late day is counted as full day.

– 1 day late – mark capped at 75%

– 2 days late – mark capped at 50%

– 3 days late – mark capped at 25%

– more than 3 days late – no marks available

Assessment criteria

30% of this assignment weighting is for the code, and 70% for the report. The code and the report are marked per rubric.

Academic Integrity Declaration

By submitting this assignment, I declare that this assessment item is my own work, except where acknowledged, and has not been submitted for academic credit elsewhere. I acknowledge that the assessor of this item may, for the purpose of assessing this item, reproduce this assessment item and

provide a copy to another member of the University; and/or communicate a copy of this assessment item to a plagiarism checking service (which may then retain a copy of the assessment item on its database for the purpose of future plagiarism checking). I certify that I have read and understood the University Rules in respect of Student Academic Misconduct.