Assignment 2

NLP application mini project: information retrieval and question answering system.

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.

Dataset

The data a set of news articles (**news_dataset.csv**) in random order. It contains 7 columns:
id article author date year month topic
All columns can be used as needed.

Project choices

Project option 1: Information Retrieval system (one person work).

A. The question should be a simple sentence in natural language referring to a fact from a single sentence of an article given by the user.
B. Answers will be just snippets of text extracted from the article. An example of question for article number 17574 would be: "Who is the vice chairman of Samsung", answer: "Jay Y. Lee".
C. The user submits the number of one of the articles from that dataset and asks a question. The question may contain different phrases than those used in the sentence of the article. For given question, find the most relevant sentence, retrieve text snippets from the sentence, select one or more most relevant snippets (or phrases) to answer the question and display the snippets as the answer.
D. If the system cannot find a high confidence answer (select the confidence parameter), just say that there is no answer.

Programming tasks should include:

1. **Read and pre-process** dataset.
    a. Download and read the dataset **news_dataset.csv** and use the code template provided for coding. To make the testing faster, you may use a random sample >= 100 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 etc.
2. **Coreference Resolution utility**. This will help to identify which entity a phrase in the sentence is referring to. For example: "The de facto leader, **Jay Y. Lee**, the **vice chairman of Samsung**"
3. **Text matching utility**. Find most relevant sentence and its confidence score in the article based on the user question.
4. **Test utility**.

   a. Develop a test utility that accepts an article, a set of test questions with answers and outputs a metric e.g. F1, MRR or MAP. Choose the metric which is better for your purpose. (Jurafsky textbook ch. 14.7)
   b. Test your application with at least 10 test questions and show the results using your chosen metric. You can use Stanford SQuAD for your testing. (https://rajpurkar.github.io/SQuAD-explorer/)
5. **Other** utilities as required.

Suggested reading:

- Jurafsky textbook ch. 14 Question Answering and Information Retrieval

Project option 2: Information Retrieval and Question Answering system (for a team of two)

The system will be an extension of Option 1 with the following additional features:

A. The user does not specify the article, the system must find relevant articles in given dataset, and answer the question from one of the top ranked articles.
B. More than one sentence from the article may be required to answer the question, but the answer is still just snippets (or phrases) of text.
C. The system prompts the user to ask next questions in the loop, until the user enters "quit" command.

Programming tasks should include:

6. **Points from Option 1**, designed for Option 2 as needed. For example, Text matching utility should now retrieve articles based on a question. The utility should output the article number and the score of matching with the question. And it can now use Article Indexing and NER.
7. **Named Entity Recognition**. This utility will help to find entity given a question, and to build the article index. NER may also help to answer the question more precisely.
8. **Article indexing**. Develop an indexing method that would make is faster to answer queries. The index may be thematic (e.g. by topic, high level topics are given in the dataset) or by Named Entity (which article talks about which entity).
9. **Other** utilities as required.

Suggested reading:

- Jurafsky textbook ch. 14 Question Answering and Information Retrieval

Project option 3: Question Answering and Dialog system (for a team of three)

The dialog system will be an extension of Option 2 with the following additional features:

A. All system responses are in natural language.
B. Questions may need more than one article to answer.
C. The user may ask a related question referring to **previous** questions and/or answers. Related questions are also in natural language. You can specify a limit on how many related questions can be asked and/or what context they can refer to.

Programming tasks should include:

10. **Points from Option 2**, designed for Option 3 as needed.
11. **Entity linking** utility. This utility should be able to recognise the same entity used in different articles.
12. **Implementation of response in natural language.** This utility should take the original question, article(s) that contain the answer and output an answer in natural language.
13. Utility that decides if user's question is a new question or continuation of previously asked questions.
14. **Other** utilities as required.

Suggested reading:

- Jurafsky textbook ch. 15.4, 15.5 and 15.6

Project report structure

Front page: project title and group members

Abstract:

Briefly summarize the objectives, methodologies, key findings and limitations of the project.

1. Introduction:

- Introduce the importance of question answering dialog systems in the context of news articles. That is, what kind of business your system is suited to and what kind of users.
- Identify limitations of your system: e.g. what kind of questions it can and/or cannot answer. Whether the question needs to be in a specified format.
- Provide an overview of the tasks to be completed.

2. Preprocessing:

- Describe the data preprocessing steps, such as cleaning, tokenization, and data augmentation and anything else as needed.

3. System Architecture:

- Detail the overall architecture of the system.
- Explain how the machine learning models are integrated into the system.

4. Model Selection and Training:

- Discuss the choice of the NLP or ML models (as relevant to your project Option).
- Specify evaluation metrics and why they are selected.
- Outline the fine-tuning process on the validation (aka development) dataset.
- Present and discuss the evaluation results of the models that you have tested. Describe how did you choose models, supported by evaluation results.

5. User interaction with the system:

- Explain the how the user can interact with the system as relevant to your project Option.
- Give an example of user session (user and system interaction from the first user question until the "quit" (as relevant to your project Option)

6. System evaluation:

- Present test cases that you have used to evaluate the system's performance.
- Show and discuss the test results from running these tests.

7. Conclusion:

- Summarize the key findings and contributions of the project.
- Discuss the challenges faced during the development process.
- Discuss potential areas for future improvements.

8. References:

- Cite relevant literature, resources, and tools used during the project.

9. Appendices (if any):

- Include any supplementary materials such as additional experimental results, additional charts etc.

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.