Natural Language Processing (CS60075)

Assignment 1: Text pre-processing and matching using spaCy

Deadline: 4th Feb 23:59

Guidelines:

- ➔ This assignment is to be done individually by each student. You should not copy any code from one another, or from any web source. We will use standard plagiarism detection tools on the submissions. Plagiarized codes will be penalized heavily.
- ➔ This assignment is meant to use Python and **spaCy** which is a library for performing several NLP related tasks easily.
- ➔ Solutions should be uploaded via the CSE Moodle website (see course website for details). Submit one .zip file containing a folder with your codes and the pdf report. Name the compressed file the same as your roll number. Example: "*24CS60R00.zip*"
- ➔ Try writing code that is organized into different functions and add explanatory comments. Make sure your code can be easily read. Codes that cannot be understood will be penalized.

Problem:

You are tasked with making a retrieval system for Quora. Given a new question (query), you want to find out the most similar questions that already exist (documents) in the Quora database, so that they can be shown as recommendations.

Dataset format:

- ➔ The dataset 'Query_Doc' contains 3 CSV files. You can download the dataset from this link.
- ➔ The *'query'* file contains 100 query ids and query texts.
- ➔ The *'docs'* file contains 10,000 doc ids and doc texts.
- ➔ The *'qdrel'* file contains the query ids and the doc ids, if they are similar. 130 such relations are present in the file. Use this for evaluating the methods.

Task 1: *(40 marks)*

1. Preprocess the docs and queries – remove characters other than alphanumeric or whitespaces.
2. Correct spelling in the queries and documents using SpaCy. Only for each query with some correction, print the original and corrected query in separate lines, followed by **two** newlines (\n).
3. Tokenize the words in the documents using spacy. Remove all words that occur in less than 5 documents or more than 85% of the documents (why?) – this forms your vocabulary for the task. For each document and query create TF-IDF vectors. (you may or may not use Sklearn library for this part).
4. For each query, find the **cosine similarity** of its vector with that of the documents. Use this to find the top 5 and top 10 most similar documents.
5. Calculate the Precision@k scores: report **P@1, P@5** and **P@10** averaged over all queries.

Task 2: *(20 marks)*

1. Improve the performance of Task1 by **stemming** the tokens (using spacy) before calculating the vocabulary.
2. Improve the performance of Task1 by **lemmatizing** the tokens (using spacy) before calculating the vocabulary.
3. Report the size of the vocabulary you obtained as part of Task 1, the vocabulary size after stemming and the vocabulary size after lemmatization.
4. Report the performance metrics in both these cases and discuss the results (why or why not performance has increased).

Task 3: *(30 marks)*

1. Improve the model from Task 2.2 further with Named Entity Recognition (NER) and Parts-Of-Speech (POS) tagging using spaCy.
2. For each query and document vector, give more weightage to some important words. In essence, for each of the tf-idf vectors, multiply **2** along the dimensions which contain **nouns**, and multiply **4** for the **named entities**.
3. Report the performance metrics considering these additional weights. Discuss the results (why or why not performance has increased).

Task 4: *(10 marks)*

Try to improve efficiency and/or performance of the matching model. In the report explain the changes you've made and the reasons behind them. Marks will be given based on the creativity of the solution.

To be submitted:

1. **Codes: Submit a main.ipynb notebook or a main.py file from where execution will be started. You are free to create other files as per convenience for legibility.**
2. **The dataset folder "Query_Doc"**
3. **A report (as a doc or pdf). Directions for preparing the Report:**
    - Short description of the work
    - Performance of all the models in terms of average P@1, P@5 and P@10 scores.
    - Highlight any advanced modifications that you've done in your report.
