FIT5196-S1-2019 assessment 1

This is an individual assessment and worth 35% of your total mark for FIT5196.

Due date: 11:59 PM, Sunday, April 14, 2019

Text documents, such as crawled web data, are usually comprised of topically coherent text data, which within each topically coherent data, one would expect that the word usage demonstrates more consistent lexical distributions than that across data-set. A linear partition of texts into topic segments can be used for text analysis tasks, such as passage retrieval in IR (information retrieval), document summarization, recommender systems, and learning-to-rank methods.

Task 1: Parsing Raw Text Files (%55)

This assessment touches the very first step of analyzing textual data, i.e., extracting data from semi-structured text files. Each student is provided with a data-set that contains information about several units in the Monash University (please find your own file from this link, i.e., **<your_student_number>.txt**). Each data-set contains information about the unit, e.g., unit code, unit title, synopsis, requirements, output, chief examiner, etc. (**see test.txt**). Your task is to extract the data and transform the data into the **XML** and **JSON** format with the following elements:

1. Unit code: is a 7-character string (three uppercase letter followed by 4 digits).
2. Pre-requisites: **only the unit codes** of the units that are pre-requisite + co-requisite for the current unit ('NA' if the value is Null).
3. Prohibitions: only the unit codes of the units that are prohibited to be taken with the current unit ('NA' if the value is Null).
4. Synopsis: a string containing the synopsis of the unit ('NA' if the value is Null)
5. Requirements: the list of requirements of the current unit ('NA' if the value is Null).
6. Outputs: the list of outputs of the current unit ('NA' if the value is Null).
7. Chief-examiners: the list of the chief-examiners of the current unit ('TBA' if Null).

The output and the documentation will be marked separated in this task, and each carries its own mark.

Output (50%)

See test.xml and test.json for detailed information about the output structure. Note that your output must exactly follow the sample outputs' structure (i.e. test.xml and test.json files) and any deviation from this structure (e.g. wrong key names which can be caused by different spelling, different upper/lower case, …. or wrong hierarchy which can be caused by the wrong usage of '[' in the json files) will result to zero marks for the output. So please be careful.

Please note that for this task, the **re** and the **json packages** in **Python** are the only packages that you are allowed to use and the following must be performed to complete the assessment.

- Designing efficient regular expressions in order to extract the data from your dataset **<your_student_number>.txt**

- Storing and submitting the extracted data into an XML file, **<your_student_number>.xml** following the format of **test.xml**
- Storing and submitting the extracted data into a JSON file **<your_student_number>.json** following the format of **example.json**
- Explaining your code and your methodology in **task1_<your_student_number>.ipynb**

Methodology (25%)

The report should demonstrate the methodology (including all steps) to achieve the correct results.

Documentation (25%)

The solution to get the output must be explained in a well-formatted report (with appropriate sections and subsections). Please remember that the report must explain both the obtained results and the approach to produce those results. You need to explain both the designed regular expression and the approach that you have taken in order to design such an expression.

Task 2: Text Pre-Processing (%45)

This assessment touches on the next step of analyzing textual data, i.e., converting the extracted data into a proper format. In this assessment, you are required to write Python code to preprocess a set of unit information and convert them into numerical representations (which are suitable for input into recommender-systems/ information-retrieval algorithms).

The data-set that we provide contains 400 unit information crawled from Monash University. Please find your pdf file from this link.The pdf file contains a table in which each row contains information about a unit which is unit code, synopsis, and requirements. Your task is to extract and transform the information for each unit into a vector space model considering the following rules:

Generating sparse representations for the unit information

In this task, you need to build sparse representations for the unit information, which includes word tokenization, vocabulary generation, and the generation of sparse representations.

Please note that the following tasks must be performed (**not necessarily in the same order**) to complete the assessment.

1. The word tokenization must use the following regular expression, **"\w+(?:[-']\w+)?"**
2. The context-independent and context-dependent (with the threshold set to %95) stop words must be removed from the vocab. The stop words list (i.e, **stopwords_en.txt**) provided in the zip file must be used.
3. Tokens should be stemmed using the Porter stemmer.
4. Rare tokens (with the threshold set to %5) must be removed from the vocab.
5. Tokens must be normalized to lowercase except the capital tokens appeared in the middle of a sentence/line.

6. Tokens with the length less than 3 should be removed from the vocab.
7. First 200 meaningful bigrams (i.e., collocations) must be included in the vocab using **PMI** measure.

The output and the documentation will be marked separated in this task, and each carries its own mark.

Output (50%)

The output of this task must contain the following files:

8. **task2_<your_student_number>.ipynb** which contains your report explaining the code and the methodology
9. **<student_number>_vocab.txt**: It contains the **bigrams and unigrams** tokens in the following format, **token_string:integer_index.** Words in the vocabulary must be sorted in alphabetical order.
10. **student_number>_countVec.txt:** Each line in the txt file contains the sparse representations of one of the units (i.e. a table row in the pdf file) in the following format **unit_code, token1_index:wordcount, token2_index:wordcount,...**

Methodology (25%)

The report should demonstrate the methodology (including all steps) to achieve the correct results.

Documentation (25%)

The solution to get the output must be explained in a well-formatted report (with appropriate sections and subsections). Please remember that the report must explain both the obtained results and the approach to produce those results.

Note: all submissions will be put through a plagiarism detection software which automatically checks for their similarity with respect to other submissions. Any plagiarism found will trigger the Faculty's relevant procedures and may result in severe penalties, up to and including exclusion from the university.