FIT5196-S2-2018 assessment 1

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

Due date: 11:55pm, 2 September 2018

Text documents, such as resumes and job postings, 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 (%40)

This assessment touches the very first step of analyzing textual data, i.e., extracting data from unstructured text files. Each student is provided with a data-set that contains several job postings (please find your own file from **task1.rar**, i.e., **<your_student_number>.dat**). Each data-set contains information about the job advertisements, e.g., job title, job description, start date, required qualifications (**see sample.pdf and sample.txt for the data dictionary**). Your task is to extract the data and transform the data to the **XML** and **JSON** format. Please note that the **re** and **json packages** in **Python** are the only packages that you are allowed to use in this task and the following must be performed to complete the assessment.

- Designing an efficient regex in order to extract the data from the file
- Storing and submitting the extracted data into an XML file, **<your_student_number>.xml** following the format of **example.xml**
- Storing and submitting the extracted data into a JSON file **<your_student_number>.json** following the format of **example.json**
- Submitting **task1_<your_student_number>.ipynb**

Task 2: Text Pre-Processing (%30)

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 resumes and convert them into numerical representations (which are suitable for input into recommender-systems/ information-retrieval algorithms).

The data-set that we provide contains 250 CVs for each student. Please find the **resume_dataset.txt** to know the PDF files in your own data-set. Each line in the csv file contains the id of the resumes that a student needs to include in the data-set (**for example 1111111111: [3 34 5 …] means that the student 1111111111 data-set includes resume_(3), resume_(34), resume_(5),...**). CVs contain information about the applicants represented in the PDF format.

The information includes, for example, personal information, skills, work experience, education, etc. Your task is to extract and transform the information for each applicant.

Generating sparse representations for the resumes

The aim of this task is to build sparse representations for the resumes, 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.

- **Pdfminer** package must be used to convert PDFs to txt files
- The word tokenization must use the following regular expression, **"\w+(?:[-']\w+)?"**
- The context-independent and context-dependent (with the threshold set to %98) 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.
- Tokens should be stemmed using the Porter stemmer.
- Rare tokens (with the threshold set to %2) must be removed from the vocab.
- Tokens must be normalized to lowercase except the capital tokens appeared in the middle of a sentence/line.
- Tokens with the length less than 3 should be removed from the vocab.
- First 200 meaningful bigrams (i.e., collocations) must be included in the vocab.
- The output of this task must contain the following files:
    - **task2_<your_student_number>.ipynb**
    - **<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.
    - **student_number>_countVec.txt:** the txt file contains all the “selected” resumes in the data-set. Each line in the txt file contains the sparse representations of one of the resumes in the data-set in the following format **file_name, token_index:count, token_index:count,...**

Documentation (%30)

Both of the above tasks 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. For example, in task 1 you need to explain both the designed regular expression and the approach that you have taken in order to design such an expression. Please take a look at the attached **example.ipynb** for an example of a decent report.

Bonus: Ranking resumes w.r.t the job advertisements (%10):

In this task, you move outside data wrangling and enter to the data analysis realm. The purpose of this task is to demonstrate how wrangled data from different sources can be used to help the organizations to make informative decisions. This task will be marked with a binary scheme, which means that you'll be rewarded with the full mark **if and only if** you do a decent study otherwise you'll get zero. So, you should only try this task if you are %100 sure that you have completed the first two tasks.

In this task, you are required to recommend the top 10 resumes that you think are the best fit for the first 500 job advertisements in task 1 w.r.t their **"required qualifications"** section. No more specifications are required as this is an open problem and you are allowed to formulate this problem as you wish. However, IR and RS methods can be a good place to start!

Output files: you need to generate two files i.e., **bonus_<your_student_number>.ipynb** and **bonus_<your_student_number>.txt** which contains the recommended resumes for the first 500 job advertisements in your task 1 data-set. The txt file must contain 500 lines and each line of the txt file must follow the following format: **Job_advertisment_id: first_ranked_resume_id, second_ranked_resume_id, …., tenth_ranked_resume_id**

Note 1: 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.

Note 2: sample files are just to demonstrate the format and structure of the files and their content should not be used to assess your output/methodology.

Note 3: the maximum possible marks of the assignment is 100 (including bonus).