FIT5196-S2-2019 assessment 2

This is a group assessment and worth 25% of your total mark for FIT5196.

Due date: 11:55 PM, Sunday, September 15, 2019

Text Pre-Processing & Feature Generation

This assessment touches on the next step of analyzing textual data, i.e., extracting data from non-structured format and converting the extracted data into a proper format suitable for a downstream modelling task. In this assessment, you are required to write Python code to preprocess a set of published papers and convert them into numerical representations (which are suitable for input into NLP AI systems, recommender-systems, information-retrieval algorithms, etc.)

Students Dataset

Each group is provided with a data-set containing 200 URLs for papers published in a popular AI conference. Please find your pdf file from this link.The pdf file (**<GroupName>.pdf**) contains a table in which each row contains a paper unique id and a URL where it can be downloaded.

Assessment Requirements

Each group is required to complete the following two tasks:

1. Generate a sparse representation for **Paper Bodies** (i.e. paper text without Title, Authors, Abstract and References). The sparse representation consists of two files:
    a. Vocabulary index file
    b. Sparse count vectors file

2. Generate a CSV file (stats.csv) containing three columns:
    a. Top 10 most frequent terms appearing in all **Titles**
    b. Top 10 most frequent **Authors**
    c. Top 10 most frequent terms appearing in all **Abstracts**

    Note: In case of ties in any of the above fields, settle the tie based on alphabetical ascending order. (example: if the author named John appeared as many times as Mark, then John shall be selected over Mark)

Solution Guidelines

To be able to complete the above tasks, please follow the below guidelines:

1. Use the given URLs to programmatically download the PDF files (**requests** package can be used - manual download will be penalised, so please only use it as a last resort)
2. Read the PDF files into text and extract the required entities to complete the above tasks. (**pdfminer** and **re** packages must be used for this task)

Sparse Feature Generation

Before building the sparse representation, you will need to perform text preprocessing on **Paper Bodies**. Please follow the below text preprocessing steps (**not necessarily in the same order**, you will need to figure out the correct order of operations that produces the correct set of vocabulary)

A. The word tokenization must use the following regular expression, r**"[A-Za-z]\w+(?:[-'?]\w+)?"**
B. The context-independent and context-dependent (with the threshold set to %95) stop words must be removed from the vocab. The context-independent stop words list (i.e, **stopwords_en.txt**) provided in the zip file must be used.
C. Unigram tokens should be stemmed using the Porter stemmer. (be careful that stemming performs lower casing by default)
D. Rare tokens (with the threshold set to 3%) must be removed from the vocab.
E. Tokens must be normalized to lowercase except the capital tokens appearing in the middle of a sentence/line. (use sentence segmentation to achieve this)
F. Tokens with the length less than 3 should be removed from the vocab.
G. First 200 meaningful bigrams (i.e., collocations), based on highest total frequency in the corpus, must be extracted and included in your tokenization process. Bigrams should not include context-independent stopwords as part of them and they should be separated using double underscore i.e. "__" (example: "artifical__intelligence")

Statistics Generation

To complete the second task you will need to perform the following preprocessing steps on the **Titles** and **Abstracts** before extracting the required stats:

A. The word tokenization must use the following regular expression, r**"[A-Za-z]\w+(?:[-'?]\w+)?"**
B. The context-independent stop words (i.e, **stopwords_en.txt**) must be removed
C. For **Abstracts**, Tokens must be normalized to lowercase except the capital tokens appearing in the middle of a sentence/line. (use sentence segmentation to achieve this). For **Titles**, tokens must be all normalised to lowercase.

The output and the documentation will be marked separately in this assessment.

Output (60%)

The output of this task must contain the following files - please refer to sample outputs in case of any doubts:

1. **<GroupName>_ass2.ipynb** which contains your report explaining the code and the methodology. (example: Group001_ass2.ipynb)
2. **<GroupName>_vocab.txt**: It contains the **bigrams and unigrams** tokens in the following format, **token_string:token_index.** Words in the vocabulary must be sorted in alphabetical ascending order.
3. **<GroupName>_count_vectors.txt:** Each line in the txt file contains the sparse representations of one of the papers in the following format
   paper_id, token1_index:token1_wordcount, token2_index:token2_wordcount, etc.
   Note that tokens with zero wordcount should NOT be included in the sparse representation.
4. **<GroupName>_stats.csv:** A dataframe with three columns containing the required statistics: **top10_terms_in_abstracts**, **top10_terms_in_titles, top10_authors**
5. **assignment-cover-<GroupName>.pdf:** Documenting contribution percentage of each member of the group using the provided assignment-cover-group template. The submitted file must be signed by each member of the group.
6. **reflective-diary-<GroupName>.pdf:** Documenting the details of the individual contributions of each member of the group using the provided reflective-diary template.

Notes:

- Please zip above files into a single file with the name **<GroupName>_ass2.zip**
- **<GroupName>** should be replaced with the group name (i.e. the same name used in the students dataset. Ex: Group001)
- Sample outputs provided are only to illustrate the required structure of output files

Methodology (25%)

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

Documentation (15%)

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.