MONASH

Chao-Kai Hsu

FIT5196 Data wrangling S2 2017

Dashboard ▶ Faculty of Information Technology ▶ Active Categories ▶ 2017 ▶ FIT S2 2017 ▶ FIT5196-S2-2017 ▶ My Assessments ▶ Assessment 1: Text Preprocessing

Assessment 1: Text Preprocessing

Please carefully review all the requirements below to ensure you have a good understanding of what is required for your assessment.

1. Due Date

This specific due date and time can be **viewed below in the Grading Summary**.

2. Assessment description

Statistical relational learning studies methods for the statistical analysis of relational, or graph-structured data. It has gained a wide range of applications in, for example, social network analysis, recommender systems, knowledge graph completion, and bioinformatics. For example, recommending movies to users of Netflix could use a relational model built on top of users' watching histories. To improve the accuracy of the recommendation, one can also build information such as user's' profiles, movie's categories, and movie's description into the learning process. This assessment touches on the first step of analyzing relational data, i.e., extracting data from semi-structured text files and convert the data into a proper format.

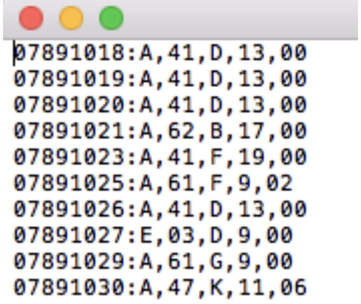
The data that we provide contains 2,500 patents. Each patent contains a large amount of information represented in the notoriously verbose XML format. The information includes, for example, publication reference, application reference, abstract, description, claims, citations, etc. Here, assume that we are going to focus on the analysis of patent citation network, and the analysis will take into account both the network itself and information associated with each patent (i.e., International Patent Classification (IPC) code and abstract). Therefore, your task is to extract the citation network, hierarchical IPC code, and abstract for each patent.

This is an individual assignment and worth 30% of your total mark for FIT5196.

Details of tasks:

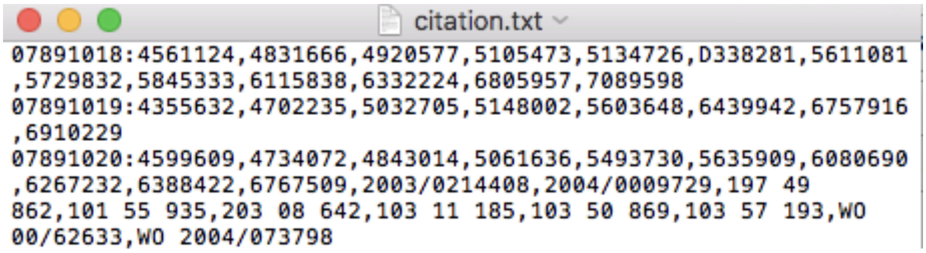
- **Extract the hierarchical IPC code.** World Intellectual Property Organization has a hierarchical classification scheme that contains Section, Class, Subclass, Main Group, and Subgroup. Please refer to the scheme web page for more details. Your task here is to extract the hierarchical IPC codes for all the patents, and store them in a file, called "classification.txt", in the following format:

patent’s_ID:Section,Class,Subclass,Main_group,Subgroup.
Your output file should look like



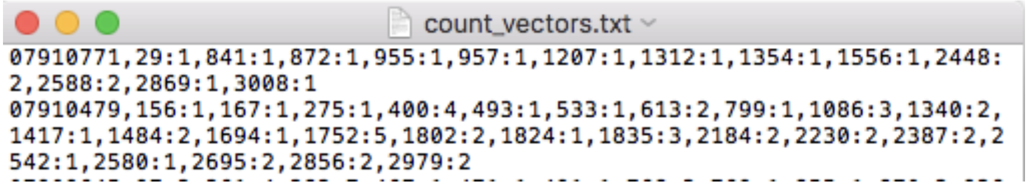
```
07891018:A,41,D,13,00
07891019:A,41,D,13,00
07891020:A,41,D,13,00
07891021:A,62,B,17,00
07891023:A,41,F,19,00
07891025:A,61,F,9,02
07891026:A,41,D,13,00
07891027:E,03,D,9,00
07891029:A,61,G,9,00
07891030:A,47,K,11,06
```

- **Extract the citation network.** Each patent cites a number of existing granted patents. Here, you are required to extract all the references for each patent, and store them in a file, called “citations.txt”, in the following format: ***citing_patent_id:cited_patent_id,cited_patent_id,....***
Your output file should look like



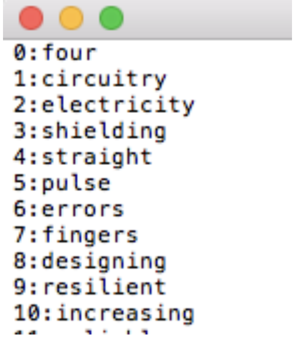
```
07891018:4561124,4831666,4920577,5105473,5134726,D338281,5611081
,5729832,5845333,6115838,6332224,6805957,7089598
07891019:4355632,4702235,5032705,5148002,5603648,6439942,6757916
,6910229
07891020:4599609,4734072,4843014,5061636,5493730,5635909,6080690
,6267232,6388422,6767509,2003/0214408,2004/0009729,197 49
862,101 55 935,203 08 642,103 11 185,103 50 869,103 57 193,WO
00/62633,WO 2004/073798
```

- **Identify number of patent citation:** Write the necessary code to identify number of times a particular patent has been cited. The output file "cited.txt" should be in the following format: ***cited_patent_id: <number of times it is cited>***.
- **Extract and preprocess abstracts.** Abstract provides important information that can assist in learning the citation network. In this task, you are required to extract all the abstracts for all the patents, and then process and store those abstracts as sparse count vectors. Your output file, called “count_vectors.txt”, should look like



```
07910771,29:1,841:1,872:1,955:1,957:1,1207:1,1312:1,1354:1,1556:1,2448:
2,2588:2,2869:1,3008:1
07910479,156:1,167:1,275:1,400:4,493:1,533:1,613:2,799:1,1086:3,1340:2,
1417:1,1484:2,1694:1,1752:5,1802:2,1824:1,1835:3,2184:2,2230:2,2387:2,2
542:1,2580:1,2695:2,2856:2,2979:2
```

where each row corresponds to a patent’s abstract, starting with patent_id, followed by “word_index:count” pairs. “word_index” is the index of a word in the vocabulary file that you must also generate and name it “vocab.txt”. It should look like



```
0:four
1:circuitry
2:electricity
3:shielding
4:straight
5:pulse
6:errors
7:fingers
8:designing
9:resilient
10:increasing
```

For example, “29:1” means the 30th word in the vocabulary appears once in the abstract of patent “07910771”. In order to finish this task, **you must**

  - Tokenize the abstracts with both unigram and meaningful bigram collocations. For example, given a sentence like “data wrangling is one compulsory unit in the data science course”, and if both “data wrangling” and “data science” are identified as meaningful bigram collocations, you are supposed to derive the following list of tokens: “data_wrangling”, “is”, “one”, “compulsory”, “unit”, “in”, “the”, “data_science” and “course”. **You must include at least 100 meaningful bigrams in the tokenization** (e.g., “is a” is not meaningful, whereas “wind turbine” is meaningful).
  - Before generating the count vectors, remove all the stopwords (Hint: use the NLTK built-in stopword list), top-20 most frequent words based on word’s document frequency, and words only appearing in one

abstract.

3. Assessment Resources

Before you start writing your code, you will need to download the following data file:

- patents.xml: contains 2,500 patents stored in XML format.
- notebook template for this assessment. While using Jupyter Notebook, you should consider the use of different cells to make notes as you go. However, be precise and concise, please do not include things that are not relevant to the tasks in your notebook.

You must ensure that you can clearly explain what your code does with clear comments.You may need to review the FIT citation style tutorial to make you're familiar with appropriate citing and referencing for this assessment. Also, review the demystifying citing and referencing for help.

4. Assessment Criteria

The following outlines the criteria which you will be assessed against.

4.1 Grading Rubric

Specific grading details for this assessment are:

- The submitted code should work without any errors and should give the correct results.
- The code should be well structured and properly commented.
- The methods used in text preprocessing should be well justified in the Jupyter notebook.
- The notebook should be structured in a logical way so that it clearly shows how students finish the tasks in the assessment.
- The outcome of the interview, where students should demonstrate that the submitted assignments are their own work by be able to communicate their processes, justify their approaches, and answer some questions.

4.2 Penalties

- Late submission: for all assessment items handed in after the official due date, and without an approved extension, a 5% penalty applies to the student's mark for each day after the due date (including weekends, and public holidays) for up to 10 days. **Assessment items handed in after 10 days will not be considered.**

5. How to Submit

Once you have completed your work, take the following steps to submit your work.

1. **Save** all of your files as a **ZIP file, named as "surname_studentID_ass1.zip",** which includes:
    1. An **IPython Notebook file**, whose extension must be ".ipynb". The file name must be in the following format: **surname_studentID_ass1.ipynb**
    2. All the output files including
        - classification.txt
        - citations.txt
        - count_vectors.txt
        - vocab.txt

2. Click the **Add Submission** button below to submit and upload your assignment. **Please do remember to accept the submission statement! Only the submitted assignments will be marked. Those shown as a draft won't be marked!**

If you need further guidance on how to submit an assessment item, please review the **Submitting an assessment** overview. If you need further assistance, please go to **Help & Support**.

Submission status

Attempt number	This is attempt 1.
Submission status	No attempt
Grading status	Not graded
Due date	Wednesday, 30 August 2017, 12:55 AM
Time remaining	15 days 13 hours
Last modified	-
Submission comments	▸ Comments (0)

Add submission

Make changes to your submission

NAVIGATION

Dashboard

- Site home
- Site pages
- Current unit
  - FIT5196-S2-2017
    - Participants
    - Unit Updates
    - Unit Information
    - Consultation Times
    - Forums
    - My Assessments
    - Academic Integrity Tutorials
    - Student Academic Integrity Policy
    - **Assessment 1: Text Preprocessing**
    - Assessment_1_patent_data
    - How to submit an assignment in Moodle
    - Notebook template for assessment 1

- Week 1 (24 July - 30 July) Introduction to Data Wr...
- Week 2 (31 July - 6 Aug) Introduction to Regular E...
- Week 3 (7 Aug - 13 Aug) Text Data Preprocessing
- Week 4 (14 Aug - 20 Aug) Text Data Preprocessing
- Week 5 (21 Aug - 27 Aug) Handling Raw Data in Diff...

My units

Jump to Content

My.Monash

IT Academic Integrity

IT Student Portal

ADMINISTRATION

Unit administration