12/10/2020
CS 410 - Text Information System
Difan Gu, Yanyue Wang, Wen Long

Team APlus Final Documentation

Overview:

Oftentimes, users found it's hard to fathom the information that they barely learnt before. For example, how are we able to answer the question, "what is data science?", especially to an outsider? How are we able to clarify the definition of "data science" or "computer science" by borrowing more basic or common words or terms to further help our users to understand? Our implementation of a semantic annotation algorithm based on the paper, "Generating semantic annotations for frequent patterns with context analysis", can achieve the goal. The final goal is to automatically decipher certain words, terms, and even sentences by providing its highly-associated while distinct frequent patterns in semantic text form.

In our case, to be more specific, we used the algorithm to summarize what specialty each college published in major computer science conferences. The Digital Bibliography & Library Project (DBLP) computer science bibliography acts as good study material for our project. It contains the metadata of more than 1.8 million publications in thousands of journals and conferences proceeding series written by over 1 million authors. It first started to be a bibliography on database systems and logic programming but has since expanded to all fields of computer science.

From this well-structured dataset, we selected three top U.S.-based universities including Massachusetts Institute of Technology (MIT), Georgia Institute of Technology (GT), and the University of Maryland (UMD) as our use case. By implementing the algorithm, we can extract a series of words or terms to differentiate their academic focus based on thousands of paper titles published throughout the years: some colleges will be more inclined to data analysis, the others will more concentrate on wireless systems. In the real world, the utilization of automatic annotation can also be universal: users can use the algorithm to understand not-well-defined text information such as "NLP", "Machine Learning" and "Deep Learning" that is not defined in the dictionary such as Merriam Webster.

Step 1: Load, clean raw data, and tokenization
As can be seen in Figure 1, the software first converts the XML file downloaded from DBLP into a string format and output a table. Every publication was saved as a single string. Then all the elements inside the XML schema were saved as an item inside a dictionary, with the key being the name of the element and the value being the content. Another list has been created for every single publication as an element, describing the affiliated university as well as the publication title. We then ranked these universities by their number of publications and printed the top 50. Every title of the publication was then tokenized word by word. Each token was assigned an integer for further mining.

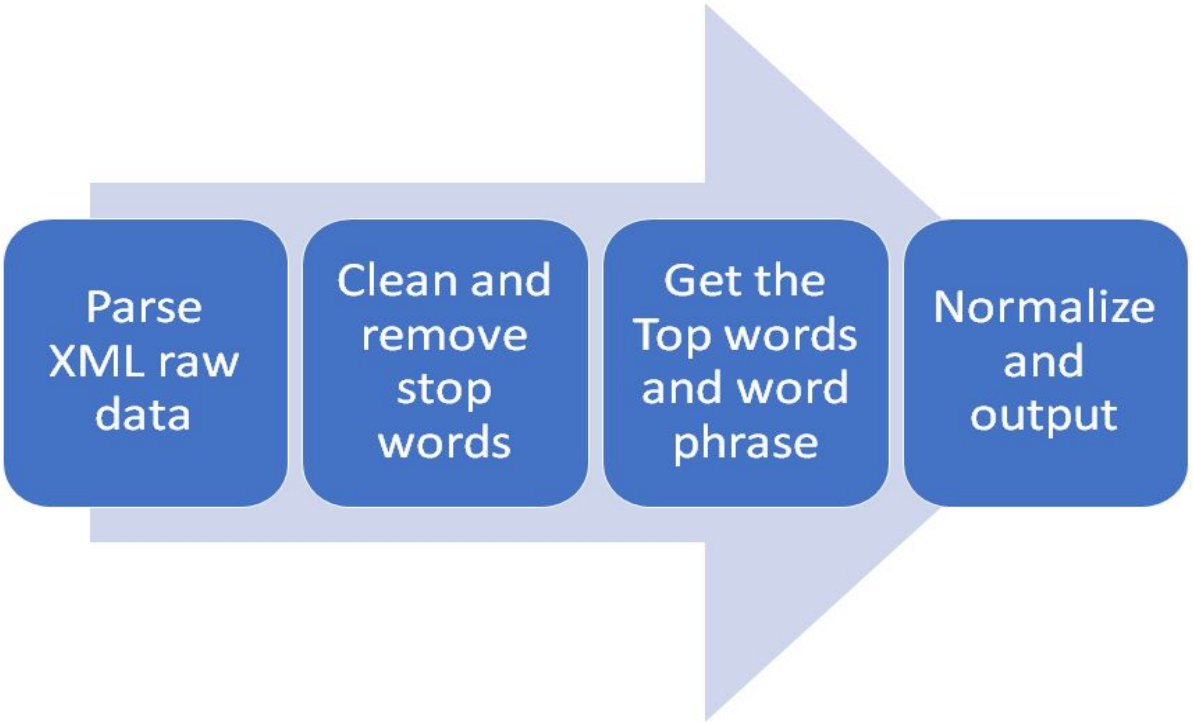


Figure 1. Workflow of Step 1.

Step 2: Selecting sample titles from target schools
Step 2 serves as a pre-processing step for pattern mining. We ranked top 10 universities around the world based on their publication and saved their titles in a long single string, but only those publications from US universities were used for pattern mining simply because English titles are easier to understand. The universities chosen are MIT, GT, and UMD.

Step 3: Pattern Mining
In this step, we loaded publication titles by the above universities and loaded the frequent patterns mined by CloSpan, a software package of mining closed sequential patterns in a sequence database.

Step 4. Feature pattern selection
Redundancy removal was performed so that duplicate items are removed from the mined frequent patterns. The resulting patterns from each university were ranked based on mutual information. The integrated patterns were finally converted to words via a built-in dictionary.

Technical Details:

a) Load, clean raw data, and tokenization

-input:

-dblp.xml is the source file downloaded from the dblp computer science bibliography

	Column_0
1	<?xml version="1.0" encoding="ISO-8859-1"?>
2	<!DOCTYPE dblp SYSTEM "dblp.dtd">
3	<dblp>
4	<phdthesis mdate="2016-05-04" key="phd/dk/Heine2010">
5	<author>Carmen Heine</author>
6	<title>Modell zur Produktion von Online-Hilfen.</title>
7	<year>2010</year>
8	<school>Aarhus University</school>
9	<pages>1-315</pages>
10	<isbn>978-3-86596-263-8</isbn>
11	<ee>http://d-nb.info/996064095</ee>

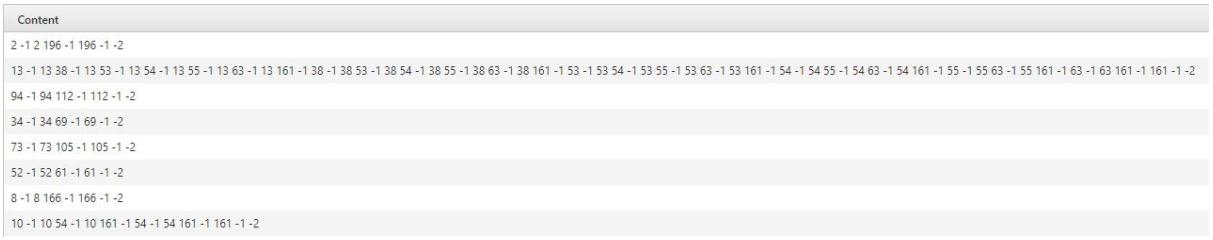
Figure 2. DBLP XML File

- implementation of the software
- script file: dblp_1.ipynb
- language: PySpark
- Environment: Azure Synapse(Spark)
- output
    - dblp_school.txt is the text documents that contain all school names that are in the same order as dblp_title.txt

Content
RWTH Aachen University, Germany
University of Satilde;o Paulo, Brazil
Massachusetts Institute of Technology, Cambridge, MA, USA
Massachusetts Institute of Technology, Cambridge, MA, USA
Massachusetts Institute of Technology, Cambridge, MA, USA
Massachusetts Institute of Technology, Cambridge, MA, USA
University of Satilde;o Paulo, Brazil
University of Satilde;o Paulo, Brazil
University of Satilde;o Paulo, Brazil

Figure 3. School Names in DBLP

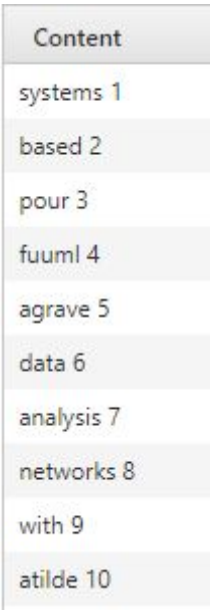
    - dblp_title.txt is the text document that contains all titles of papers published by varieties of colleges across the world. It shares the same order as dblp_school.txt. Most importantly, all of the words have been converted into integers where the mapping can be found in dblp_word.txt



Content
2 -1 2 196 -1 196 -1 -2
13 -1 13 38 -1 13 53 -1 13 54 -1 13 55 -1 13 63 -1 13 161 -1 38 -1 38 53 -1 38 54 -1 38 55 -1 38 63 -1 38 161 -1 53 -1 53 54 -1 53 55 -1 53 63 -1 53 161 -1 54 -1 54 55 -1 54 63 -1 54 161 -1 55 -1 55 63 -1 55 161 -1 63 -1 63 161 -1 161 -1 -2
94 -1 94 112 -1 112 -1 -2
34 -1 34 69 -1 69 -1 -2
73 -1 73 105 -1 105 -1 -2
52 -1 52 61 -1 61 -1 -2
8 -1 8 166 -1 166 -1 -2
10 -1 10 54 -1 10 161 -1 54 -1 54 161 -1 161 -1 -2

Figure 4. Integerized Publication Titles.

- dblp_word.txt is the dictionary that has the mapping between integers to words.



Content
systems 1
based 2
pour 3
fuuml 4
agrave 5
data 6
analysis 7
networks 8
with 9
atilde 10

Figure 5. The Word-to-integer Dictionary

b) Selecting Sample Titles from Target Schools:

- Input:
  - dblp_school.txt is the text documents that contain all school names that are in the same order as dblp_title.txt
  - dblp_title.txt is the text document that contains all titles of papers published by varieties of colleges across the world. It shares the same order as dblp_school.txt. Most importantly, all of the words have been converted into integers where the mapping can be found in dblp_word.txt
  - dblp_word.txt is the dictionary that has the mapping between integers to words.

- Process:
  - school_pattern_creator.ipynb is the pre-processing step that achieves the goal of selecting the sample school that users want to make the comparison. In our case, we select all of the titles published by 3 major U.S colleges.
  -
- Output:

The output is a series of titles grouped by each college. For example, in the picture below where you can find 0 is mapped to University of Satilde, therefore output0.txt includes all

titles belonging to University of Satilde. However, in order to improve interpretability, we choose output3.txt, output5.txt, and output8.txt as our samples. However, we encourage our users to explore more colleges.

- school0.txt
- school1.txt
- school2.txt
- school3.txt
- school4.txt
- school5.txt
- school6.txt
- school7.txt
- school8.txt
- school9.txt

```
0 RWTH Aachen University, Germany
1 University of Satilde;o Paulo, Brazil
2 Massachusetts Institute of Technology, Cambridge, MA, USA
3 Karlsruhe Institute of Technology, Germany
4 Technical University Munich, Germany
5 Dresden University of Technology, Germany
6 Darmstadt University of Technology, Germany
7 Joseph Fourier University, Grenoble, France
8 Georgia Institute of Technology, Atlanta, GA, USA
9 University of Maryland, College Park, MD, USA
```

c) Pattern Mining:

- Input:
  - school2.txt contains all titles from MIT
  - school8.txt contains all titles from GT
  - school9.txt contains all titles from UoM

- Process:
  - CloSpan is a closed pattern mining algorithm that is able to find a series of highly associated patterns in the sequence database. It was proposed by Yan et al. (2003). We use CloSpan to extract “most common” word or term patterns in each title grouped by each college.
    - spmf.jar is required to run CloSpan in the same directory. The following java code is to generate the closed frequent patterns for each college. There is only 1 main parameter, called support. It defines pattern commonality. Users can adjust the parameters to explore.
      - java -jar spmf.jar run CloSpan school3.txt output3.txt 1%
      - java -jar spmf.jar run CloSpan school5.txt output5.txt 0.5%
      - java -jar spmf.jar run CloSpan school8.txt output8.txt 3%
  - Output:
    - output2.txt is the frequent pattern from University of Maryland
    - output8.txt is the frequent pattern from MIT
    - output9.txt is the frequent pattern from Georgia Institute of Technology

d) Feature Pattern Selection:

- input:
    - school2.txt
    - school8.txt
    - school9.txt
    - output2.txt
    - output8.txt
    - output9.txt i
    - dblp_word.txt

- Process:
    - Remove Redundancy is to mitigate redundancy issues in the frequent pattern. For example, in the picture shown below, there are a lot of redundant words in a single transaction due to the way we pre-process the terms. And we found simply taking unique items in the pattern while keeping the word/term orders will be effective.
      For example, [1,-1,1,1,1, -1, 2, -2] will be reduced to [1, -1, 2, -2] where all positive integers represent a unique word while -1 separates two itemsets and -2 imply the end of the sentence.
    - Feature Pattern Selection by Mutual Information is to extract the most distinct pattern among all of the schools with high frequency. Two schools might have shared the same focus, for example, both MIT and GT might focus on “system” while MIT might pay more attention to “design” while GT might give more weight to “architecture”. Therefore, “system” is not the best candidate to define the difference between MIT and GT, while “design” and “architecture” are. We want to give more weight to “design” and “architecture” however not losing the importance of “system” as well.

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) log \frac{P(x,y)}{P(x)P(y)}$$

    - Word Conversion is the last step. Till the previous step, we worked on a series of numbers for each transaction that represents a unique word. Now we convert back to words based on the dictionary so that we are able to understand the meaning.

- Output:
    - Each college with its most distinct and representative words or terms to demonstrate their academic focus in computer science conferences. All of the words/terms have been ranked descendingly. We will get the top 5 terms/words for demonstration.

Conclusion:

We may sense a different focus for each school. For example, MIT focuses a lot on system control and power/energy-related topics; GT focuses more on programming and framework; UoM puts a lot of attention on wireless networks as well as object recognition, which might be in the Computer Vision area.

- MIT: “system control”, “power, energy”, “control, analysis”, “power, applications”, “systems, large”
- GT: “programming”, “framework”, “management”, “problems”, “approach”
- UoM: “resource, wireless”, “embedded, systems”, “networks, resource”, “recognition, object”, “social”

Demo Steps:

We’ve shared voiced video here

1. Download file annotation from our Github
2. Open Terminal and cd to the directory annotation
3. python school_pattern_creator.py
4. java -jar spmf.jar run CloSpan ./school_output/school9.txt ./pattern_output/output9.txt 1%
5. java -jar spmf.jar run CloSpan ./school_output/school2.txt ./pattern_output/output2.txt 0.5%
6. java -jar spmf.jar run CloSpan ./school_output/school8.txt ./pattern_output/output8.txt 3%
7. python pattern_decipher.py
8. you can find the final output in ./annotation/outcome

Team Responsibilities:

- Difan Gu is responsible for the core algorithm development including mutual information, data preprocessing, pattern mining and frequent pattern selection, also contributed to the final report, documentation and presentation.
- Wen Long is responsible for data preprocessing from XML, association mining. He also contributed to the final report and documentation.
- Yanyue Wang has brainstormed and researched publications concerning frequent pattern mining. She also documented the pattern mining process.

Reference:

KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining August 2006 Pages 337–346 https://doi.org/10.1145/1150402.1150441

CloSpan: Mining Closed Sequential Patterns in Large Datasets, by X. Yan, J. Han, and R. Afshar. Proc. of 2003 SIAM Int. Conf. Data Mining (SDM'03), 2003