ASSIGNMENT

Publication Records Analysis and Processing

AI6122 Text Data Management and Processing

September 2023

NANYANG TECHNOLOGICAL UNIVERSITY

1 Objective

The objective of this assignment is to let you getting familiar with the main components in end-to-end text management and processing applications, the challenges faced by each component and the solutions. Through this assignment, you shall also get hands on experiences on various packages available for information retrieval and natural language processing tasks.

2 Assignment Format

1. This is a group assignment. Each group has 4 to 5 students.
2. One report is to be submitted by *each group* and all members in the same group receive the same grade. However, **contributions of individual members** to the assignment shall be *cleared indicated* in the report. Group size is not a factor in grading.
3. You may use ANY programming language of your choice, *e.g.,* Java, Python, C#.
4. You may use any NLP, IR, and Machine Learning library/software as long as its license allows free use for education and/or research purpose. Some example packages are listed below for you to consider. However, relational database like MySQL is not allowed. Tools/packages that provide powerful high-level functions like Elasticsearch is not allowed as well.
   - All-in-one library: NLTK (Python), spaCy (Python), LingPipe (Java), Stanford NLP(Java), OpenNLP (Java)
   - Indexing and Search: Lucene (Java)

3 Assignment (100 marks)

The assignment consists of the following components: Dataset Analysis (30 marks), Development of a Simple Search Engine (30 marks), Development of a Research Trend Explorer (30 marks), and Development of an Application (10 marks).

3.1 Domain Specific Dataset Analysis (30 marks)

You are tasked to form three (3) datasets for analysis. Each dataset should contain about 10 - 20 documents in one selected topical domain. You may need to define "document" in your report, e.g., a thread is considered as a document or a post is considered as a document in an online forum. Each domain shall have its own linguistic characteristics with some specific terms specific to the domain. Examples are codes in programming forums, mechanical parts in manufacturing, chemical compounds, and maths equations in test papers. Some example domains are given as follows:

- Questions on StackOverflow, e.g., `https://stackoverflow.com/questions/63883827/`
- Patents for Jet Engine, e.g., `https://patents.google.com/patent/US2474359`
- Research papers in medical or chemical areas, e.g., `https://pubs.acs.org/doi/10.1021/jacs.0c07212`

- Life insurance policy document

Tokenization and Stemming. Tokenize all documents in each domain using a selected library (e.g., NLTK) and observe the tokens obtained. Discuss your observations from the following perspectives. Has the tokenizer correctly recognized the domain specific tokens? Use examples to illustrate what the expected tokens are, and what are not, particularly on the domain specific terms. Discuss how to identify the tokens that are incorrect through programs? If you were to improve the tokenizer, what are the possible solutions.

Perform stemming and compare the token distributions before and after the stemming (you may choose any stemming algorithm implemented in any toolkit). You may compare the number of distinct tokens, and the length distribution of the tokens. The length distribution can be compared in a plot: the x-axis is the length of a token in number of characters, and the y-axis is the number of tokens of each length. Discuss your findings.

Sentence Segmentation. Perform sentence segmentation on all documents in each domain. Compare the distribution of the sentence length in the three domains. Here, the x-axis is the length of a sentence in number of words/tokens, and the y-axis is the number of sentences of such length. Discuss your findings.

POS Tagging. Randomly select 3 sentences from each dataset, and apply POS tagging. Discuss the POS tagging results. Are the results as expected? Can the POS tagger well handle the domain specific terms?

3.2 Development of a Simple Search Engine (30 marks)

We will use the data collection published by DBLP in this assignment `https://dblp.org/faq/How+can+I+download+the+whole+dblp+dataset.html` DBLP provides open bibliographic information on major computer science journals, proceedings, and other form of publications. Proceedings are collections of papers presented in research conferences. DBLP currently contains more than 6.8 million publications (or papers), written by more than 3.3 million authors.

Write a search engine to index and search publications. Your index shall include at least the following fields: title of the paper, author of the paper, publication venue (e.g., journal or conference name), and publication year. You may use Lucene or other libraries specific to IR.[1] In this part of the assignment, you may use (i) One main IR specific library for most of the operations; (ii) Any other third-party libraries if and only if the main library does not provide the required functionality; and (iii) Any stopword list of your choice, if necessary. However, you are not allowed to use very high-level libraries like Elasticsearch.

In this search engine, each publication (or a paper) is a "document". Other than the necessary fields defined above, you are free to choose which additional fields to be included in your indexing. For all fields that are indexed, detail your choice of parsing/linguistic processing on the words/terms in the chosen fields, *e.g.,* whether to perform stemming, case folding, stopword removal, in these fields. Based on the number of "documents" to be indexed in the dataset, collect the time needed to index every 10% of the documents. Discuss your findings on the indexing time.

Your search engine should at least support free text keyword queries (including single keyword query and phrase query) on the textual field of a paper, e.g., title, publication venue. Top N (the number of N is configurable) results should be returned via the console[2] along with rank, scores, docID, and snippets when-

[1] See `http://en.wikipedia.org/wiki/List_of_information_retrieval_libraries` for a list.

[2] Note, a text-based command line system is sufficient; a GUI or web-based interface to the search engine is NOT encouraged.

ever possible. Your search engine shall also support search for publications meeting multiple requirements: e.g., containing a keyword "search", published in SIGIR conference, in the year of 2020.

Randomly choose a few queries (including both single keyword query and phrase queries), discuss whether the results returned by the search engine are as expected. You may also record the time taken to process a query.

3.3 Development of a Research Trend Explorer (30 marks)

With the help of your search engine (or other filtering tools), we now focus on the publications that were published in one particular conference over the years. You may select one conference from the following list: SIGIR, WSDM, CIKM, SIGMOD, VLDB, ICML, AAAI, ACL, EMNLP, or some major conferences of your interest. The chosen conference shall have publications for more than 15 years. After choosing one conference, we would like to get the research trend demonstrated over the years. The research trend can be represented by some key phrases that were popular among the papers published in a specific year, e.g., "Support Vector Machines", "Recommender System", "ChatGPT".

Discuss the criteria to select key phrases. Note that, a key phrase may contain one or more words. Then discuss how to find the representative key phrases in each year, considering the research topics of a conference in each year can be represented by a predefined fixed number of key phrases, e.g., 5. Then plot a graph or use other form of visualization to show the research trend of the conference from its very first version, to the last version. Note that, a key phrase may last for a few years. Based on the results obtained, discuss the limitations of your proposed solution.

3.4 Application (10 marks)

Define and develop a simple application based on the DBLP dataset. An example application is to search for similar research papers by title. You may define your own application with similar (estimated) difficulty level. Note that, application here means a small tool to analysis or to mine the data. Application here does not mean a web-based application or mobile app.

4 Submission of Report and Source Code

4.1 *Final Report in Hardcopy*

- The hardcopy report must be submitted on or before **31 Oct 2023** (Tuesday, Week 11), through SCSE General Office.
- The report must use the provided cover page, and the main content shall be formatted following the ACM "sigconf" proceedings templates[3] (either MS Word or Latex). The main content of the report ***must not exceed 10 pages***, *i.e.,* excluding cover page and appendix.
- DO NOT include in your report all the source code and complete results sets. However, you must include *code snippets* which are important for the main functions for your task. You should cite all third-part libraries used in your assignment.

[3] https://www.acm.org/publications/proceedings-template

- The report shall be printed in double-sided format whenever possible. A plastic cover or ring-binding leads to 2% penalty.
- Before submission, please read the hardcopy of your own report. **Make sure any words or pictures in your report are readable**.

4.2 *Final Report in softcopy, Source Code, and Documentation*

- An AI6122-Gxx.zip file containing the following files and folder shall be submitted: Report.PDF, Readme.txt, SourceCode.
  - The Gxx is your Group ID.
  - Report.PDF shall be the same as the hardcopy report submitted.
  - Readme.txt shall include
    - A link to download the third-party library if you used any in your assignment.
    - An installation guide on how to setup your system, and how to use your system (*e.g.,* command lines, input format, parameters).
    - Explanations of sample output obtained from your system.
  - SourceCode folder shall contain only your source code. The dataset and the libraries shall **NOT** be included in the softcopy submission to minimize the file size.
- Softcopy submission deadline: ***31 Oct 2023 11:59PM***. Late submissions are allowed but will be penalized by 5% every calendar day (until zero). The softcopy can be submitted for at most three times, only the last submission will be considered and time-stamped.