Knowledge Discovery in COVID-19 Open Research Dataset

COMP5434 Big Data Computing Project

Lecturer: LI Zecheng
zecheng-comp.li@polyu.edu.hk

TA: CHEN Zuyao
zuyao.chen@connect.polyu.hk

1 Introduction

The COVID-19 pandemic has sparked a global scientific effort to understand the virus, its transmission, and potential treatments. The CORD-19 dataset, a compilation of over 1,000,000 research articles related to COVID-19, provides a rich source of information for analyzing the evolving landscape of research.

This project explores the vast CORD-19 dataset, using data mining techniques to uncover hidden connections and patterns within the research landscape. By applying **frequent pattern mining**, **locality-sensitive hashing**, and **clustering algorithms**, we aim to identify key areas of research focus, discover relationships between different topics, and gain insights into the evolving trends in COVID-19 research.

2 Project Objectives

The primary objectives of this project are:

1. Analyze the CORD-19 dataset to understand the evolving landscape of COVID-19 research.
2. Identify key areas of research focus within the dataset.
3. Discover relationships and connections between different research topics.
4. Generate insights that can inform future research directions and collaborations.

3 Project Requirements

This project should be conducted following five key phases as below:

3.1 Data Preparation (10 points)

- Load the CORD-19 (subset) dataset from "meta_10k.csv" and "subset.zip" (json files), and print some meta information such as data length, the first n rows of the data.
- Apply text cleaning techniques to remove irrelevant characters, stop words [5], and punctuation.
- Perform some exploratory analysis: distribution of languages, histogram of publication years (showing the number of papers per year), and histogram of journals (showing the number of papers per journal).

3.2 MapReduce (20 points)

Your assignment entails the development of a program, utilizing the MapReduce paradigm, for the purpose of word counting and index building within the dataset. You should:

- Write a MapReduce-style program to count the frequency of words.
- Construct a stop word list to exclude meaningless words, and return the top-50 prevalent words.
- Write a MapReduce-style program to generate an index, which can facilitate the retrieval of document ids to the queried words.
- You can engage in iterative refinement of the stop word list to count domain-specific words, still return the top-50 words.

3.3 Association Analysis (20 points)

- Apply frequent pattern mining algorithms (e.g., Apriori, FP-Growth) to the processed document representations.
- Identify frequent co-occurring terms and analyze their significance within the context.
- Analyze relationships between discovered topics and research trends.
- Explore topic modeling techniques (e.g., Latent Dirichlet allocation (LDA)) to identify latent topics within the dataset.

3.4 Similarity Analysis (20 points)

- Choose feasible distance metric to measure the similarity between documents.
- Implement Locality-Sensitive Hashing (LSH) to find similar research papers based on their document representations.
- Experiment with different LSH families and parameters to optimize similarity calculations.
- Explore methods to account for semantic similarity using word embedding.

3.5 Clustering Analysis (20 points)

- Apply clustering algorithms (e.g., k-means, DBSCAN, hierarchical clustering) to group research papers based on their topics, publication dates, or other relevant features.
- Experiment with different algorithms and feature engineering techniques to improve clustering performance.
- Analyze the characteristics of each research cluster and visualize the results.

3.6 Insights and Recommendations (10 points)

- Identify key research areas based on the discovered knowledge.
- Explore the potential use of these insights to develop a recommendation system.

4 Tools and Technologies

This project is suggested to be implemented using the following tools:

- Scarfold: Python or PySpark [9]
- Computation: NumPy [6]
- Data Processing: Pandas [7]
- Data Mining Algorithms: scikit-learn [10], PySpark Mlib [8]
- NLP tools: NLTK [4], spaCy [12] (tokenization, tagging, parsing, named entity recognition, text classification and more.)
- Visualization: Matplotlib [2], seaborn [11]

You are also encouraged to implement algorithms from scratch, or explore some other Python libraries to complete your tasks:

- MLxtend [3]: A Python library of useful tools for the day-to-day data science tasks.
- Gensim [1]: A Python library for representing documents as semantic vectors.

Though deep learning is not the high priority of this course, you could leverage some deep learning techniques (e.g., BERT and LLM) to complete tasks if you are interested.

5 Project Report

Your report should contain 2 parts: group report and individual report.

5.1 Group Report (no more than 12 pages)

The group report is a critical part of your project, as it will demonstrate your understanding of key concepts and your ability to apply them to a real-world problem. Please adhere to the following structure and guidelines when preparing your group report:

- **Introduction.** You should write some preliminary discovery of the dataset. That is to say, you should give some brief discussion of the dataset. For example, you could describe the schema of the dataset.
- **MapReduce.** Describe how you implemented the MapReduce-style program in your project, including code snippets and a discussion of the logic behind your Map and Reduce functions.
- **Association Analysis.** Describe your implementation for association analysis, including the specific code snippets and an explanation of how it works to find associations within the data.
- **Similarity Analysis.** Explain the similarity metrics you implemented to assess the similarity between documents. Describe the methods you implemented to find similar documents, including code snippets and a discussion of its logic.
- **Clustering Analysis.** Explain the clustering algorithms (e.g., K-Means, Hierarchical, DBSCAN) you implemented, including the code and an explanation of the algorithm's logic. Discuss how these algorithms segment the data and their effectiveness.
- **Knowledge Discovery.** Present the insights and patterns your team uncovered through the implementation of the algorithms. Discuss how these findings contribute to solving the problem and understanding the domain. Try to talk about potential real-world applications of your discoveries and how they can inform further research.

5.2 Individual Report (no more than 500 words)

As part of our commitment to ensuring a fair and equitable assessment of each student's contribution to the group project, we require every student to submit an individual report. Your individual report could include following contents:

- **Highlighting Individual Contributions.** Clearly articulate the tasks and responsibilities you undertook within the group project, including but not limited to topic research, report writing, coding, algorithm design, results analysis.
- **Assessing Team Corporation.** Provide your view on how the team functioned together, including any challenges encountered and how they were addressed, discussion among group members, and distribution of tasks.
- **Reflection on Learning Outcomes.** Discuss the skills and knowledge you gained through your participation in the projects.

6 Project Evaluation

Your project will be assessed based on the following criteria. Each category will be scored to reflect the quality and completeness of the work submitted. The total score will determine the project grade.

1. **Code Quality and Functionality (40%)**
   - Correctness: Does the code implement required algorithms correctly? Are the results accurate and reliable?
   - Readability: Is the code well-organized and easy to read? Are variables and functions named descriptively?
   - Efficiency: Is the code optimized for performance? Are there any unnecessary computations or redundancies?
2. **Algorithm Description and Explanation (40%)**
   - Implementation Detail: Is there a thorough explanation of how the algorithms were implemented? Are the code snippets and descriptions detailed and accurate?
   - Algorithm Understanding: Does the report demonstrate a clear understanding of the algorithms and their application to the problem?
   - Problem-Solving: How effectively do the implemented algorithms address the problem?
3. **Analysis and Knowledge Discovery (15%)**
   - Insightfulness: Are the insights and patterns discovered through the analysis meaningful?

- Application: Is there a clear explanation of how the findings can be applied in a real-world context or for further research?
- Depth of Analysis: Does the report go beyond surface-level findings to explore deeper implications and relationships in the data?

4. **Report Quality (5%)**
    - Organization: Is the report well-structured? Are the sections clearly defined with appropriate headings?
    - Writing Quality: Is the report written in a clear, formal style and free of grammatical errors and typos?
    - Citations and References: Are all sources properly cited? Is the bibliography formatted correctly?
    - Adherence to Guidelines: Does the report meet the specified length and content requirements?

Final scores for individual reports will be adjusted based on team report scores. Individual report scores may fluctuate up or down a certain percentage from the team report score. The concrete percentage will be based on the individual's contribution and understanding as demonstrated in the individual report.

7 Submission Guidelines

Your final submission should contain the following three elements:

- Jupyter Notebook. A comprehensive `.ipynb` format notebook that includes all the code, analysis, and findings related to this project.
- Group Report: A detailed report prepared by the group, summarizing the project's objectives, methodology, analysis processes, and conclusions.
- Individual Report: A personal report from each group member, reflecting on their contributions, learning outcomes, and personal insights gained from the project.

For each group, pack all the files into a zip file and upload it once. No need to upload it repeatedly by every group member. All components must be submitted through Blackboard on or before **14 July 2024**. Late submissions will be subject to penalties as per the course policy unless an extension has been granted prior to the deadline.

8 No Plagiarism

Any evidence of plagiarism or academic dishonesty will result in a failing grade for the project and may lead to further disciplinary action.

References

[1] Gensim documentation. Available online at: `https://radimrehurek.com/gensim/auto_examples/`.

[2] Matplotlib documentation. Available online at: `https://matplotlib.org/stable/`.

[3] mlxtend documentation. Available online at: `https://rasbt.github.io/mlxtend/`.

[4] NLTK] documentation. Available online at: `https://www.nltk.org/howto`.

[5] NLTK's list of english stopwords. Available online at: `https://gist.github.com/sebleier/554280`.

[6] NumPy documentation. Available online at: `https://numpy.org/doc/stable/`.

[7] pandas documentation. Available online at: `https://pandas.pydata.org/docs/`.

[8] PySpark machine learning library (MLlib) guide. Available online at: `https://spark.apache.org/docs/latest/ml-guide`.

[9] PySpark overview. Available online at: `https://spark.apache.org/docs/latest/api/python`.

[10] scikit-learn examples. Available online at: `https://scikit-learn.org/stable/auto_examples/`.

[11] seaborn user guide and tutorial. Available online at: `https://seaborn.pydata.org/tutorial`.

[12] spaCy project. Available online at: `https://pypi.org/project/spacy/`.