Information Retrieval: Assignment 3
Plagiarism Detection

Prof. Toon Calders, Ewoenam Tokpo
{toon.calders, ewoenamkwaku.tokpo}@uantwerpen.be

Deadline: 05/01/2022

This project is to be executed in groups of 2 students. For further inquiries about the project, please email *ewoenamkwaku.tokpo@uantwerpen.be*.

In this project, your task is to build a plagiarism detector to detect plagiarised news articles. The goal of this project is to apply some of the theoretical concepts of Locality Sensitive Hashing discussed in class in a practical context. This will give you hands-on exposure as to how Locality Sensitive Hashing can be applied in real-world cases and how to implement it for such uses.

1 Introduction

"CNN has fired a news editor in its London bureau for repeated plagiarism offenses, it announced in an editor's note today."-Erik Wemple, The Washington Post. Headlines like this illustrate the problem of news plagiarism faced by many media houses. One of such media houses has collected several news articles from the internet with the suspicion that their news articles are being plagiarised by some blogs and media houses. Considering the size of the corpus they have obtained, it will be strenuous for a manual plagiarism check to be done, hence, your group has been tasked to build a plagiarism detection system that can identify highly similar news articles from the corpus.

2 Dataset

The dataset for this project is a corpus of news articles. There are two available corpora to work with; a large version and a small version.

The large dataset is a csv file *news_articles_large.csv* containing 10000 news articles. The first column is the unique ID for each news article starting from 0 to 9999, and the second column contains the main news article. The small dataset, *news_articles_small.csv* consists of 1000 news articles in the same format as the former.

The small dataset can be used for the experiments and analysis that will be performed in the report whereas the large dataset will be used to generate the final result for submission.

3 Assignment

Your task for this project is to implement a plagiarism detector using Locality Sensitive Hashing on the dataset provided.

Suggested steps:

- Similarity analysis of the small dataset (*news_articles_small.csv*): Calculate the Jaccard similarity between each pair and plot a bar graph of the number of documents per a given range of similarity percentage; see wiki-plagiarism example in lecture slide number 85. The similarity scores calculated here will be used as ground-truth data to evaluate the LSH system that will be implemented.

- Preprocessing of data, shingling, and minhashing to generate a signature matrix using *news_articles_small.csv* dataset.

- Implementation of LSH on the signatures matrix generated. It is okay to use already existing libraries or codes to implement these first two steps, although implementing them on your own will be a plus.

- Evaluate varying values of signature length M, and similarity threshold s and analyse how these affect the performance of the system in detecting duplicates and near-duplicates. Use the bar plot above to estimate a suitable range to vary the similarity threshold; see lecture slide 93. You can carry out the evaluation by using some plagiarised documents as queries and analysing the sensitivity, specificity, or precision values based on the candidate pairs produced. Analyse these results and discuss your observations in the report.

- From the analysis above, choose suitable values for signature length M, number of rows per band r, number of bands b and similarity threshold s that gets most pairs with similar signatures and eliminates most pairs that do not have similar signatures.

- Implement LSH on *news_articles_large.csv* and find all pairs of documents that are duplicates or near-duplicates (that have a similarity of at least 0.8). Save this file as ***result.csv*** which should be included in the GitHub repository. Save the results such that each row represents a plagiarised pair, and each column represents one document of the pair - doc_id1, doc_id2.

Please specify all external/existing libraries or codes you used in the report.

4 Deliverables

1. The code of your project. Do not include bulky software libraries or large datasets in emails. The preferred way to share code is via a link to a publicly available GitHub repository. Include the link in your report. The ***result.csv*** file should also be included in the GitHub repository.

2. The report in pdf format, to be submitted via BlackBoard. Do **not** submit zip-files, word documents, etc., only the submission of a single pdf file will be accepted. The report should be approximately 10 pages in length.

   The report should include the following:

   - A brief introduction.
   - Details of your implementation: This should highlight the various aspects or phases of your implementation. It should also capture the main algorithms, frameworks, libraries and environments used.
   - Analysis and evaluation of the work. Tables and graphs showing the results of your experiments and evaluations should be captured here.

A Note on Plagiarism

There is absolutely nothing wrong with using existing materials, you will even be commended for not reinventing the wheel, as long as you are not violating the copyright of other authors. Nevertheless, it is expected from you to clearly indicate whenever you used material that was not created by yourself. Clearly indicate in your submissions which parts constitute original work, which parts are taken from other works, and which parts were adapted from external sources. These sources have to be properly acknowledged in all your submissions. Concretely, this means at least the following guidelines are observed:

- Papers, books, webpages, blogs, etc. that were inspected while making the assignment will be referenced in a separate section "References". Citations to these materials are included in the text where appropriate.

- Text fragments exceeding one sentence that are copied from other sources are clearly marked as such. You could for instance include quoted text, definitions, etc. in italics, followed by a reference. An example of how to do this: Bela Gip (2014) defines plagiarism as *"The use of ideas, concepts, words, or structures without appropriately acknowledging the source to benefit in a setting where originality is expected"*

  References: (at the end of the document) Gipp, Bela. Citation-based plagiarism detection. Springer Vieweg, Wiesbaden, 2014. 57-88.

- When using code from other sources, indicate so in the report, and in the source code. This could for instance be done by adding a comment with a reference to the source of the function for each function that was copied from another source. It is recommended to include a separate folder "sources" in your GitHub repository with the original files from other authors that you used. Include source in the message of your commits.