CSCI 1952-Q: Algorithmic Aspects of Machine Learning (Spring 2023)

Coding Assignment 3

Due at 2:30pm ET, Monday, May 8

Getting Started.

- You are free to use any programming language of your choice.
- You cannot use functions/packages that directly solve the problem.
- You may use resources on the web and you are encouraged to discuss with other students. You must write your code independently and acknowledge who you discussed with and what resources you used. Failure to provide such acknowledgment is considered a violation of academic integrity.

Assignment Overview. In this assignment, you will implement scalable algorithms for finding similar documents. You will be given a list of articles from a real-world dataset. Your task is to find the articles that are most similar to each other.

Input. You are given an input file `documents`. The first line of this file contains three integers n, k, and q. Next are n lines, where each line specifies a document. Each document is a sequence of alphanumeric tokens separated by a single space. See the Dataset section below for more details on how these documents are generated.

Your task is to output q pairs of documents that are similar to each other, where similarity is measured by the Jaccard similarity between the sets of k-shingles of two documents. Note that you cannot choose k. Your solution will be evaluated based on the specified k.

Output. The output file should have q lines, each with two integers $1 \leq i \neq j \leq n$, specifying q pairs of documents. The q pairs of integers must be distinct, where (i, j) and (j, i) are considered the same pair. Note that the documents are numbered from 1 to n.

Submission.

- Your submission should consist of exactly 3 files:
  1. an output file `lsh_ans` in the specified format,
  2. a text file (e.g., `.cpp`, `.py`) containing your source code, and
  3. a `pdf` file containing a detailed explanation of your approach.
- We may ask you to show us that running the submitted code does produce the submitted output file.

Evaluation. Let S_i denote the set of all k-shingles (i.e., substrings of length k) that appear at least once in document i (without eliminating whitespace). For this assignment, we define the similarity between document i and document j as

$$\mathrm{sim}(i, j) = \frac{|S_i \cap S_j|}{|S_i \cup S_j|} \; .$$

Suppose your output is $(i_1, j_1), \ldots (i_q, j_q)$. Your solution will be evaluated based on the minimum similarity of the pairs in your output.

$$F = \min_{1 \le \ell \le q} \mathrm{sim}(i_\ell, j_\ell) \; .$$

Grading. This assignment will be graded out of 6 points:

- (1 point) Your code should have good readability and should be well-commented.
- (1 point) Your explanation `.pdf` must be typed (e.g., MS Word or LaTeX). You should give an overview of your ideas and your approach in the first 2 pages. Material beyond the first 2 pages will be read at the sole discretion of the instructor/TAs.
- (4 points) You will get a score of $(30F - 1.4)$ where F is the minimum similarity as defined earlier. If the score is lower than 0 or higher than 4, it is set to 0 or 4. In particular, you will receive full credit if your minimum similarity F is 0.18 or higher.
- (1 bonus point) you will receive 1 bonus point if your minimum similarity F is among the top 20% of all received submissions.
- We may deduct up to 4 points for any formatting error in your output (including but not limited to, not naming the output file `lsh_ans`, not outputting q lines, or not outputting $2q$ distinct pairs of integers between 1 and n).

Dataset. The input file was processed from the WikiText Long Term Dependency Language Modeling Dataset [MXBS17]. More specifically, the `WikiText-103` word level dataset was used [1]. We merged the training, validation, and test datasets. This dataset contains 28592 articles chosen from the set of verified Good and Featured articles on Wikipedia [2].

For this assignment, we processed the `WikiText-103` dataset as follows. First we identified the title of all articles using the regular expression “ `\n = [^=]*[^=] = \n \n`”. That is, the previous line and the next line must be empty (i.e., a single space) while the current line starts and ends with exactly one pair of “ `=` ”. Then we convert consecutive whitespace characters to single space “␣” and put one article on each line (keeping the article title and section titles). Finally, we converted all letters to lowercase and removed all tokens with non-alphanumeric characters (e.g., “I-95”).

Remarks/Hints. You are free to use any algorithms to find similar articles.

- One possible approach is to use MinHash and locality sensitive hashing.

[1] See `https://blog.salesforceairesearch.com/the-wikitext-long-term-dependency-language-modeling-dataset/`. This dataset is available under the Creative Commons Attribution-ShareAlike License.

[2] See `https://en.wikipedia.org/wiki/Wikipedia:Good_articles` and `https://en.wikipedia.org/wiki/Wikipedia:Featured_articles`.

- A less systematic approach (which is viable for this assignment) is to check the similarity of $t \geq q$ pairs of articles and output the top q pairs. Choosing the value of t allows you to make a tradeoff between runtime and solution quality. One can consider checking all pairs among the first $\Theta(\sqrt{t})$ documents, or sample t pairs uniformly at random.
- As a sanity check, for the specified $k = 6$, the first two documents have $|S_1| = 11018$ and $|S_2| = 11112$ unique k-shingles respectively. One can verify that $|S_1 \cap S_2| = 2160$ and therefore $\text{sim}(1, 2) \approx 0.108$.
- The dataset (and hence the input file) contains duplicate articles, which have similarity 1. You are allowed to output these duplicate articles as a pair.
- The titles of all 28592 articles are provided in a supplemental file `wiki.titles`.

Optional Tasks. After completing the assignment, you can explore the following questions. There are no bonus points for answering these questions.

- Can we estimate the distribution of the similarity between a pair of random articles drawn from this dataset?
- How does choosing a different k affect the most similar pairs, the similarity distribution of two random documents, and the running time? How would you choose the value of k?
- What happens if we work with multi-set of k-shingles (i.e., count how many times a k-shingle occurs) and use the Jaccard similarity for multi-sets?
- Recall that topic modeling and matrix factorization can be used to measure similarity of documents. Compare the most similar pairs found by the two approaches (topic modeling and k-shingles).
- What possible tasks (e.g., image search, nearest neighbor search, audio fingerprint) and datasets are suitable for testing a scalable implementation of finding similar items?

References

[MXBS17] S. Merity, C. Xiong, J. Bradbury, and R. Socher. Pointer sentinel mixture models. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*. OpenReview.net, 2017.