CSCI 1520: Algorithmic Aspects of Machine Learning (Spring 2025)

Coding Assignment 1

Due at 11:59pm ET, Thursday, Feb 27

Getting Started.

- You can use any programming language for this coding assignment.
- You cannot use packages or functions that directly solve the problem.

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

Input. You will be given an input file `documents`. The first line of this file has three integers: n, k, and q. This is followed by n documents, one on each line. Each document is a sequence of alphanumeric tokens separated by a single space.

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.

Output. The output file should have q lines. Each line should have two integers $1 \leq i \neq j \leq n$, separated by a single space. These q pairs of integers must be distinct, where (i, j) and (j, i) are considered the same pair. Note that 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.
  3. A `.pdf` file providing 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 in document i (without removing spaces). For this assignment, the similarity between document i and document j is defined as

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

Note that the value of $k = 6$ is fixed. You cannot choose k.

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

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

Grading. This assignment will be graded out of 14 points:

- (3 point) Your code should have good readability and should be well commented.
- (3 point) Your explanation `pdf` must be typed (e.g., MS Word or LaTeX). You should give an overview of your ideas and approach in the first 2 pages. Material beyond the first 2 pages will be read at the discretion of the instructor/TAs.
- (8 points) You will receive a score of $(60F - 2.8)$, where F is the minimum similarity 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 $F \ge 0.18$.
- (2 bonus point) You will receive 2 bonus point if your minimum similarity F is among the highest 20% of all received submissions.
- We may deduct up to 8 points for any formatting error in your output (including but not limited to, not naming the output file `lsh_ans`, not outputting exactly q lines, or not outputting exactly q distinct pairs of integers between 1 and n).

Dataset. The input file was obtained from the WikiText Dataset introduced in [MXBS17]. Specifically, the `WikiText-103` word level dataset was used[1]. This dataset contains 28592 articles selected from verified Good and Featured articles on Wikipedia [2].

For this assignment, we processed the `WikiText-103` dataset as follows: We used the regular expression " `\n = [^=]*[^=] = \n \n`" to find the title of each article. We converted all letters to lowercase and removed all tokens with non-alphanumeric characters (e.g., "I-95"), converted consecutive whitespace characters to a single space, and placed one article on each line (keeping the article/section titles).

Remarks/Hints. You are free to use any algorithms to find similar documents. Due to this reason, the following hints may not apply to your solution.

- One possible approach is to use MinHash and locality sensitive hashing.
- The dataset (and the input file) contains duplicate articles, which have similarity 1. You are allowed to output these duplicate articles as a pair.
- As a sanity check, for $k = 6$, the first two documents have $|S_1| = 11018$ and $|S_2| = 11112$ unique k-shingles, and $|S_1 \cap S_2| = 2160$, so $\mathrm{sim}(1, 2) \approx 0.108$.
- A less systematic approach is to check the similarity of t pairs of articles and output the top q pairs. One can check all pairs for the first $\Theta(\sqrt{t})$ documents, or sample t pairs uniformly at random. One can choose the value of t to trade off runtime and solution quality.
- The titles of all 28592 articles are provided in a supplemental file `wiki.titles`.

[1] See `https://blog.salesforceairesearch.com/the-wikitext-long-term-dependency-language-modeling-dataset/`, 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`.

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

- What is the distribution of the similarity between a random pair of articles in this dataset?
- How does the value of k affect the similarity distribution, most similar pairs, and runtime?
- What if we work with multi-set of k-shingles and use the Jaccard similarity for multi-sets?
- Recall that topic modeling and matrix factorization can be used to measure the similarity of documents. Compare the most similar pairs found by topic modeling and k-shingles.

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.