Overview:

Dataset: *similarity.txt* (≈1.5k paragraphs), with each paragraph treated as a document.

We implement a manual LSH pipeline—shingling, Minhashing, and banding—to identify similar paragraphs. Each paragraph is one document. Exact Jaccard on shingles is used for evaluation; Minhash agreement provides an estimate.

Representation & Preprocessing

Lowercasing, punctuation normalization, and whitespace collapse. Paragraphs split on blank lines. Word-level shingles at k=3 (baseline) and k=5 (comparison). Similarity = Jaccard on shingle sets; Minhash uses num_perm hash functions to form signatures. Shorter and more structured texts are more sensitive to increasing k (3→5) because small edits remove proportionally more k-grams.

Why LSH vs. Naïve

All-pairs exact Jaccard is $O(N^2)$. LSH compresses documents into signatures and uses banding to generate a small candidate set. Trade-off: strict banding (fewer bands, more rows) can miss true pairs; looser banding increases recall but may admit near-misses.

Results & Sensitivity

Parameter sweep (times and candidates are from terminal outputs):

k	num_perm	bands×rows	total (s)	cands	Top-5 MAE
3	50	10×5	0.22	82	0.000
3	100	20×5	0.43	103	0.000
3	200	25×8	0.87	69	0.000
5	100	20×5	0.40	60	0.000

k=3 produced 5,574 unique shingles; k=5 produced 6,209. At k=5 with 100 permutations, candidates dropped 103→60 with similar sub-second runtime (precision↑, recall↓).

Looser banding to raise recall (verbatim):

num_perm=100, bands=25, rows=4 (bands*rows=100)

Times: build=0.01s, sig=0.38s, lsh=0.03s, total=0.43s

Candidates generated: 178

Near-duplicates under 100 perms, 25×4 (selected):

(i,j)	Jaccard	Est.	len_i/len_j
(15,53)	0.8235	0.8100	17/18
(13,53)	0.8235	0.8100	17/18
(14,53)	0.8235	0.8100	17/18
(12,53)	0.8235	0.8100	17/18
(19,61)	0.7222	0.7600	18/17
(17,61)	0.7222	0.7600	18/17

(18,61)	0.7222	0.7600	18/17
(16,61)	0.7222	0.7600	18/17

Top-5 overall (baseline k=3, 100 perms, 20×5): all exact duplicates (Jaccard=1.000).

- (121,277) Jaccard=1.000 est=1.000 len=75/75
- (298,309) Jaccard=1.000 est=1.000 len=81/81
- (266,286) Jaccard=1.000 est=1.000 len=72/72
- (164,229) Jaccard=1.000 est=1.000 len=76/76
- (287,318) Jaccard=1.000 est=1.000 len=75/75

Qualitative Analysis

Correct near-duplicate: (15,53) J=0.8235 (est 0.8100). 15: A centuries old windmill turns gracefully in the breeze overlooking a patchwork of fields and quiet lanes... 53: A centuries old windmill turns gracefully in the breeze overlooking a patchwork patchwork of fields and quiet lanes... The duplicated token and parallel phrasing explain high overlap without being identical.

Borderline/misleading example: (6,30) J≈0.6667 (est 0.7200). Despite frequent shared phrases, the meaning diverges—a reminder that shingle overlap can reflect structure rather than semantics.

Sensitivity Discussion

Candidate counts were 82 → 103 → 69 as num_perm increased 50 → 100 → 200 (more bucket collisions at 100, then fewer at 200 because the factorization 25×8 is stricter than 20×5). k: moving 3→5 increased unique shingles (5,574→6,209) and reduced candidates (103→60). Banding: increasing bands (e.g., to 25×4) raised recall and surfaced near-duplicates; if highly similar docs land in different buckets, the banding is too strict for the distribution—decrease rows per band (or increase bands) at fixed num_perm, or consider increasing num_perm for more stable estimates.

Proposed Improvement

Re-rank LSH candidates with TF-IDF cosine similarity. This reduces structure-only false positives among J≈0.6–0.8 pairs while preserving recall, since re-ranking is applied only to the small candidate set.

Takeaways

- Manual LSH meets the goal: fast candidate generation with sub-second runs on this corpus.
- k and banding control precision/recall; num_perm stabilizes estimates.
- A light TF-IDF re-rank makes final results more semantically meaningful.

Limitation: Character k-grams or a TF-IDF re-rank can mitigate structure-only matches and micro-edits that inflate Jaccard.