Δ. Retrieval and ranking

The corpus to be indexed for search contains three documents *doc1*, *doc2* and *doc3* containing text as described in Table 1. Answer the below questions on this content.

doc	text
doc1	elvis presley pop pop presley pop elvis presley pop elvis presley pop pop
doc2	elvis mississipi life elvis life mississipi elvis life elvis
doc3	elvis pop music elvis presley pop music elvis music pop elvis presley pop music elvis

Table 1: Content in the three documents *doc1*, *doc2* and *doc3*.

Δa. Fill in the term-frequencies (tf) and inverse-document-frequencies (idf) in Table 2 and construct an inverted index for the above documents with term frequency and position information. Add the postings list for each of the terms in Table 3.
10 points

	elivs	presley	mississipi	pop	music	life
idf						
tf *doc1*	3	4	0	6	0	0
tf *doc2*	4	0	2	0	0	3
tf *doc3*	5	2	0	4	4	0

Table 2: Prepare frequency table.

term	postings list
elvis	

Δb. Identify the set of relevant documents with their final ranking in Table 4 for each of the queries listed there using cosine similarity with two different feature vectors; tf vectors and tf-idf vectors. **OR** requires either of terms to occur to be relevant while **PHRASE** enforces ordering.
12 points

	query = **OR(elvis music)**		query = **PHRASE(elvis presley)**	
Rank	tf vec *doc*: score	tf-idf vec *doc*: score	tf vec *docN*: score	tf-idf vec *docN*: score

Φ. Deduplication and MapReduce

Consider the MinHash method of deduplication of documents described above including the banding technique.

1. A corpus has N=100M (million) documents each of size w=100 KB from which the vector $L_d \in \{0,1\}^{1\times V}$ is derived indicating the occurrence (or otherwise) of the V shingles in document d and $s_{ij} = J(L_i, L_j)$ denotes the Jaccard similarity of the corresponding doc pair (d_i, d_j).
2. Assume $b = 256$ bands of $r = 32$ rows each are hashed to `unit32` buckets for banding.
3. A MapReduce cluster with the configuration described in Table 5 is used for computations.

Number of nodes in cluster	$U = 10$
250GB HDD disk attached to each node	1 MB read/write 0.25 ms
Inter-node network link	1 Gbps
Job scheduler overhead	2 s
Map task startup overhead	200 ms
Map CPU cost per raw input	0.20 ms
Reduce CPU cost per reducer input	0.10 ms

Table 5: MapReduce cluster configuration parameters.

Φa. What is the probability that a pair of documents (d_i, d_j) with Jaccard score s_{ij} will make it to the candidate list of duplicates D_c after applying the banding technique?
2 points

Φb. If one were to use MapReduce to implement it, what would the map and reduce functions look like when distributing documents across mappers? Use the following notation to describe it very concisely (anything else necessary should be defined clearly before use).

1. $\mathcal{H} = \{h_i : N \to N\}$ is a family of hash functions used for signatures and $h : N \to M$ for banding technique with b bands of r rows each.
2. $\mathbb{I}[c]$ denotes the indicator function returning one when condition c is true and zero otherwise.
3. $[f(i)]_{i=p}^{q}$ denotes the vector with values from applying function f with parameters $i \in [p, q]$ in that order.
4. $\text{argmin}_i(f(i))$ returns an argument i for which $f(i)$ takes the minimum value.
5. The `construct_list` can be used to aggregate values into a list and `all_pairs` will return all distinct pairs of elements from the input list.

10 points

Φc. Compute the time taken to run the MapReduce job for de-duplication of the the document corpus in the above described setting. Choose number of mappers and reducers to minimize for overall time while ascertaining (by explicitly computing volume of data generated before/after map and reduce steps) it allocates sufficient disk space for the tasks. Explicitly state any assumptions made or define variables used. Assume data is already available on the cluster for reading and the final output needs to be written back to the same distributed file-system.

10 points

Hashing

Φd. Consider hash function h^{nm} that uniformly maps keys to buckets. A collision pair C_{ij} is a pair of keys (i, j) that are hashed into the same bucket.

1. What is the expected number of collision pairs $\mathbb{E}[\sum_{i<j} C_{ij}]$? .
2. What is the expected number of empty buckets after hashing all n keys?

4 points

Φe. Let the density function describing the probability of documents being duplicates follow $\mathbb{P}(s) \sim \mathcal{K}(s; a=2, b=2)$ and the fraction of pairs from the N document corpus with Jaccard similarity of s follows $\mathcal{K}(s; a=2, b=1)$ where $\mathcal{K}(s; a, b) = abs^{a-1}(1-s^a)^{b-1}$ is the Kumaraswamy's double-bounded distribution. If those with $s > \tau$ were marked as duplicates what would be the expected number of false-positives and false-negatives.

8 points