Clustered Elias-Fano Indexes

ACM Transactions on Information Systems (TOIS), 2017

Giulio Ermanno Pibiri **Rossano Venturini**

University of Pisa, Italy

1. Inverted Index Compression

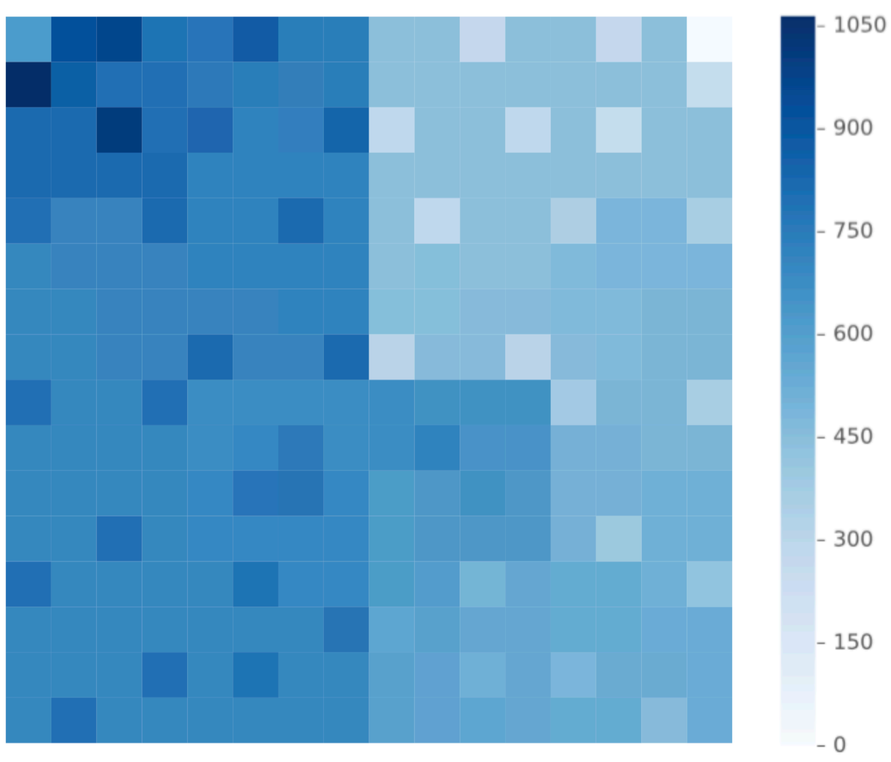
Inverted indexes are collections of sorted integer sequences, called ***inverted lists***. The inverted list of the term t stores the sequence of the identifiers of the documents (docIDs) that contain t.

Inverted lists can be tens of millions long, thus they must be compressed to allow efficient query processing.

2. Inverted Lists are Redundant

Inverted lists naturally present some amount of redundancy, i.e., they tend to share many docIDs because the indexed documents are likely to share many terms.

For example, the picture below shows the top-300 most frequent docIDs for a set of ≈1000 inverted lists drawn at random from the Gov2 dataset. We plot their frequencies along a Hilbert curve in order to highlight the regions of redundant docIDs.



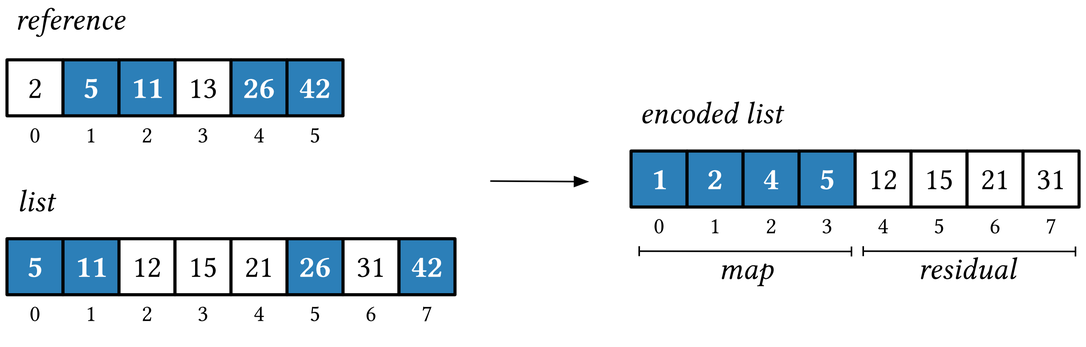
However, compressors for inverted lists represent each list individually and, thus, none of such techniques exploits the redundancy that may exist between two or more lists.

In order to exploit this repetitiveness, we propose a ***clustered*** inverted index representation.

3. Clustered Representation

Clusters of "similar" inverted lists, i.e., the ones sharing as many docIDs as possible, are created using a k-means-like algorithm.

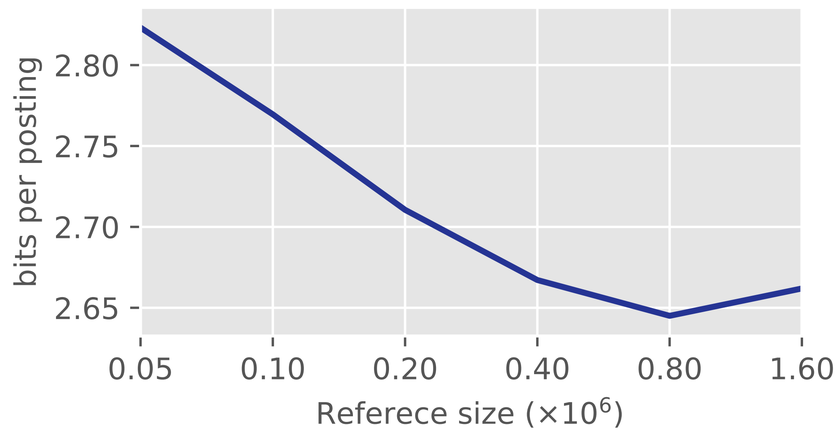
For each cluster, a ***reference*** list is synthesized. This list contains the most frequently occurring docIDs in the cluster lists.

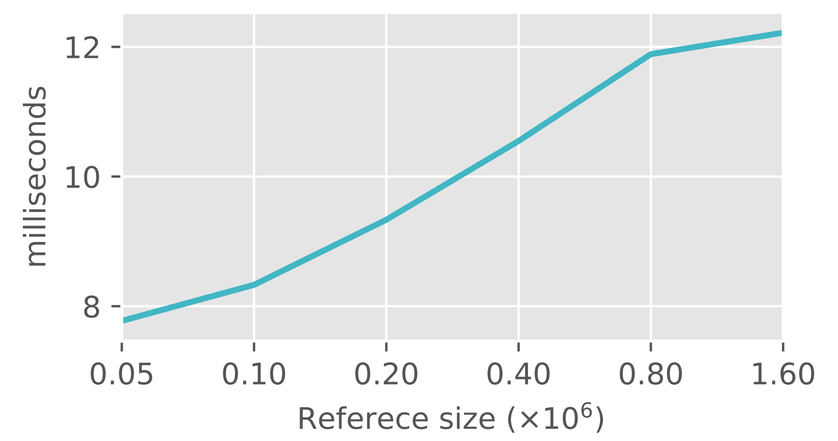


Each list is encoded ***relatively*** to the reference: the integers in common between the list and the reference (*map*) are represented in a smaller universe. We use *partitioned Elias-Fano* (PEF) to compress both *map* and *residual* segments.

4. Space/Time Trade-offs

By varying the size of the reference list for each cluster, we obtain space/time trade-offs as exemplified below using the Gov2 dataset and performing boolean AND queries from the TREC 06 query log.





5. Analysis in Selected Trade-off Points

We compare our *clustered* representation (CPEF) against PEF in three selected trade-off points – MIN, MID and MAX – corresponding to different reference list sizes.

	Gov2		ClueWeb09	
PEF	2.94		4.80	
CPEF@MIN	2.77	(−6%)	4.66	(−3%)
CPEF@MID	2.71	(−8%)	4.57	(−5%)
CPEF@MAX	2.65	(−10%)	4.50	(−6%)

Average number of bits per encoded docID.

	TREC 05		TREC 06	
PEF	3.7		6.1	
CPEF@MIN	5.3	(+43%)	8.3	(+36%)
CPEF@MID	5.9	(+59%)	9.3	(+52%)
CPEF@MAX	7.8	(+111%)	11.9	(+95%)

(a) Gov2

	TREC 05		TREC 06	
PEF	14.6		17.7	
CPEF@MIN	17.7	(+21%)	21.2	(+20%)
CPEF@MID	20.6	(+41%)	25.0	(+41%)
CPEF@MAX	29.1	(+99%)	35.6	(+101%)

(b) ClueWeb09

Average milliseconds spent per boolean AND query.

6. Further Resources

For more experiments, refer to Chapter 4 and 7 of

Giulio Ermanno Pibiri. *Space- and Time-Efficient Data Structures for Massive Datasets.* Ph.D. Dissertation. University of Pisa. 2019.
`http://pages.di.unipi.it/pibiri/papers/phd_thesis.pdf`

Source code.
`https://github.com/jermp/clustered_elias_fano_indexes`