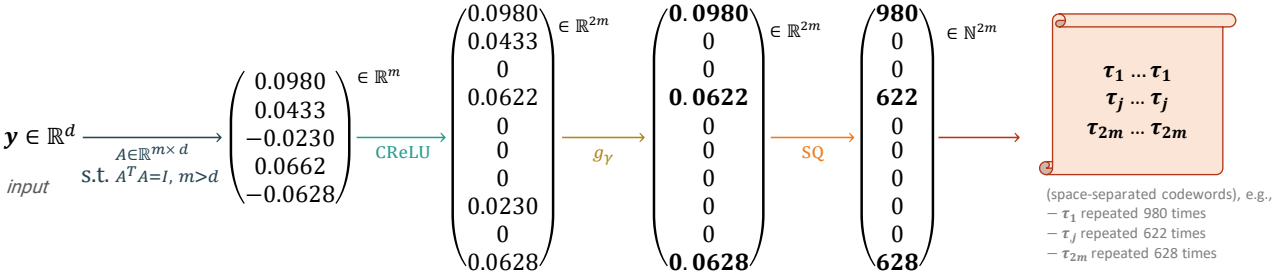
Vec2Doc: Transforming Dense Vectors into Sparse Representations for Efficient Information Retrieval

Fabio Carrara, Claudio Gennaro, Lucia Vadicamo, Giuseppe Amato

Institute of Information Science and Technologies, National Research Council of Italy (CNR-ISTI)

fabio.carrara@isti.cnr.it https://github.com/fabiocarrara/str-encoders



1. Semi-orthogonal transformation

To expand the dimensionality of the vectors while preserving the dot product and distributing information across the different dimensional components

2. Positivization

To transform the vector into a positive one. We used the Concatenated Rectified Linear Unit Transformation

$CReLu(\boldsymbol{v}) = \max([\boldsymbol{v}, -\boldsymbol{v}], \mathbf{0})$

3. Sparsification

Apply a component-wise thresholding Function, e.g.

$g_\gamma(x) = x, if\ x > \gamma$

4. Integer Quantization

Transform real components into integer, e.g., using a quantization factor $s > 1$

$SQ(x) = \lfloor s\, x \rfloor$

4. Surrogate Text

A document is formed repeating the i-th term a number of times indicated by the i-th component of the text frequency vector.

5. Full-text Indexing

Text is indexed with search engines based on the vector model (e.g., Apache Lucene / Elasticsearch). TF-only scoring for inner product.

Vec2Doc Transformation to Term Frequency Vector

The dense real vector $\boldsymbol{y} \in \mathbb{R}^n$ *is transformed into a positive integer vector* $\overline{\boldsymbol{y}} \in \mathbb{N}^{2m}$ *representing a term frequency vector over a codebook* $C = \{\tau_1, \dots, \tau_{2m}\}$ *of* $2m$ *terms*

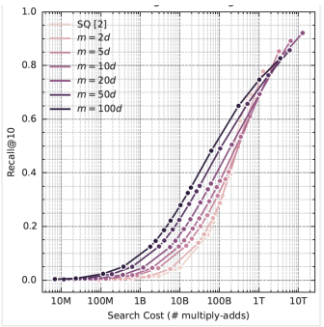
Experiments

- Maximum inner product search on dense real vector datasets
- In-memory inverted index (independent from specific engine implementation
- Compared techniques:
  - Vec2Doc versus Scalar Quantization (SQ)
  - Voronoi Partitioning Vec2Doc versusVoronoi Partitioning Scalar Quantization (VP-SQ), which are Voronoi-partitioned versions of the two approaches using a different codebook for each partition
- Params:
  - Vec2Doc: number of rows m of the semi-orthogonal transformation, sparsification factor γ (fixed quantization factor $s = 10^5$)
  - SQ: sparsification factor γ (fixed the quantization factor $s = 10^5$)
  - VQ-X: number of partitions, number of accessed partitions at query time, plus the parameters of the underline "X" technique

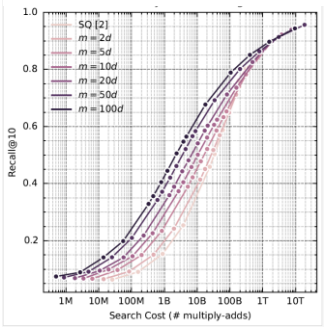
Results

- Effectiveness (Recall@10) vs Efficiency (Search Cost, Index Size)
- Showing parameter configurations on the Pareto frontier
- As m increases, the obtained recall increases when considering a fixed search cost, thus achieving a better recall-speed trade-off on both benchmarks. However, as m increases, the index size increase!
- Vec2Doc provides an improved effectiveness-efficiency trade-off also in the Voronoi-partitioning version.

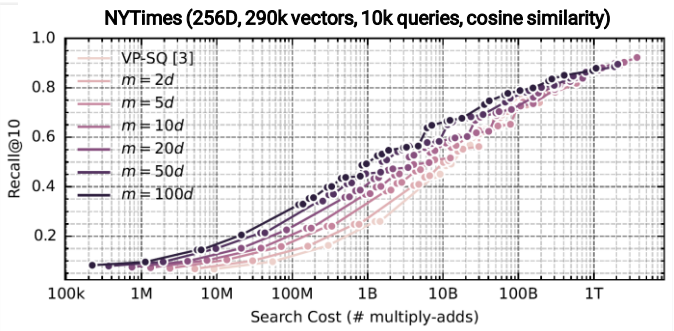
GloVe (100D, 1.18M vectors, 10k queries, cosine similarity)



NYTimes (256D, 290k vectors, 10k queries, cosine similarity)



NYTimes (256D, 290k vectors, 10k queries, cosine similarity)



Conclusions

- We extended the family of Surrogate Text Representations (STR) techniques with a new aapproach for transforming dense real vectors into surrogate texts, suitable to be indexed and searched using off-the-shelf textual search engines.
- Our approach use a semi-orthonormal transformation to allow expanding the codebook size utilized in the encoding, whereas codebooks used by other STR approaches are constrained by the dimensionality of the dense vectors to be searched.
- Improved recall-throughput trade-off on standard (non ad-hoc) textual search engines.

GitHub