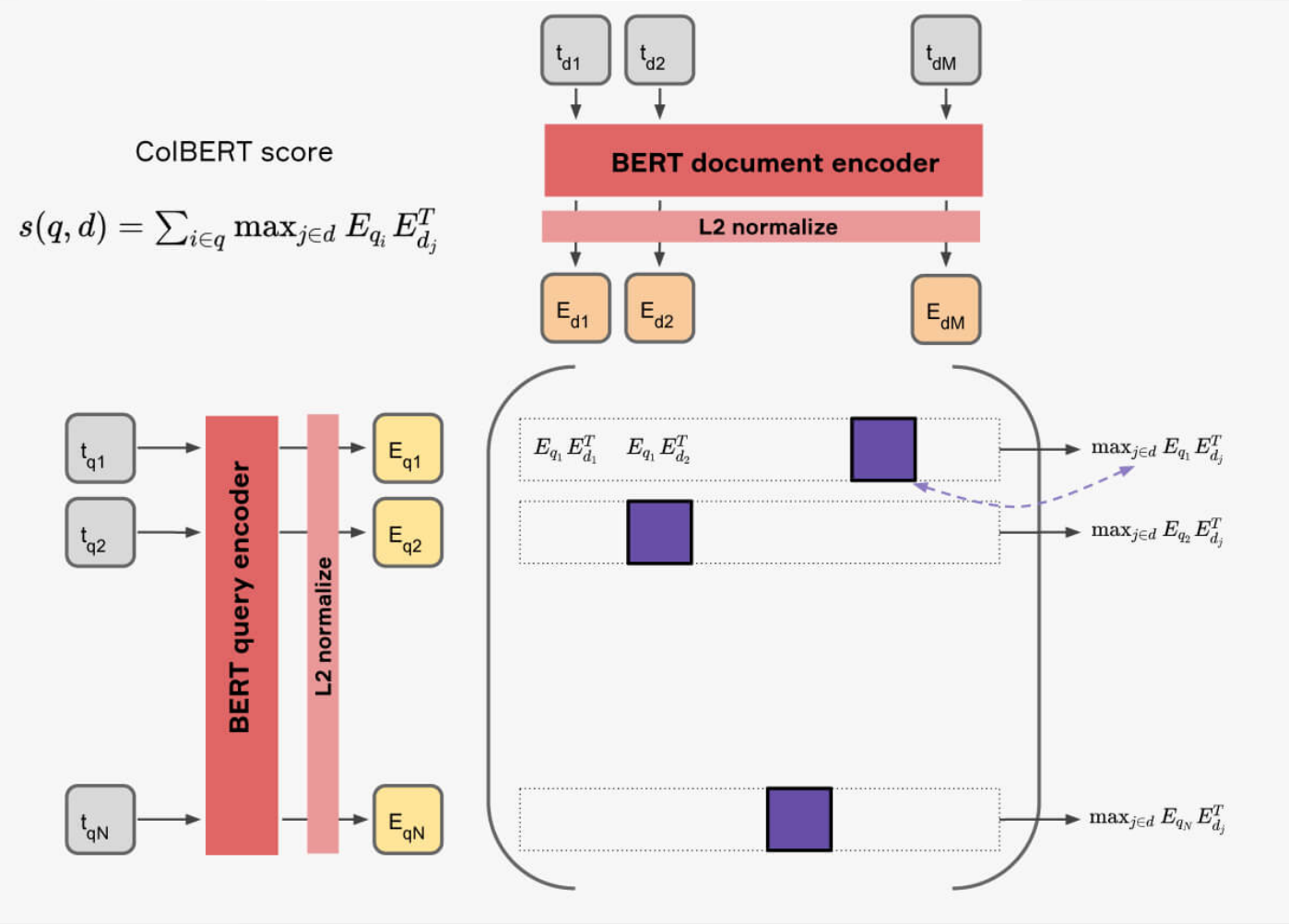
ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction

Thiago Coelho Vieira - thiagotcvieira@gmail.com

1.1 Main Concepts

- **ColBERTv1**
  - dense vector representation for each token
  - "ranking methods based on dense representations can achieve effectiveness that are competitive with a cross-encoder at a fraction of query latency"
  - **exact and soft match** - ColBERT can distinguish terms of which exact match is important
    - for each term check average score in exact and soft cases (how?), if the difference is higher then favors exact match otherwise favors soft match
  - exact match is
- **single-vector** - a pretrained language model is used to encode each query and each document into a single high-dimensional vector, and relevance is modeled as a simple dot product between both vectors
- **multi-vector**
  - for a query/doc q encoder outputs a matrix nxD, not a vector

1.2 ColBERTv1



1.3 Interactions

1.2 MaxSim

- similarity score
- largest cosine similarity between each query token matrix and all passages token matrix.
- “ Since each of these vectors has unit length, the similarity is the sum of maximum cosine similarities between each query term and the “best” matching term contained in the text from the corpus ”

2.1 Contribution

- improvements on ColBERTv1
  - dense vectors compressions (*destillation*) + better negative selection (*hard-negative mining*)
  - ColBERTv1
    - 128dim vectors with 2 bytes = 256 bytes/vector
  - ColBERTv2
    - dimensionality reduction by arranging vectors in clusters indexed by 4 bytes (2^{32}) clusters)
    - improvement that enable 20-36bytes/vector
    - memory improvement ~6-10x (*residual compression*)
- multi-vectors are stored in cluster based on *MaxSim*
- new dataset *LoTTE (Long-Tail Topic-stratified Evaluation)*

2.2 How it works

- **Training**
  - uses same BERT model to encode queries q and docs/texts d
  - prepended queries with special token $[Q]$ and docs/texts with $[D]$
- **Dimensionality Reduction - Product Quantization**
  - high dim vectors splitted in same size smaller vectors
  - each sub-vector is associated with the nearest centroid on vector space
  - replace the values of the centroids by the unique ids
  - outputs a vector of unique ids for each centroid
- **Inverted Index**
  - centroids ids
- **Search**
  - At search time, the query q is encoded into a multi-vector representation and its similarity to a passage d is computed as the summation of query-side *MaxSim* operations.

3. interesting/unexpected results

- in-domain
  - beats DPR and $SPLADEv2$
- gigantic index
  - ColBERTv1 $154GiB$ 🤯
  - ColBERTv2 $16GiB(1bit)$ and $25GiB(2bit)$
- $MMR@10$
  - 1bit 36.2
  - 2bit 35.5
- success@5 metric
- LoTTE dataset

4. basic doubts

- *long-tail topics* - `out of domain topics?`
- on ColBERTv1
  - the vector representation of each token is normalized to a unitary L2 norm; this makes computing inner products equivalent to computing cosine similarity.
  - relevance score is the sum of the max similarity between each vector of the query q and all doc d vectors