An Effective Post-training Embedding Binarization Approach for Fast Online Top-K Passage Matching

Yankai Chen[1], **Yifei Zhang**[1], **Huifeng Guo**[2], **Ruiming Tang**[2] **and Irwin King**[1]
[1]The Chinese University of Hong Kong [2]Huawei Noah's Ark Lab
{ykchen,yfzhang,king}@cse.cuhk.edu.hk, {huifeng.guo,tangruiming}@huawei.com

Abstract

With the rapid development of Natural Language Understanding for information retrieval, fine-tuned deep language models, e.g., BERT-based, perform remarkably effective in passage searching tasks. To lower the architecture complexity, the recent state-of-the-art model ColBERT employs *Contextualized Late Interaction* paradigm to independently learn fine-grained query-passage representations. Apart from the architecture simplification, *embedding binarization*, as another promising branch in model compression, further specializes in the reduction of memory and computation overheads. In this concise paper, we propose an effective post-training embedding binarization approach over ColBERT, achieving both *architecture-level* and *embedding-level* optimization for online inference. The empirical results demonstrate the efficaciousness of our proposed approach, empowering it to perform online query-passage matching acceleration.

1 Introduction

The Information Retrieval community has witnessed an emerging slew of BERT (Devlin et al., 2018)-based deep ranking models that achieves performance superiority in various retrieval benchmarks (Dai and Callan, 2019b; MacAvaney et al., 2019; Nogueira and Cho, 2019; Yilmaz et al., 2019). Despite their advantage in learning deeply-contextualized semantic representations, a major issue however is the heavy computational complexity. A recent model ColBERT (Khattab and Zaharia, 2020) detaches the query-passage contextual encoding in the proposed *Contextualized Late Interaction* mechanism, achieving substantial progress in optimizing the runtime resource footprints.

Orthogonal to architecture simplification, *embedding binarization*, i.e., another model compression technique, has received growing attention across various applications (Lin et al., 2017; Zhang and Zhu, 2019; Qin et al., 2020; Chen et al., 2022a). Despite the promising advantages, it usually suffers from large performance degradation even with adequate training supports (Bai et al., 2021), in which the crux generally lies in:

- **Inevitable semantic erosion.** Compared to the original embeddings, binarized targets are naturally less informative to represent the semantics. Consequently, this leads to a degraded model capability in distinguishing and ranking passages for query-based requests.
- **Inaccurate gradient estimation.** Due to the non-differentiability of binarizer $\mathrm{sign}(\cdot)$, several gradient estimators are proposed (Darabi et al., 2018; Yang et al., 2019; Liu et al., 2019; Qin et al., 2020; Gong et al., 2019). However, these estimators usually are based on *visually similar* simulation to $\mathrm{sign}(\cdot)$, but not necessarily are *theoretically relevant* to it, which may lead to inaccurate gradient estimation in backpropagation.

To tackle these issues, we propose an effective post-training binarization approach by introducing:

1. **Semantic diffusion** technique to "distribute" informative latent semantics to the embedding matrix more uniformly (instead of to the condensed sub-areas) to hedge the binarization information erosion (§ 3.1).
2. **Approximation of Unit Impulse Function** to approximate the derivatives of $\mathrm{sign}(\cdot)$ more rigorously to provide the consistent optimization direction in both forward and backward propagation of the model training workflow (§ 3.2).

Related work & Future directions. There exist several other methods to close the performance disparity, such as *knowledge distillation* (Hinton et al., 2015; Anil et al., 2018), *multi-bit quantization* (Li et al., 2016), and *various augmentation strategies* (Ning et al., 2020; Jang and Cho, 2021). In this paper, we base on ColBERT (2020) to evaluate the proposed post-training binarization approach, and will study its generalization to other appropriate deep language models as future work.

Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 102–108
November 20–23, 2022.

2 Preliminaries

ColBERT (Khattab and Zaharia, 2020). It comprises: (1) a query encoder f_Q, (b) a passage encoder f_D, and (3) a query-passage score predictor. Specifically, given a query q and a passage d, f_Q and f_D encode them into a bag of fixed-size embeddings $\boldsymbol{E}_q$ and $\boldsymbol{E}_d$ as follows:

$$\boldsymbol{E}_q := \texttt{Normalize}(\texttt{CNN}(\texttt{BERT}(\text{“}[Q]q_0q_1\cdots q_l\#\#\cdots\#\text{”}))),$$
$$\boldsymbol{E}_d := \texttt{Filter}(\texttt{Normalize}(\texttt{CNN}(\texttt{BERT}(\text{“}[D]d_0d_1\cdots d_n\text{”})))), \tag{1}$$

where q and d are tokenized into tokens $q_0q_1\cdots q_l$ and $d_0d_1\cdots d_n$ by BERT-based WordPiece (Wu et al., 2016), respectively. $[Q]$ and $[D]$ indicate the sequence types and $\#$ denotes the special padding token when a query has fewer tokens than a predefined token number.

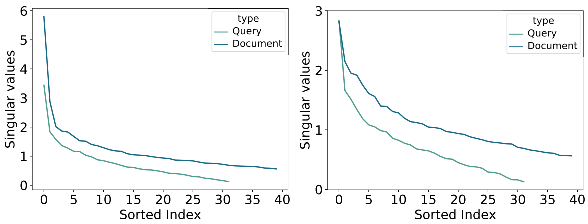
Embedding Binarization and Optimization. The conventional methods (Gersho and Gray, 2012; Courbariaux et al., 2016; Lin et al., 2017; Chen et al., 2021) generally adopt $\mathrm{sign}(\cdot)$ function for binarization mainly because of its $O(1)$ simplicity. However, as $\mathrm{sign}(\cdot)$ is non-differentiable, previous *visually similar* gradient estimators (2018; 2019; 2019; 2020; 2019) are not necessarily *theoretically relevant* to $\mathrm{sign}(\cdot)$. For example, estimator $1-\tanh^2(\cdot)$ provides executable gradient estimation, which however is the factual derivative of $\tanh(\cdot)$ (Qin et al., 2020; Gong et al., 2019). This may distract the main direction of the factual gradient for model optimization in forward and backward propagation, which thus leads to performance degradation of downstream tasks.

3 Bi-ColBERT Methodology

To tackle the aforementioned issue, we propose Bi-ColBERT by introducing two effective and lightweight techniques: (1) semantic diffusion to hedge the information loss against embedding binarization, and (2) approximation of Unit Impulse Function (Dirac, 1927; Bracewell and Bracewell, 1986) for more accurate gradient estimation.

3.1 Semantic Diffusion

Binarization with $\mathrm{sign}(\cdot)$ inevitably smoothes the embedding informativeness into the binarized space, e.g., $\{$-1,1$\}^d$ regardless of its original values. Thus, intuitively, we want to avoid condensing and gathering informative latent semantics in (relatively-small) sub-structures of embedding bags, e.g., $\boldsymbol{E}_q$; in other words, we seek to *diffuse the embedded semantics in all embedding dimensions as one effective strategy* to hedge the inevitable information loss caused by the numerical binarization and *retain the semantic uniqueness after binarization as much as possible*.



(a) No semantic diffusion. (b) With semantic diffusion.

Figure 1: Singular value distribution example (sorted in descending order): using semantic diffusion on MS MARCO dataset can well balance the matrix spectrum.

Recall in singular value decomposition (SVD), singular values and vectors reconstruct the original matrix; normally, large singular values can be interpreted to associate with major semantic structures of the matrix (Wei et al., 2018). Hence, based on this observation, we can achieve semantic diffusion via normalizing singular values for equalizing their respective contributions in constituting latent semantics. To achieve this, Power Normalization (Li et al., 2017; Koniusz et al., 2016) is one of the solutions that tackle related problems such as *feature imbalance* in image processing (Koniusz et al., 2018; Quattoni and Torralba, 2009). Inspired by the recent approximation attempt (Yu et al., 2020), we introduce a lightweight semantic diffusion technique as follows.

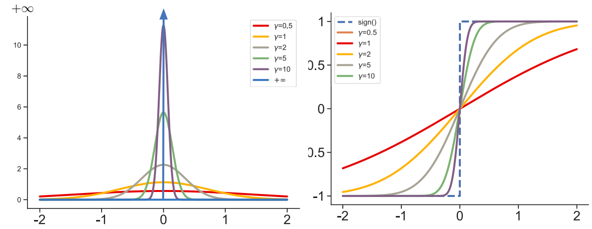
Concretely, let $\boldsymbol{I}$ denote the identity matrix, we start from generating a *standard normal random vector* $\boldsymbol{p}^{(0)}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})$ where $\boldsymbol{p}^{(0)}\in\mathbb{R}^d$. Based on the embedding matrix for semantic diffusion, e.g., $\boldsymbol{E_q}$, we compute the **diffusion vector** $\boldsymbol{p}^{(h)}$ by iteratively performing $\boldsymbol{p}^{(h)} = \boldsymbol{E}_q^{\top}\boldsymbol{E}_q\boldsymbol{p}^{(h-1)}$. Next we can obtain the projection matrix $\boldsymbol{P}_q$ of $\boldsymbol{p}$ via:

$$\boldsymbol{P}_q = \frac{\boldsymbol{p}^{(h)}\boldsymbol{p}^{(h)\top}}{||\boldsymbol{p}^{(h)}||_2^2}. \tag{2}$$

Then we have the **semantic-diffused** embedding bag with the hyper-parameter $\epsilon\in(0,1)$ as:

$$\widehat{\boldsymbol{E}}_q = \boldsymbol{E}_q(\boldsymbol{I}-\epsilon\boldsymbol{P}_q). \tag{3}$$

We conduct similar operations to passage embedding bags, e.g., $\boldsymbol{E}_d$, for semantic diffusion. Compare to the unprocessed embedding bag, i.e., $\boldsymbol{E}_q$, embedding $\widehat{\boldsymbol{E}}_q$ presents a diffused semantic structure with a *more balanced spectrum (distribution of singular values) in expectation*. We theoretically explain this by Theorem 1 in Appendix A and illustrate a visual comparison in Figure 1.



(a) Approximation to Unit Impulse Function. (b) Cumulative distribution.

Figure 2: Proposed gradient estimation illustration.

3.2 Gradient Estimation

Rescaled Binarization. After obtaining the semantic-diffused embedding bag, e.g., $\widehat{\boldsymbol{E}}_q$, we conduct the *rescaled embedding binarization* for each one embedding of the contextualized bag as:

$$\boldsymbol{B}_{q_i} := \omega_{q_i} \cdot \operatorname{sign}(\widehat{\boldsymbol{E}}_{q_i}), \text{ where } \omega_{q_i} = \frac{||\widehat{\boldsymbol{E}}_{q_i}||_1}{c}. \tag{4}$$

Here $i \in [|\widehat{\boldsymbol{E}}_q|]$ and c denotes the embedding dimension. The binarized embedding bag $\boldsymbol{B}_q$ sketches the original embeddings via (1) binarized codes (i.e., $\{-1, 1\}^c$) and (2) embedding scaler (i.e., $\omega_{q_i} \in \mathbb{R}^+$), both of which collaboratively reveal the value range of original embedding entries. Moreover, such rescaled binarization supports the bit-wise operations for computation acceleration in match-scoring prediction, which will be introduced later.

Approximation of *Unit Impulse Function*. Although previous gradient estimators are *visually similar* (e.g., $\tanh(\cdot)$) (Gong et al., 2019; Qin et al., 2020) to provide an executable gradient flow, it however may lead to the inconsistent optimization direction in forward and backward propagation. This is because, the integral of the approximation function (e.g., derivatives of $\tanh(\cdot)$) may not be consistent with $\operatorname{sign}(\cdot)$. To tackle this issue and furnish the accordant gradient estimation, we utilize the approximation of *Unit Impulse Function* (Dirac, 1927; Bracewell and Bracewell, 1986) as follows.

It has been proved that *Unit Impulse Function* defined in the right-hand side of Equation (5) is the derivatives of *Unit Step function* $u(t)$[1], where $u(t)$ $= 0$ for $t \leq 0$ and $u(t) = 1$ otherwise.

$$\frac{\partial u(t)}{\partial t} = \begin{cases} 0 & t \neq 0 \\ \infty & t = 0. \end{cases} \tag{5}$$

It is obvious to take a translation by $\operatorname{sign}(t) = 2u(t)$ - 1, and theoretically $\frac{\partial \operatorname{sign}(t)}{\partial t} = 2\frac{\partial u(t)}{\partial t}$. Furthermore, $\frac{\partial u(t)}{\partial t}$ can be introduced with zero-centered Gaussian probability density function as:

$$\frac{\partial u(t)}{\partial t} = \lim_{\beta \to \infty} \frac{|\beta|}{\sqrt{\pi}} \exp(-(\beta t)^2), \tag{6}$$

[1] https://en.wikipedia.org/wiki/Heaviside_step_function

which implies that:

$$\frac{\partial \operatorname{sign}(t)}{\partial t} \approx \frac{2\gamma}{\sqrt{\pi}} \exp(-(\gamma t)^2). \tag{7}$$

As shown in Figure 2, hyper-parameter $\gamma \in \mathbb{R}^+$ determines the curve sharpness to approximate $\operatorname{sign}(\cdot)$. Intuitively, this estimator in Equation (7) follows the main direction of factual gradients of $\operatorname{sign}(\cdot)$, which produces a coordinated embedding optimization for inputs with diverse value ranges. Its performance superiority over other recent estimators is demonstrated in experiments later.

3.3 Online Query-passage Matching

Similarly to ColBERT (Khattab and Zaharia, 2020), we employ its proposed *Late Interaction Mechanism* for matching score computation, which is implemented by a sum of maximum similarity computation with embedding dot-products:

$$S_{q,d} := \sum_{i \in [|\boldsymbol{B}_q|]} \max_{j \in [|\boldsymbol{B}_d|]} \boldsymbol{B}_{q_i} \cdot \boldsymbol{B}_{d_j}^{\mathsf{T}}, \tag{8}$$

Which can be equivalently implemented with bit-wise operations as follows:

$$S_{q,d} := \sum_{i \in [|\boldsymbol{B}_q|]} \max_{j \in [|\boldsymbol{B}_d|]} \omega_{q_i}\omega_{d_j} \cdot \texttt{count}\big(\texttt{xnor}\big(\operatorname{sign}(\boldsymbol{B}_{q_i}) \cdot \operatorname{sign}(\boldsymbol{B}_{d_i}^{\mathsf{T}})\big)\big), \tag{9}$$

Equation (9) replaces most of floating-point arithmetics with bit-wise operations, providing the potentiality of online computation acceleration. We plan to develop hardware-adapted computation operators (e.g., *"bit-wise tensors"*) in future. Lastly, Bi-ColBERT adopts the training paradigm of ColBERT (2020) that is optimized via the pairwise softmax cross-entropy loss over the computed scores of positive and negative passage samples.

4 Experimental Evaluation

We now evaluate our approach with the aim of answering the following research questions:

- **RQ1.** How does Bi-ColBERT perform in the fine-grained Top-K passage searching task?
- **RQ2.** Is the proposed semantic diffusion technique effective to hedge the information loss?
- **RQ3.** How does the proposed gradient estimator compare to the previous counterparts?

We implement our embedding binarization approach directly on pretrained ColBERT, denoted as $\text{ColBERT}_{\text{pretrain}}$. To give a fair comparison, we use the same dataset (i.e., MS MARCO) and evaluation metric (i.e., MRR@10) with ColBERT. Detailed experimental setups and baseline introduction are attached in Appendix B.

Table 1: Top-1000 Reranking results on MS MARCO.

Model	MRR@10
$BM25_{official}$ (Robertson et al., 1995)	16.7
KNRM (Xiong et al., 2017; Dai et al., 2018)	19.8
Duet (Mitra et al., 2017)	24.3
FT+ConvKNRM (Hofstätter et al., 2019)	29.0
$BERT_{base}$ (Nogueira and Cho, 2019)	34.7
$BERT_{large}$ (Nogueira and Cho, 2019)	36.5
$ColBERT_{official}$ (Khattab and Zaharia, 2020)	34.9
$ColBERT_{pretrain}$	32.8
Bi-ColBERT ($r_s = 15.1\times, r_t = 7.3\times$)	31.7

4.1 Overall Performance (RQ1)

Similar to ColBERT (2020), we evaluate the fine-grained searching capability via the official Top-1000 reranking on MS MARCO *w.r.t.* MRR@10. From Table 1, we have the following observations:

(1) Bi-ColBERT works better than prior non-BERT-based models, owing to the power of *fine-tuned* BERT-based methods in learning deep contextualized semantic representations.

(2) Furthermore, ColBERT and Bi-ColBERT make the tradeoff between passage searching quality and retrieval cost, where ColBERT aims to simplify the neural architecture and our proposed methods focus on effective embedding binarization. We use r_s and r_t to denote the ratios of Bi-ColBERT over ColBERT *w.r.t.* embedding size compression and online score computation acceleration on CPUs (details are in Appendix B). Considering the advantages in memory reduction and inference acceleration, i.e., r_s=15.1×, r_t=7.3×, Bi-ColBERT provides an alternative option for ColBERT, especially in resource-limited scenarios.

(3) Despite the performance gap between ColBERT and our approach, we argue that it is mainly caused by the inevitable information loss in numerical binarization, which is unfortunately common in prior work (Lin et al., 2017; Darabi et al., 2018; Gong et al., 2019; Qin et al., 2020). To narrow the gap, as briefly introduced in § 1, several independent yet advanced methods can be further studied and deployed for model improvement. We provide a detailed discussion later in § 5.

4.2 Analysis of Semantic Diffusion (RQ2)

In this section, we study the effectiveness of our proposed semantic diffusion (SD) by setting two groups of ablation experiments. From Table 2(A),

(1) We first disable the embedding binarization (EB) and check the effect of SD on our model. Results show that simply using SD will not *negatively* affect the holistic model performance. This validates our analysis in Appendix A that SD aims to balance the spectrum of embedding matrix (e.g.,

Table 2: (A) Ablation study of Semantic Diffusion. (B) Gradient estimator comparison.

Components	Results
SD (✗) + EB (✗)	32.8
SD (✓) + EB (✗)	32.9
SD (✗) + EB (✓)	30.3
SD (✓) + EB (✓)	31.7

Estimator	Results
STE	29.7
PBE	30.4
Sigmoid	30.8
SignSwish	31.1
Tanh	31.2
Bi-ColBERT	31.7

$\boldsymbol{E}_b$) with *its associated orthonormal bases for matrix reconstruction intact*.

(2) In the second experiment group, we trigger EB and the results demonstrate that SD together with our proposed gradient estimation can effectively approach our target to hedge the information loss for representation binarization.

4.3 Gradient Estimator Comparison (RQ3)

Lastly, the experimental results in Table 2(B) show the consistent performance superiority of our proposed gradient estimator over all prior counterparts. This generally follows our observation explained in § 2. On the contrary, our approach to approximate Unit Impulse Function follows the main optimization direction of factual gradients with sign(·); and different from previous solutions, this guarantees the coordination in both forward and backward propagation of model optimization.

5 Discussion for Future Work

We summarize five promising future directions.

1. It is pragmatic to evaluate the adaptability of our approach to other BERT-based models.
2. A promising direction could be using embedding binarization for other scenarios with efficiency demands (Zhang and Zhu, 2020; Chen et al., 2022b; Zhang et al., 2022; Chen et al., 2022c; Yang et al., 2021).
3. ColBERT also employs `faiss` (Johnson et al., 2019), a tool for large-scale vector-similarity search. Thus, it is worth developing a similar index-based data structure specifically for retrieval in the discrete embedding space.
4. Data augmentation, e.g., *feature-based augmentation* (Wang et al., 2019), is another effective technique to boost embedding informativeness before and after the binarization.
5. If the training resource is adequate, quantization-aware training (Zafrir et al., 2019) resembles the standard fine-tuning and thus is promising to compensate for the performance degradation.

A Semantic Diffusion Analysis

Theorem 1 (Semantic Diffusion)**.** *For each pair of unprocessed and processed embedding bags, i.e., ($\widehat{\boldsymbol{E}}$, $\boldsymbol{E}$), $\boldsymbol{E} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathsf{T}}$, where $\boldsymbol{U}$ and $\boldsymbol{V}$ are unitary matrices and descending singular value matrix $\boldsymbol{\Sigma}$ $= \mathrm{diag}(\sigma_1, \sigma_2, \cdots, \sigma_d)$. Then $\mathbb{E}(\widehat{\boldsymbol{E}}) = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{\Sigma}_{\mu}\boldsymbol{V}^{\mathsf{T}}$ where $\boldsymbol{\Sigma}_{\mu} = \mathrm{diag}(\mu_1, \mu_2, \cdots, \mu_d)_{0<\mu_{1\cdots d}<1}$ is in the ascending order.*

Proof. Conducting SVD decomposition on $\boldsymbol{E}$, we have $\boldsymbol{E} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathsf{T}}$, where $\boldsymbol{U}$ and $\boldsymbol{V}$ are unitary matrices of singular vectors. Then following $\boldsymbol{p}^{(h)} = \boldsymbol{E}^{\mathsf{T}}\boldsymbol{E}\boldsymbol{p}^{(h-1)}$, we shall have $\boldsymbol{p}^{(h)} = (\boldsymbol{E}^{\mathsf{T}}\boldsymbol{E})^h\boldsymbol{p}^{(0)}$. Replacing $\boldsymbol{E}$ with its SVD decomposition, we get the following equation:

$$\boldsymbol{p}^{(h)} = (\boldsymbol{V}\boldsymbol{\Sigma}^{2h}\boldsymbol{V}^{\mathsf{T}})\boldsymbol{p}^{(0)}. \tag{10}$$

Then we transform the projection matrix computed in Equation (2) as follows:

$$\begin{aligned}\boldsymbol{P} &= \frac{\boldsymbol{p}^{(h)}\boldsymbol{p}^{(h)\mathsf{T}}}{\boldsymbol{p}^{(h)\mathsf{T}}\boldsymbol{p}^{(h)}} = \frac{(\boldsymbol{V}\boldsymbol{\Sigma}^{2h}\boldsymbol{V}^{\mathsf{T}})\boldsymbol{p}^{(0)}\boldsymbol{p}^{(0)\mathsf{T}}(\boldsymbol{V}\boldsymbol{\Sigma}^{2h}\boldsymbol{V}^{\mathsf{T}})}{\boldsymbol{p}^{(0)\mathsf{T}}(\boldsymbol{V}\boldsymbol{\Sigma}^{2h}\boldsymbol{V}^{\mathsf{T}})(\boldsymbol{V}\boldsymbol{\Sigma}^{2h}\boldsymbol{V}^{\mathsf{T}})\boldsymbol{p}^{(0)}} \\ &= \boldsymbol{V}\boldsymbol{\Sigma}^{2h}\frac{\boldsymbol{V}^{\mathsf{T}}\boldsymbol{p}^{(0)}\boldsymbol{p}^{(0)\mathsf{T}}\boldsymbol{V}}{\boldsymbol{p}^{(0)\mathsf{T}}\boldsymbol{V}\boldsymbol{\Sigma}^{4h}\boldsymbol{V}^{\mathsf{T}}\boldsymbol{p}^{(0)}}\boldsymbol{\Sigma}^{2h}\boldsymbol{V}^{\mathsf{T}}.\end{aligned} \tag{11}$$

Let $\boldsymbol{t} = \boldsymbol{V}^{\mathsf{T}}\boldsymbol{p}^{(0)}$, we can further simplify the above equation to:

$$\boldsymbol{P} = \boldsymbol{V}\boldsymbol{\Sigma}^{2h}\frac{\boldsymbol{t}\boldsymbol{t}^{\mathsf{T}}}{\boldsymbol{t}^{\mathsf{T}}\boldsymbol{\Sigma}^{4h}\boldsymbol{t}}\boldsymbol{\Sigma}^{2h}\boldsymbol{V}^{\mathsf{T}}, \tag{12}$$

where scalar $\boldsymbol{t}^{\mathsf{T}}\boldsymbol{\Sigma}^{4h}\boldsymbol{t}$ is defined as:

$$\boldsymbol{t}^{\mathsf{T}}\boldsymbol{\Sigma}^{4h}\boldsymbol{t} = \textstyle\sum_{j=1}^{d} t_j^2\sigma_j^{4h}. \tag{13}$$

Recalling that $\widehat{\boldsymbol{E}} = \boldsymbol{E}(\boldsymbol{I} - \epsilon\boldsymbol{P})$, $\mathbb{E}(\widehat{\boldsymbol{E}}) = \boldsymbol{E} - \epsilon \cdot \mathbb{E}(\boldsymbol{E}\boldsymbol{P})$. Then we focus on the term $\mathbb{E}(\boldsymbol{E}\boldsymbol{P})$:

$$\mathbb{E}(\boldsymbol{E}\boldsymbol{P}) = \tfrac{1}{\boldsymbol{t}^{\mathsf{T}}\boldsymbol{\Sigma}^{4h}\boldsymbol{t}}\boldsymbol{U}\boldsymbol{\Sigma}^{2h+1}\cdot\mathbb{E}(\boldsymbol{t}\boldsymbol{t}^{\mathsf{T}})\cdot\boldsymbol{\Sigma}^{2h}\boldsymbol{V}^{\mathsf{T}}. \tag{14}$$

Since $\boldsymbol{p}^{(0)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and $\boldsymbol{V}$ is a unitary matrix, thus $\boldsymbol{t} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. This indicates that each element of $\boldsymbol{t}$, e.g., $t_j \in \boldsymbol{t}$, is *i.i.d.* random variable. Thus, $\mathbb{E}(t_j \cdot t_k) = 0$ for $j \neq k$ and $\mathbb{E}(\boldsymbol{t}\boldsymbol{t}^{\mathsf{T}})$ is a diagonal matrix, i.e., $\mathbb{E}(\boldsymbol{t}\boldsymbol{t}^{\mathsf{T}}) = \mathrm{diag}(t_1^2, t_2^2, \cdots, t_d^2)$. We then have:

$$\mathbb{E}(\boldsymbol{E}\boldsymbol{P}) = \boldsymbol{U}\cdot\mathrm{diag}\,(\frac{\sigma_1 t_1^2\sigma_1^{4h}}{\sum_{j=1}^{d} t_j^2\sigma_j^{4h}}, \cdots, \frac{\sigma_d t_d^2\sigma_d^{4h}}{\sum_{j=1}^{d} t_j^2\sigma_j^{4h}})\cdot\boldsymbol{V}^{\mathsf{T}}. \tag{15}$$

Therefore,

$$\mathbb{E}(\widehat{\boldsymbol{E}}) = \boldsymbol{U}\cdot\mathrm{diag}\,(\sigma_1 - \epsilon\frac{\sigma_1 t_1^2\sigma_1^{4h}}{\sum_{j=1}^{d} t_j^2\sigma_j^{4h}}, \cdots, \sigma_d - \epsilon\frac{\sigma_d t_d^2\sigma_d^{4h}}{\sum_{j=1}^{d} t_j^2\sigma_j^{4h}})\cdot\boldsymbol{V}^{\mathsf{T}}. \tag{16}$$

Let $\mu_k = 1 - \epsilon\frac{t_k^2\sigma_k^{4h}}{\sum_{j=1}^{d} t_j^2\sigma_j^{4h}}$, with $\epsilon \in (0, 1)$, obviously, $0 < \mu_k < 1$. Furthermore, $\forall k_1 \geq k_2$, we have:

$$\begin{aligned}\mu_{k_1} - \mu_{k_2} &= \epsilon\mathbb{E}(\frac{t_{k_1}^2\sigma_{k_1}^{4h}}{\sum_{j=1}^{d} t_j^2\sigma_j^{4h}} - \frac{t_{k_2}^2\sigma_{k_2}^{4h}}{\sum_{j=1}^{d} t_j^2\sigma_j^{4h}}) \\ &\geq \epsilon\sigma_{k_1}^{4h}\cdot\mathbb{E}(\frac{t_{k_1}^2 - t_{k_2}^2}{\sum_{j=1}^{d} t_j^2\sigma_j^{4h}}) = 0,\end{aligned} \tag{17}$$

as $\sigma_{k_2}^{4h} \geq \sigma_{k_1}^{4h}$, and t_{k_1} and t_{k_2} are *i.i.d.* random variables with same normal distribution. Equation (17) proves that μ_k is *monotone non-decreasing* in expectation, which completes the proof. □

Intuitively, given the same orthonormal bases, compared to unprocessed embedding bag $\boldsymbol{E}$, it is harder in expection to reconstruct $\widehat{\boldsymbol{E}}$ with informative semantics being diffused out in larger matrix sub-structures, which however hedges the information loss in numerical binarization.

B Experiment Setup

Dataset and Metric. Similar to work (2019a; 2019a; 2019b; 2020), we evaluate our model on the MS-MARCO Ranking (2016) dataset. It is a collection of 8.8M passages from 1M real-world queries to Bing. Each query is associated with sparse relevance judgments of one (or a small number of) documents marked as relevant and no documents explicitly marked as irrelevant. Similar to ColBERT (2020), we use metric MRR@10 for performance evaluation.

Baselines. We include baselines for comparison from prior (1) learn-to-rank models, i.e., BM25 (offical) (1995), KNRM (2018; 2017), Duet (2017), FastText+ConvKNRM (2019) (denoted as FT-ConvKNRM), and (2) BERT-based models, i.e., $\text{BERT}_{\text{base}}$ (2019), $\text{BERT}_{\text{large}}$ (2019) and ColBERT (2020). We use subscripts, i.e., $_{\text{official}}$, $_{\text{base}}$ and $_{\text{large}}$, to denote respective refered versions. $\text{ColBERT}_{\text{pretrain}}$ denotes the pretrained version.

Implementations. Our model is implemented under Python 3.7 and PyTorch 1.6.0. We initialize our model by using the pretrained ColBERT model under its reported default settings, i.e., $\text{ColBERT}_{\text{pretrain}}$. Then we fine-tune our proposed model with: the same learning rate - 3×10^{-6}, the batch size - 32, and embedding dimension - 128, iteration number for diffusing vector computation h - 2, and hyper-parameter $\gamma = 0.5$. For other evaluation settings, we directly follow ColBERT (2020). We train our model in a Linux machine with 4 GPUs, each of which is a NVIDIA V100 GPU, 4 Intel Core i7-8700 CPUs, 32 GB of RAM with 3.20GHz. For Top-K reranking tasks, we use CPUs per query for the passage retrieval. To evaluate the embedding compression ratio r_s, we measure the size of embeddings produced by Bi-ColBERT and ColBERT per query. For embeddings from ColBERT, we use float32 as the default. Then to measure online score computation time cost ratio r_t, based on the computed embeddings, we conduct experiments on CPUs with the vanilla NumPy (2022) implementation.

References

Rohan Anil, Gabriel Pereyra, Alexandre Passos, Robert Ormandi, George E Dahl, and Geoffrey E Hinton. 2018. Large scale distributed neural network training through online distillation.

Haoli Bai, Wei Zhang, Lu Hou, Lifeng Shang, Jing Jin, Xin Jiang, Qun Liu, Michael Lyu, and Irwin King. 2021. Pushing the limit of bert quantization.

Ronald Newbold Bracewell and Ronald N Bracewell. 1986. *The Fourier transform and its applications*, volume 31999. McGraw-Hill New York.

Yankai Chen, Huifeng Guo, Yingxue Zhang, Chen Ma, Ruiming Tang, Jingjie Li, and Irwin King. 2022a. Learning binarized graph representations with multi-faceted quantization reinforcement for top-k recommendation. In *SIGKDD*, pages 168–178. ACM.

Yankai Chen, Menglin Yang, Yingxue Zhang, Mengchen Zhao, Ziqiao Meng, Jianye Hao, and Irwin King. 2022b. Modeling scale-free graphs with hyperbolic geometry for knowledge-aware recommendation. In *WSDM*, pages 94–102.

Yankai Chen, Yaming Yang, Yujing Wang, Jing Bai, Xiangchen Song, and Irwin King. 2022c. Attentive knowledge-aware graph convolutional networks with collaborative guidance for personalized recommendation. In *ICDE*, pages 299–311.

Yankai Chen, Yifei Zhang, Yingxue Zhang, Huifeng Guo, Jingjie Li, Ruiming Tang, Xiuqiang He, and Irwin King. 2021. Towards low-loss 1-bit quantization of user-item representations for top-k recommendation. *arXiv preprint arXiv:2112.01944*.

Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2016. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv*.

Zhuyun Dai and Jamie Callan. 2019a. Context-aware sentence/passage term importance estimation for first stage retrieval. *arXiv preprint arXiv:1910.10687*.

Zhuyun Dai and Jamie Callan. 2019b. Deeper text understanding for ir with contextual neural language modeling. In *SIGIR*, pages 985–988.

Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *WSDM*, pages 126–134.

Sajad Darabi, Mouloud Belbahri, Matthieu Courbariaux, and Vahid Partovi Nia. 2018. Bnn+: Improved binary network training.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Paul Adrien Maurice Dirac. 1927. The physical interpretation of the quantum dynamics. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 113(765):621–641.

Allen Gersho and Robert M Gray. 2012. *Vector quantization and signal compression*, volume 159. Springer Science & Business Media.

Christoph Gohlke. 2022. https://www.lfd.uci.edu/~gohlke/pythonlibs/.

Ruihao Gong, Xianglong Liu, Shenghu Jiang, Tianxiang Li, Peng Hu, Jiazhen Lin, Fengwei Yu, and Junjie Yan. 2019. Differentiable soft quantization: Bridging full-precision and low-bit neural networks. In *ICCV*, pages 4852–4861.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Sebastian Hofstätter, Navid Rekabsaz, Carsten Eickhoff, and Allan Hanbury. 2019. On the effect of low-frequency terms on neural-ir models. In *SIGIR*, pages 1137–1140.

Young Kyun Jang and Nam Ik Cho. 2021. Self-supervised product quantization for deep unsupervised image retrieval. In *ICCV*, pages 12085–12094.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *SIGIR*, pages 39–48.

Piotr Koniusz, Fei Yan, Philippe-Henri Gosselin, and Krystian Mikolajczyk. 2016. Higher-order occurrence pooling for bags-of-words: Visual concept detection. *TPAMI*, 39(2):313–326.

Piotr Koniusz, Hongguang Zhang, and Fatih Porikli. 2018. A deeper look at power normalizations. In *CVPR*, pages 5774–5783.

Fengfu Li, Bo Zhang, and Bin Liu. 2016. Ternary weight networks. *arXiv preprint arXiv:1605.04711*.

Peihua Li, Jiangtao Xie, Qilong Wang, and Wangmeng Zuo. 2017. Is second-order information helpful for large-scale visual recognition? In *ICCV*, pages 2070–2078.

Xiaofan Lin, Cong Zhao, and Wei Pan. 2017. Towards accurate binary convolutional neural network.

Chunlei Liu, Wenrui Ding, Xin Xia, Yuan Hu, Baochang Zhang, Jianzhuang Liu, Bohan Zhuang, and Guodong Guo. 2019. Rbcn: Rectified binary convolutional networks for enhancing the performance of 1-bit dcnns. *arXiv*.

Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. Cedr: Contextualized embeddings for document ranking. In *SIGIR*, pages 1101–1104.

Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to match using local and distributed representations of text for web search. In *WWW*, pages 1291–1299.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.

Lin Ning, Guoyang Chen, Weifeng Zhang, and Xipeng Shen. 2020. Simple augmentation goes a long way: Adrl for dnn quantization. In *ICLR*.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019a. Multi-stage document ranking with bert. *arXiv preprint arXiv:1910.14424*.

Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019b. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*.

Haotong Qin, Ruihao Gong, Xianglong Liu, Mingzhu Shen, Ziran Wei, Fengwei Yu, and Jingkuan Song. 2020. Forward and backward information retention for accurate binary neural networks. In *CVPR*, pages 2250–2259.

Ariadna Quattoni and Antonio Torralba. 2009. Recognizing indoor scenes. In *CVPR*, pages 413–420. IEEE.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at trec-3. *Nist Special Publication Sp*, 109:109.

Yulin Wang, Xuran Pan, Shiji Song, Hong Zhang, Gao Huang, and Cheng Wu. 2019. Implicit semantic data augmentation for deep networks. *NeurIPS*, 32.

Xing Wei, Yue Zhang, Yihong Gong, Jiawei Zhang, and Nanning Zheng. 2018. Grassmann pooling as compact homogeneous bilinear pooling for fine-grained visual classification. In *ECCV*, pages 355–370.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end neural ad-hoc ranking with kernel pooling. In *SIGIR*, pages 55–64.

Jiwei Yang, Xu Shen, Jun Xing, Xinmei Tian, Houqiang Li, Bing Deng, Jianqiang Huang, and Xian-sheng Hua. 2019. Quantization networks. In *CVPR*, pages 7308–7316.

Menglin Yang, Min Zhou, Marcus Kalander, Zengfeng Huang, and Irwin King. 2021. Discrete-time temporal network embedding via implicit hierarchical learning in hyperbolic space. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1975–1985.

Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Cross-domain modeling of sentence-level evidence for document retrieval. In *EMNLP-IJCNLP*, pages 3490–3496.

Tan Yu, Yunfeng Cai, and Ping Li. 2020. Toward faster and simpler matrix normalization via rank-1 update. In *ECCV*, pages 203–219. Springer.

Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8bert: Quantized 8bit bert. In *EMC2-NIPS*, pages 36–39. IEEE.

Xinni Zhang, Yankai Chen, Cuiyun Gao, Qing Liao, Shenglin Zhao, and Irwin King. 2022. Knowledge-aware neural networks with personalized feature referencing for cold-start recommendation. *arXiv preprint arXiv:2209.13973*.

Yifei Zhang and Hao Zhu. 2019. Doc2hash: Learning discrete latent variables for documents retrieval. In *NAACL-HLT*, pages 2235–2240.

Yifei Zhang and Hao Zhu. 2020. Discrete wasserstein autoencoders for document retrieval. In *ICASSP*, pages 8159–8163. IEEE.