Quantization Aware Matryoshka Adaptation: Leveraging Matryoshka Learning, Quantization, and Bitwise Operations for Reduced Storage and Improved Retrieval Speed

We introduce *Quantization Aware Matryoshka Adaptation (QAMA)*, a unified framework for creating compact yet semantically rich embeddings through **Matryoshka Representation Learning** and multi-level quantization. Our approach learns nested embeddings that gracefully shrink to smaller dimensional subsets and leverages bitwise operations (XOR, NOT, POPCOUNT) for efficient retrieval. By augmenting transformer-based encoders with lightweight feedforward layers and specialized regularization (Matryoshka Loss, Orthogonality, Information Bottleneck, and Quantization Loss), we produce quantization-friendly representations that preserve essential information in early dimensions.

We explore 0.5-bit, 1-bit, 1.5-bit, and 2-bit quantization levels, as well as a Hybrid Quantization scheme that adaptively allocates higher precision to dimensions with greater information content. Across extensive evaluations on Modern BERT (MB) and MiniLM models, **2-bit quantization** and **Hybrid Quant** consistently recover up to 95–98% of the original full-precision (FP32) performance while reducing memory usage by over 90%. Even at low embedding dimensions (e.g., 96–192), QAMA's hierarchical training ensures that performance remains surprisingly robust, highlighting the effectiveness of our bit-level expansions and nested representation learning.

Furthermore, our experiments show that end-to-end training with quantization-aware losses yields embeddings that map cleanly into discrete levels, supporting rapid Hamming distance calculations using bitwise operations for large-scale retrieval. Ablation studies reveal that our suggested component losses contributes to preserving semantic fidelity under aggressive compression. Our framework **QAMA** offers a practical means to store, retrieve, and compute similarities between embeddings efficiently, enabling high-accuracy search with faster retrieval speeds under stringent memory and latency constraints.

CCS Concepts: • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Additional Key Words and Phrases: datasets, neural networks, gaze detection, text tagging

ACM Reference Format:
. 2018. Quantization Aware Matryoshka Adaptation: Leveraging Matryoshka Learning, Quantization, and Bitwise Operations for Reduced Storage and Improved Retrieval Speed. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

In the era of big data and large-scale machine learning applications, efficient representation of textual information is crucial. Embeddings serve as foundational elements in natural language processing (NLP) and information retrieval (IR), transforming text into high-dimensional vectors that capture semantic and syntactic nuances. While high-dimensional embeddings provide rich representations, they come with substantial storage costs and computational overhead, particularly in environments with limited resources or when deploying models to edge devices.

Reducing the size of embeddings without sacrificing performance has become an important research area. Traditional methods such as dimensionality reduction and quantization have been employed, but often at the expense of degrading the quality of embeddings or losing critical information. Optimizing embeddings to be both compact and efficient while retaining high performance remains a significant challenge.

In this paper, we present **Tiny Embeddings**, an innovative approach that addresses the challenges of storage and retrieval efficiency. Our method leverages *Matryoshka Representation Learning*, named after the Russian nesting dolls, where embeddings of smaller dimensions are nested within larger ones. By adding additional feed-forward neural network (FFN) layers and specialized loss functions to existing embedding models, we imbue them with the Matryoshka property. This ensures that essential information is concentrated in the early dimensions, allowing for the use of lower-dimensional embeddings without significant performance degradation.

The **Matryoshka property** refers to the hierarchical nesting of embeddings, analogous to Russian nesting dolls, where embeddings of smaller dimensions are subsets of larger ones. This property allows for dynamic selection of embedding dimensions based on resource constraints or performance requirements without retraining the model. By concentrating more information in the early dimensions, we enable the use of lower-dimensional embeddings that still capture essential semantic content.

Furthermore, we introduce trainable quantization techniques, applying 0.5-bit, 1-bit, 1.5-bit, and 2-bit quantization levels to the embeddings. By reducing the numeric precision, we decrease storage requirements substantially. The quantization is made trainable through added FFN layers and loss functions that promote quantization during training.

To enhance retrieval speed, especially on commodity hardware, we propose extending the use of Hamming distance and similarity to multi-bit quantized embeddings. By mapping multi-bit embeddings to higher-bit binary representations (e.g., converting 1.5-bit to 2-bit or 2-bit to 3-bit), we enable efficient computation of similarity using bitwise operations such as XOR, NOT, and POPCOUNT. These operations are highly optimized on modern CPUs, resulting in significantly faster retrieval compared to traditional floating-point calculations used in cosine similarity.

Our hybrid architecture assigns higher precision quantization (e.g., 2-bit) to the early embedding dimensions, which contain more critical information, and lower precision quantization (e.g., 1-bit or 0.5-bit) to later dimensions with less information content. This

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

alignment ensures an optimal balance between storage efficiency and representation quality.

An important insight from our work is that simply selecting embedding dimensions based on variance or gradient importance did not yield satisfactory results. This observation led us to employ Matryoshka representation learning, which effectively concentrates information in early dimensions and overcomes the limitations of naive dimension selection methods.

Our contributions can be summarized as follows:

(1) **Integration of Matryoshka Representation Learning**: We enhance existing embedding models by adding FFN layers and specialized loss functions to enforce the Matryoshka property, empirically demonstrating that addtional FFN layers can learn this property with light weight training without requiring pre-training of full models for this property. This allows the use of fewer embedding dimensions without significant performance loss.
(2) **Trainable Quantization with Multiple Bit Levels**: We introduce trainable quantization techniques using 0.5-bit, 1-bit, 1.5-bit, and 2-bit quantization levels through additional FFN layers and quantization-promoting loss functions, reducing storage requirements by decreasing numeric precision.
(3) **Novel Quantization Levels**: We explore the application of **0.5-bit, 1-bit, 1.5-bit, and 2-bit quantization**. By reducing the bits used per dimension, we significantly decrease the storage requirements of the embeddings while controlling the trade-off between compression and performance.
(4) **Hybrid Quantization Architecture**: We propose a hybrid architecture that applies different quantization levels to different segments of the embedding dimensions, allocating higher precision to dimensions with more information content. Specifically, we use 2-bit quantization for the early K dimensions, 1.5-bit for the next L dimensions, 1-bit for the subsequent M dimensions, and 0.5-bit for the remaining dimensions. This design ensures that dimensions with higher information content receive finer-grained representation, aligning storage precision with the importance of the information captured.
(5) **Efficient Similarity Computation Using Bitwise Operations**: We propose using Hamming distance and Hamming similarity on the bitwise representations of embeddings. For 1.5-bit and 2-bit quantization levels, we map embeddings to higher-bit binary representations (e.g., converting 1.5-bit to 2-bit and 2-bit to 3-bit) to facilitate efficient computation using bitwise operations such as XOR, NOT, and POPCOUNT. This approach improves retrieval speed by leveraging CPU instructions optimized for such operations.

simply selecting embedding dimensions based on variance or gradient importance did not yield satisfactory results. This observation underscores the effectiveness of our Matryoshka learning approach, which inherently prioritizes information content in the early dimensions. Our experimental evaluations on benchmark datasets demonstrate that Tiny Embeddings achieve significant reductions in storage requirements and retrieval times while maintaining competitive accuracy compared to uncompressed models. The proposed methods offer practical solutions for deploying efficient embeddings in large-scale IR systems and resource-constrained environments.

The remainder of this paper is organized as follows: Section 2 reviews related work on embedding compression and quantization. Section 3 details our proposed methods, including the mathematical formulations of Matryoshka learning and quantization techniques. Section 4 presents the experimental setup and results. Finally, Section 6 concludes the paper and discusses future research directions.

2 RELATED WORK

Large-scale language models and high-dimensional embeddings have greatly advanced Natural Language Processing (NLP) and Information Retrieval (IR). However, they often demand substantial storage and compute. Various compression techniques have thus emerged to reduce size while retaining performance. This section discusses embedding approaches in IR, combined strategies for compressing large language models (LLMs) and embeddings, and Matryoshka Representation Learning (MRL).

2.1 Embedding Models in Information Retrieval

Early word embeddings like Word2Vec [**?**] and GloVe [**?**] paved the way for dense vector representations. Subsequent transformers, including BERT [**?**] and RoBERTa [**?**], introduced contextual embeddings with higher accuracy but also higher memory and compute costs. Recent research prioritizes compact architectures for deployment efficiency.

2.2 Compression Techniques for Language Models and Embeddings

Pruning and Distillation. **?**] proposed *Deep Compression*, combining pruning, quantization, and coding. The *Lottery Ticket Hypothesis* [**?**] highlights subnetworks that can perform on par with their original dense models. Knowledge distillation [**? ? ?**] transfers capability from large "teacher" models to smaller "student" models.

Parameter-Efficient Training and Factorization. Adapters [**?**], Prefix Tuning [**?**], and other techniques train only a fraction of parameters. Low-rank factorization [**? ?**] further trims storage and compute.

Quantization. Both post-training [**?**] and quantization-aware training [**? ?**] constrain numerical precision. Binary and ternary embeddings [**? ?**] operate efficiently via bitwise operations. Product Quantization [**?**] partitions embeddings into sub-vectors for approximate nearest neighbor search.

Dimensionality Reduction and Pruning for Embeddings. Methods like PCA [**?**], SVD [**?**], and autoencoders [**?**] systematically compress embeddings, while t-SNE and UMAP focus primarily on visualization by preserving local neighborhoods at the expense of global structure and are unsuitable for information retrieval tasks due to their non-parametric nature and inability to process new data without recomputation. In contrast, our approach creates hierarchical, nested embeddings through explicit loss functions that concentrate essential semantic information in early dimensions while maintaining cross-scale consistency, enabling dynamic dimension selection without retraining and preserving both local and global relationships crucial for retrieval tasks. Pruning low-variance dimensions [**?**] can yield more compact models with minimal loss in semantic fidelity, but lacks the systematic information organization and quantization awareness of our approach.

Efficient Similarity Computation. In large-scale IR, hashed or binarized representations expedite retrieval via Hamming distance (XOR, POPCOUNT) [**?**], or via Locality-Sensitive Hashing (LSH) [**?**] to approximate nearest neighbors.

2.3 Matryoshka Representation Learning

Matryoshka Representation Learning (MRL) [8] aims to create hierarchical embeddings where smaller embeddings are nested within larger ones, analogous to Russian nesting dolls. This approach allows models to adaptively use embeddings of different sizes based on resource constraints, without the need for retraining. MRL focuses on generating these hierarchical embeddings such that truncated dimensions can still yield functional representations. While prior works have attempted to promote such nesting properties, they typically did so without explicit loss functions tailored to enforce them.

2.4 Our Approach in Context

We unify MRL with finer-grained quantization (e.g., 0.5-bit, 1.5-bit) and bitwise operations for efficient storage and retrieval. Unlike previous MRL methods, we incorporate three explicit *Losses to Promote Matryoshka Property* to concentrate key information in early dimensions. We also use a hybrid quantization scheme assigning varying bit-levels to embedding slices, balancing performance and storage for diverse IR scenarios.

3 METHODOLOGY

3.1 Overview

Our approach combines multiple techniques to create efficient and compact embeddings while preserving semantic relationships. The key components include:

- Neural transformation layers for improved quantization and matryoshka representation learning
- Hybrid architecture combining different quantization levels
- Loss functions to promote quantization and matryoshka representation learning
- Efficient bit-packing and similarity computation

Fig. 1. Overall architecture of Tiny Embeddings system showing the transformation, quantization, and bit-packing stages.

3.2 Quantization Framework

Our quantization framework converts high-precision embeddings into compact, discrete representations while preserving semantic relationships through a combination of neural transformations and adaptive threshold-based discretization. We first apply a feedforward transformation $\mathbf{z} = \text{FFN}(\mathbf{x}; \theta)$ to optimize the embedding space for quantization, where $\mathbf{x}$ is the input embedding and θ represents trainable parameters.

Multi-level Quantization. We implement multiple quantization levels for flexible storage-performance trade-offs: 2-bit (4 levels), 1.5-bit (3 levels), 1-bit (2 levels), and 0.5-bit (combines dimension pairs) quantization. For k-bit quantization with 2^k levels, we use $(2^k - 1)$ learnable thresholds per dimension to partition the continuous embedding space. For instance, 2-bit quantization employs three thresholds $\{\theta_1, \theta_2, \theta_3\}$ to create four bins, while 1-bit quantization uses a single threshold for binary partitioning.

Threshold-based Discretization and Learning. For a d-dimensional embedding space $\mathbf{X} \in \mathbb{R}^{N \times d}$, we initialize thresholds using percentile statistics to ensure balanced quantization:

$$\theta_{i,j} = \text{Quantile}\big(\{x_{1,j}, \ldots, x_{N,j}\},\ q_i\big) \tag{1}$$

where q_i are evenly-spaced quantiles in $(0, 1)$. During training, thresholds adapt via exponential moving average:

$$\theta_{i,j}^{(\text{new})} = \mu\, \theta_{i,j}^{(\text{old})} + (1 - \mu)\ \widehat{\theta}_{i,j}^{(\text{batch})} \tag{2}$$

with momentum coefficient $\mu \in [0, 1]$ and batch-computed thresholds $\widehat{\theta}_{i,j}^{(\text{batch})}$.

Quantization Process. The complete process for an input embedding involves: (1) applying the trainable transformation $\mathbf{z} = \text{FFN}(\mathbf{x})$, (2) normalizing $\hat{\mathbf{z}} = \text{normalize}(\mathbf{z})$, and (3) assigning discrete codes:

$$q_j = \sum_{i=1}^{2^k - 1} \mathbb{1}[\hat{z}_j > \theta_{i,j}] \tag{3}$$

End-to-End Trainability. Our framework achieves end-to-end trainability by optimizing continuous-valued embeddings while simultaneously guiding them toward quantization-friendly distributions. Rather than backpropagating through non-differentiable quantization operations, we apply quantization regularization losses (Section 3.4.3) that encourage embeddings to cluster away from quantization boundaries. The thresholds themselves are updated using statistics from the continuous embeddings, ensuring the quantization boundaries evolve to match the learned representation space, while the transformation network learns to produce embeddings that naturally align with discrete quantization levels, enabling training without direct gradients through the discretization step. Experimental results (Section 4) demonstrate significant compression while maintaining high retrieval accuracy across diverse tasks.

3.3 Promoting Information High Density in Early Dimensions

Matryoshka Representation Learning (MRL), inspired by Russian nesting dolls, addresses a fundamental challenge in embedding systems: creating representations that remain effective when truncated to smaller dimensions. Traditional embeddings often lose critical information when dimensions are removed since important features are distributed across all dimensions. Our approach extends MRL with additional control mechanisms to create highly efficient, nested representations where the most crucial information is concentrated in early dimensions.

3.3.1 Core Matryoshka Architecture.

Nested Representation Generation. Given an input embedding $\mathbf{x} \in \mathbb{R}^d$, we generate a series of nested representations at different dimension levels $D = \{d_1, d_2, ..., d_K\}$ where $d_1 < d_2 < ... < d_K = d$. Each level k contains all information from previous levels plus additional features:

$$\mathbf{y}_k = f_k(\mathbf{x}) \in \mathbb{R}^{d_k}, \quad k = 1, \ldots, K \tag{4}$$

The key nesting property ensures that smaller representations are perfect subsets of larger ones:

$$\mathbf{y}_i[1 : d_j] = \mathbf{y}_j \quad \text{for } j < i \tag{5}$$

Base Transformation Network. We implement the transformation function f_k using the same lightweight yet effective feed-forward network as introduced in the quantization framework, where $\mathbf{h} = \text{FFN}(\mathbf{x}) = W_2(\text{GELU}(W_1\mathbf{x}))$. Here, $W_1 \in \mathbb{R}^{32d \times d}$ expands the representation for richer feature extraction, GELU activation adds nonlinearity while maintaining smooth gradients, and $W_2 \in \mathbb{R}^{d \times 32d}$ projects back to original dimension space. This shared architecture ensures consistency across both the matryoshka representation learning and quantization stages of our pipeline.

Progressive Dimension Slicing. Unlike traditional MRL implementations that use separate projections for each scale, we employ an efficient progressive slicing mechanism:

$$\mathbf{y}_k = \text{Slice}_{d_{k-1}}^{d_k}(\mathbf{h}) \tag{6}$$

The full embedding at level k is constructed through concatenation:

$$\mathbf{e}_k = [\mathbf{y}_1; \mathbf{y}_2; ...; \mathbf{y}_k] \tag{7}$$

This approach inherently maintains the nested property without requiring explicit constraints, reducing computational overhead.

Combined Loss Function. Our training objective combines multiple loss components, each with their corresponding weights, applied at each dimension level:

$$\mathcal{L}_{\text{total}} = \sum_{d \in D} \sum_{l \in \mathcal{L}} w_l^{(d)} \mathcal{L}_l^{(d)} \tag{8}$$

where $w_l^{(d)}$ represents the weight for loss component l at dimension level d.

3.4 Loss Functions and Training Objectives

In our proposed Matryoshka Representation Learning (MRL) framework, we integrate both *similarity preservation* and *matryoshka adaptation* losses to achieve nested, multilevel embeddings that maintain semantic quality at each quantization level. Additionally, a *quantization regularization* component ensures that the learned representation is readily quantizable with minimal loss in accuracy. Below, we detail each category of losses and explain how they contribute to our overall training scheme.

3.4.1 Similarity Preservation.

Direct Similarity MSE Loss. We first ensure preservation of pairwise similarities via a mean squared error (MSE) loss on the normalized similarity matrices:

$$\mathcal{L}_{\text{sim}}(\mathbf{X}, \mathbf{Y}) = \|\mathrm{S}_X - \mathrm{S}_Y\|_F^2 \tag{9}$$

where:

$$\mathrm{S}_X = \text{normalize}(\mathbf{X})\,\text{normalize}(\mathbf{X})^T, \quad \mathrm{S}_Y = \text{normalize}(\mathbf{Y})\,\text{normalize}(\mathbf{Y})^T \tag{10}$$

preserving the cosine similarities between all pairs of embeddings $\mathbf{X}$ and $\mathbf{Y}$ at different quantization levels.

KL Divergence Loss. To align the probability distributions of similarities, we employ a symmetric KL divergence with temperature scaling:

$$\mathbf{P}_X = \text{softmax}(\mathrm{S}_X/\tau), \quad \mathbf{P}_Y = \text{softmax}(\mathrm{S}_Y/\tau), \tag{11}$$

$$\mathcal{L}_{\text{kl}} = \tfrac{1}{2}\big[D_{\text{KL}}(\mathbf{P}_X \| \mathbf{P}_Y) + D_{\text{KL}}(\mathbf{P}_Y \| \mathbf{P}_X)\big], \tag{12}$$

where τ controls the sharpness of the distribution.

Rank Preservation Loss. To maintain the *relative ordering* of similarities, we implement a differentiable ranking loss:

$$\mathcal{L}_{\text{rank}} = \sum_{i,j,k} \sigma\big((s_{ij}^X - s_{ik}^X) \cdot t\big) \odot \sigma\big((s_{ij}^Y - s_{ik}^Y) \cdot t\big), \tag{13}$$

where $\sigma(\cdot)$ is the sigmoid function, t sharpens the sigmoid, and s_{ij}^X, s_{ij}^Y denote pairwise similarities in the original and quantized embeddings respectively.

Contrastive Learning. We also incorporate contrastive losses with positive and negative pairs:

$$\mathcal{L}_{\text{contrast}} = -\log \frac{\exp(s_p/\tau)}{\exp(s_p/\tau) + \sum_n \exp(s_n/\tau)}, \tag{14}$$

where s_p and s_n represent the similarities for positive and negative pairs, respectively.

3.4.2 Losses to Promote Matryoshka Property.

To further support the *Matryoshka* property (i.e., nested and progressively expanding embeddings), we introduce three *matryoshka adaptation* losses that manipulate the distribution and uniqueness of information at each dimension level.

Progressive Information Bottleneck. We encourage essential features to concentrate in earlier dimensions:

$$\mathcal{L}_{\text{ib}} = \sum_{i=1}^{n} w_i \left(\frac{|\mathbf{x}_i|^2}{x_0 + |\mathbf{x}_i|}\right)^{\alpha}, \tag{15}$$

where $w_i = i/n$ progressively increases with dimension index i, $x_0 \approx 0.1$ is a small suppression threshold, and $\alpha \approx 0.3$ controls the gradient strength near zero. This approach pushes less important information toward zero in higher dimensions.

Inter-level Orthogonal Information Encoding. We promote uniqueness of newly added dimensions by enforcing orthogonality with respect to previously formed dimensions:

$$\mathcal{L}_{\text{orth}} = \sum_{l=2}^{L} \sum_{j=1}^{l-1} \|\Delta_l^T \cdot \text{normalize}(\mathbf{E}_j)\|_F, \tag{16}$$

where Δ_l indicates the newly added dimensions at level l, and $\mathbf{E}_j$ consists of all dimensions up to level j. This encourages each new level of dimensions to encode novel information rather than rediscovering existing patterns.

Adaptive Variance Control. To prevent embedding collapse while allowing selective dimension suppression, we employ a slowly increasing variance penalty:

$$\mathcal{L}_{\text{var}}(t) = \max\Big(0.2,\ \frac{e^{t/T} - 1}{e - 1}\Big) \sum_{d=1}^{D} \exp(-\sigma_d), \tag{17}$$

where t is the current training step, T is the total training steps, and σ_d is the standard deviation of dimension d. This loss starts with a weak penalty and grows over time, preventing early collapse and encouraging dimensions that truly matter to preserve sufficient variance.

3.4.3 Quantization Regularization Loss.

While the above mechanisms reinforce nested representation properties, we also require a loss term to facilitate *quantization readiness*. This term discourages embeddings from clustering near quantization thresholds and encourages them to lie closer to valid quantized levels.

Formulation. Let θ be a quantization threshold for 1-bit quantization, or $\{\theta_1, \theta_2, \theta_3\}$ be thresholds for 2-bit quantization. Denoting each embedding value as x, our quantization regularization can be written as:

$$\mathcal{L}_{\text{quant}} = w(t)\Big[\exp\big(-(\min_i |x - \theta_i|)\big) + \lambda\, L_{\text{range}}(x)\Big], \tag{18}$$

where $w(t)$ progresses from 0.2 to 1.0 over training to gradually strengthen quantization constraints, and $L_{\text{range}}(x)$ enforces that x lies within a valid range (e.g., $[0, 1]$ or another permissible interval). Specifically,

$$L_{\text{range}}(x) = \text{ReLU}(l - x)^2 + \text{ReLU}(x - u)^2, \tag{19}$$

where l and u are the lower and upper bounds of acceptable ranges. Intuitively, this loss repels the embedding away from threshold boundaries (to avoid ambiguity) and adjusts overly large or negative values back into a valid quantization domain.

By combining these losses, we achieve a robust training procedure that satisfies several objectives simultaneously:

- **Similarity Preservation:** $\mathcal{L}_{\text{sim}}$, $\mathcal{L}_{\text{kl}}$, $\mathcal{L}_{\text{rank}}$, $\mathcal{L}_{\text{contrast}}$ ensure semantic fidelity across different quantization levels.
- **Matryoshka Property:** $\mathcal{L}_{\text{ib}}$, $\mathcal{L}_{\text{orth}}$, $\mathcal{L}_{\text{var}}$ organize the learned features hierarchically, pushing essential features into early dimensions while maintaining uniqueness and stable variance distribution across expanding dimensional levels.
- **Quantization Readiness:** $\mathcal{L}_{\text{quant}}$ directly encourages embeddings to settle into regions that are easily separable by the quantizer, thereby reducing errors from threshold-based discretization.

Finally, these losses can be summed with appropriate weights to form the complete training objective. This unified framework delivers *nested, increasingly robust* embeddings within a single training pipeline, enabling flexible deployment scenarios where early dimensions provide quick approximate search, while further dimensions refine accuracy for more demanding tasks.

3.5 Hybrid Architecture

We employ a hierarchical scheme that applies different quantization levels to different fractions of the embedding dimensions based on their information content:

- **First 25% (highest information):** 2-bit quantization (expanded to 3-bit codes)
- **Next 25% (medium-high):** 1.5-bit quantization (3 levels, expanded to 2-bit codes)
- **Next 25% (medium-low):** 1-bit quantization (direct binary)
- **Final 25% (lowest):** 0.5-bit quantization (pairs of dimensions reduced via a lightweight FFN)

This arrangement yields an average of 1.625 bits per dimension, balancing storage efficiency and similarity preservation by granting higher precision to dimensions with greater semantic importance.

Dimension Reduction for 0.5-bit Encoding. For the last quarter of dimensions, pairs are combined:

$$\mathbf{z}_{\text{reduced}} = \text{FFN}_{\text{reduce}}\Big([\mathbf{x}_{2i}, \mathbf{x}_{2i+1}]\Big), \tag{20}$$

where a specialized FFN preserves essential information while halving the dimensional load. Figure **??** illustrates the full hybrid quantization pipeline.

3.6 Efficient Storage and Similarity Computation

We combine bit-level optimizations with carefully designed expansions that preserve semantic similarity in quantized embeddings while minimizing storage overhead.

3.6.1 *Bit Packing and Storage.*

Bit Packing Process:

(1) Convert quantized vectors to binary form.
(2) Pack bits into N-bit words ($N \in \{8, 16, 32\}$) via:

$$\mathbf{w}_i = \sum_{j=0}^{N-1} b_{Ni+j} \cdot 2^{N-j-1}, \tag{21}$$

where b_k is the k-th bit.
(3) Add padding bits if needed.
(4) Store as `uint64` arrays for optimal alignment.

3.6.2 *Binary Codebook Expansion.*

2-bit to 3-bit:

$$\text{Codebook}_{2\to3} = \begin{cases} 0 \to [0, 0, 0] \\ 1 \to [0, 0, 1] \\ 2 \to [0, 1, 1] \\ 3 \to [1, 1, 1] \end{cases} \tag{22}$$

Expanding 2-bit values (0–3) to 3 bits preserves relative distances more accurately, preventing similar embeddings from collapsing to identical Hamming codes. For example:

- Original 2-bit values [0,1] and [1,0] have equal Hamming distance to [1,1]
- Expanded codes [0,0,1] and [0,1,1] have different Hamming distances to [1,1,1]
- This better reflects the original quantization levels' relationships

1.5-bit to 2-bit:

$$\text{Codebook}_{1.5\to2} = \begin{cases} 0 \to [0, 0] \\ 1 \to [0, 1] \\ 2 \to [1, 1] \end{cases} \tag{23}$$

Maps three-level codes to 2 bits, improving similarity preservation while keeping compact storage.

3.6.3 *Hamming Similarity Computation.*

Basic Hamming Similarity: For binary vectors $\mathbf{x}, \mathbf{y}$ of length n (bits),

$$\text{sim}_H(\mathbf{x}, \mathbf{y}) = 1 - \frac{\text{POPCOUNT}(\mathbf{x} \oplus \mathbf{y})}{n}, \tag{24}$$

where $\oplus$ is XOR and POPCOUNT counts set bits.

Implementation Details: Modern CPUs offer hardware-accelerated `XOR` and `POPCNT` instructions (optionally AVX-512) enabling high-throughput Hamming distance computation on packed 64-bit words. We leverage `numpy.packbits`, `numpy.unpackbits`, `numpy.bitwise_xor`, and `numpy.bitwise_count` for fast, vectorized bit manipulation and codebook expansions. These strategies tightly pack quantized representations, expand codes to preserve fine-grained similarities,

and compute large-scale Hamming distances rapidly, yielding an efficient, similarity-aware embedding system.

4 EXPERIMENTAL SETUP

We evaluated our approach using a suite of benchmark retrieval datasets from the MTEB (Massive Text Embedding Benchmark) library [10], including ArguAna [15, 17], CQADupstackTexRetrieval [6], ClimateFEVER [4], DBPedia [5], FEVER [16], FiQA2018 [15], HotpotQA [21], MSMARCO [11], NFCorpus [2], NQ (Natural Questions) [9], QuoraRetrieval [7, 15], SCIDOCS [3], SciFact [18], TREC-COVID [14, 15], and Touche2020 [1].

For our experiments, we employed two distinct Transformer models: MiniLM [13, 19], a compact 12-layer model with 384-dimensional embeddings derived through knowledge distillation, and Modern BERT (MB) [12, 20], a more recent 22-layer architecture with 768-dimensional embeddings incorporating state-of-the-art design principles. This selection allows us to evaluate our approach across different model scales and architectural generations, from efficient distilled models to modern high-capacity architectures. Demonstrating robust results on both a smaller, older architecture and a more recent, higher-capacity model indicates that practitioners can attain substantial compression benefits—via quantization and nested dimensionality reduction—across diverse Transformer-based architectures.

We compared our approach against the following baselines: **Original Model (FP32)** (full-precision 32-bit floating point embeddings), **FP16** (half-precision 16-bit), **Int8** (8-bit integer quantization), and **Simple Threshold Quantization** (basic 2-bit, 1.5-bit, and 1-bit quantization using fixed thresholds without learned transformations or Matryoshka representation).

The training process optimizes our Matryoshka model $\mathcal{M}$ initialized from a pretrained model $\mathcal{E}$. For each batch, normalized base embeddings are transformed to produce both non-quantized and quantized representations at different dimension levels. The training objective combines multiple weighted loss terms: similarity preservation losses ($\mathcal{L}_{\text{sim}}$, $\mathcal{L}_{\text{kl}}$, $\mathcal{L}_{\text{rank}}$, $\mathcal{L}_{\text{contrast}}$), orthogonality loss for unique information across levels, information bottleneck loss to concentrate important features early, and quantization loss to facilitate discretization. Quantization thresholds are initialized using percentile-based statistics and updated with momentum. We used AdamW optimizer with $\eta = 1 \times 10^{-4}$, warm-up scheduling, and gradient clipping, training for 5 epochs. For evaluation, we used NDCG@10 which measures ranking quality by comparing the relevance scores of retrieved documents against an ideal ranking, normalized to [0,1].

5 RESULTS AND ANALYSIS

In this section, we present the experimental results of our proposed QAMA (Quantization Aware Matryoshka Adaptation) approach using two different models: Modern BERT (MB) [12, 20] and MiniLM [13, 19]. We evaluate the performance across various embedding dimensions and quantization levels to demonstrate the effectiveness of our methods in reducing storage requirements and improving retrieval speed while maintaining high accuracy.

5.1 Main Results

Tables 1 and 2 compare the performance of Modern BERT (MB) and MiniLM at various quantization levels and dimensions, using nDCG@10 average across multiple datasets as the primary metric.

Key Insights:

- **2-bit Quantization Nears FP32 Performance:** As shown in the ablation studies, 2-bit quantization retains around 95–98% of FP32 performance, demonstrating its effectiveness in preserving semantic information.
- **Hybrid Quantization Efficiency:** The Hybrid approach offers additional storage savings while maintaining performance close to 2-bit, as evidenced by the ablation results.
- **Role of Matryoshka Loss:** The introduction of Matryoshka Loss significantly enhances performance, particularly at lower dimensions, by concentrating essential information in early dimensions.

Table 1. Performance of Modern BERT (MB) with different quantization levels and embedding dimensions.

Dimension	1-bit	1.5-bit	Hybrid Quant	2-bit	FP32
768	0.4381	0.4536	0.4680	0.4687	0.4720
384	0.4167	0.4429	0.4509	0.4593	0.4695
256	0.3691	0.4228	0.4465	0.4513	0.4680
192	0.3285	0.3901	0.4245	0.4327	0.4512
96	0.2908	0.3455	0.3850	0.3919	0.4247

Table 2. Performance of MiniLM with different quantization levels and embedding dimensions.

Dimension	1-bit	1.5-bit	Hybrid Quant	2-bit	FP32
384	0.3839	0.4101	0.4160	0.4185	0.4286
192	0.3724	0.3923	0.4017	0.4109	0.4219
128	0.3571	0.3814	0.3865	0.3917	0.3963
96	0.3417	0.3649	0.3695	0.3712	0.3792
48	0.2687	0.2871	0.2919	0.2897	0.3014

From the results, we observe that our proposed QAMA framework achieves competitive performance compared to the full-precision (FP32) embeddings, even at significantly reduced storage sizes. For instance, the 2-bit quantized embeddings at 384 dimensions achieve an nDCG@10 of 0.4593 with Modern Bert and 0.4185 with MiniLM, which is close to the FP32 performance while reducing the storage requirements substantially. Although slightly behind pure 2-bit in accuracy, Hybrid Quant offers additional storage savings ($\approx$ 1–4% more than 2-bit) while keeping performance within 1–2% of the best quantized results.

5.2 Impact of Quantization Levels

We further analyze the impact of different quantization levels on the performance. Figures 2 and 3 illustrate the performance across different quantization levels for MB and MiniLM models, respectively.

Fig. 2. Impact of quantization levels on performance for Modern BERT (MB) model across different dimensions.

Fig. 3. Impact of quantization levels on performance for MiniLM model across different dimensions.

The figures show that increasing the quantization level (from 1-bit to 2-bit) consistently improves performance, highlighting the trade-off between storage efficiency and model accuracy. The Hybrid Quantization approach, which uses different quantization levels for different embedding dimensions, achieves performance close to 2-bit quantization while offering better storage efficiency.

5.3 Effect of Embedding Dimensions

Figures 4 and 5 show that higher dimensions generally yield stronger performance due to increased representational capacity. However, our methods maintain competitive accuracy at lower dimensions. In particular, the ablation results highlight that, at around 192 dims for MB or 96 dims for MiniLM, the Matryoshka Loss and Orthogonality Regularization become increasingly critical (see Table 4 and Table 6). For example, MB at 192 dims with 2-bit quantization retains 0.4327 nDCG@10, compared to 0.4512 for FP32 (i.e., around 96% of the full-precision performance). At lower dimensions, Matryoshka Loss becomes critical in preserving performance by ensuring hierarchical information encoding. This capability—preserving semantic effectiveness even in a lower-dimensional space—emerges from the synergy of hierarchical representation learning plus the advanced regularizations.

Fig. 4. Impact of embedding dimensions on performance for Modern BERT (MB) model across different quantization levels.

Fig. 5. Impact of embedding dimensions on performance for MiniLM model across different quantization levels.

These figures illustrate that while higher dimensions generally improve performance, our proposed methods enable lower-dimensional embeddings to achieve competitive results, which is beneficial for resource-constrained environments.

5.4 Ablation Studies

To evaluate the contribution of each component in our proposed methodology, we perform ablation studies on the MB and MiniLM models at two embedding dimensions each. Specifically, we consider dimensions 384 and 192 for the MB model, and dimensions 192 and 96 for the MiniLM model. Tables 3 and 4 present the ablation results for the MB model, while Tables 5 and 6 show the results for the MiniLM model. We highlight a few *aha* observations here:

- **Matryoshka Loss Yields Large Jumps in Lower Dimensions:** E.g., for MiniLM at 96 dims, final performance doubles from *Thresholds Only* to *+ MVC + Orthogonality + IB* (Table 6). This underscores that hierarchical training is essential when dimension is aggressively reduced.
- **Adaptive Variance Control as a Final "Booster" Step:** In each table, AVC tends to add a 2–4% relative improvement. This mechanism is vital for preventing degenerate embeddings in scenarios with both low bits and low dimensions.
- **Orthogonality and Bottleneck Synergy:** These regularizations spread out distinct features among the newly added dimensions (Orthogonality) while forcing early dimensions to capture the most crucial semantic content (Info Bottleneck).

Model-Specific Observations:

- **MB vs. MiniLM:** MB generally retains better performance at higher dimensions, while MiniLM benefits significantly from the matryoshka methods, indicating the approach's adaptability across different architectures.
- **Sensitivity to Compression:** MB shows resilience to dimensional downscaling, whereas MiniLM exhibits faster trade-off declines, reflecting differences in model architecture.

Table 3. Ablation study for MB model at 384 dimensions.

Training Components	1-bit	1.5-bit	Hybrid Quant	2-bit	FP32
Thresholds Only	0.3900	0.4250	0.4358	0.4370	0.4395
+ Trainable FFN Transform	0.3922	0.4275	0.4380	0.4390	0.4410
+ Quantization Loss	0.3945	0.4298	0.4402	0.4415	0.4430
+ Matryoshka Loss	0.3998	0.4318	0.4410	0.4425	0.4632
+ Orthogonality Regularization	0.4012	0.4329	0.4430	0.4440	0.4645
+ Information Bottleneck Regularization	0.4025	0.4340	0.4442	0.4455	0.4658
+ Adaptive Variance Control	**0.4167**	**0.4429**	**0.4509**	**0.4593**	**0.4695**

Table 4. Ablation study for MB model at 192 dimensions.

Training Components	1-bit	1.5-bit	Hybrid Quant	2-bit	FP32
Thresholds Only	0.2900	0.3550	0.3755	0.3850	0.3880
+ Trainable FFN Transform	0.2922	0.3575	0.3778	0.3872	0.3900
+ Quantization Loss	0.2943	0.3597	0.3801	0.3893	0.3920
+ Matryoshka Loss	0.3107	0.3732	0.3945	0.4050	0.4405
+ Orthogonality Regularization	0.3125	0.3752	0.3964	0.4068	0.4422
+ Information Bottleneck Regularization	0.3142	0.3770	0.3983	0.4087	0.4440
+ Adaptive Variance Control	**0.3285**	**0.3901**	**0.4245**	**0.4327**	**0.4512**

For the MiniLM model, the ablation results are as follows:
Analyzing the results, we observe several key findings:

Performance at Different Dimensions: At both higher and lower dimensions, our models maintain competitive performance levels. For instance, even at 192 dimensions, the MB model achieves an nDCG@10 of 0.4327 with 2-bit quantization, which is close to the full-precision performance of 0.4512. Similarly, the MiniLM model at 96 dimensions reaches an nDCG@10 of 0.3712 with 2-bit quantization, compared to 0.3792 in full precision.

Table 5. Ablation study for MiniLM model at 192 dimensions.

Training Components	1-bit	1.5-bit	Hybrid Quant	2-bit	FP32
Thresholds Only	0.2947	0.3205	0.3260	0.3324	0.3398
+ Trainable FFN Transform	0.3112	0.3538	0.3645	0.3753	0.3841
+ Quantization Loss	0.3267	0.3661	0.3773	0.3883	0.3958
+ Matryoshka Loss	0.3318	0.3709	0.3821	0.3922	0.3996
+ Orthogonality Regularization	0.3354	0.3741	0.3850	0.3940	0.4019
+ Information Bottleneck Regularization	0.3389	0.3768	0.3876	0.3961	0.4035
+ Adaptive Variance Control	**0.3724**	**0.3923**	**0.4017**	**0.4109**	**0.4219**

Table 6. Ablation study for MiniLM model at 96 dimensions.

Training Components	1-bit	1.5-bit	Hybrid Quant	2-bit	FP32
Thresholds Only	0.2050	0.2330	0.2375	0.2428	0.2483
+ Trainable FFN Transform	0.2250	0.2570	0.2715	0.2860	0.2925
+ Quantization Loss	0.2640	0.3017	0.3161	0.3304	0.3344
+ Matryoshka Loss	0.3100	0.3350	0.3450	0.3550	0.3600
+ Orthogonality Regularization	0.3150	0.3380	0.3480	0.3575	0.3625
+ Information Bottleneck Regularization	0.3175	0.3400	0.3500	0.3590	0.3640
+ Adaptive Variance Control	**0.3417**	**0.3649**	**0.3695**	**0.3712**	**0.3792**

Impact of Matryoshka Loss and Associated Regularizations: As shown in the ablation studies (Tables 3, 4, 5, and 6), introducing the Matryoshka Loss yields significant performance gains across all quantization levels and dimensions. For instance, in Table 4 at 192 dimensions (2-bit), the Modern BERT (MB) model's nDCG@10 score improves from 0.3850 (*Thresholds Only*) to 0.4050 upon adding Matryoshka Loss. An even larger jump can be seen for MiniLM at 96 dimensions (Table 6), where 2-bit quantization increases from 0.2428 (*Thresholds Only*) to 0.3550 after Matryoshka Loss—over a 46% relative gain.

Further improvements occur once Orthogonality and Information Bottleneck Regularizations are applied, especially at lower dimensions. For example, in Table 4 with MB at 192 dimensions (2-bit), these two regularizations boost the nDCG@10 score from 0.4050 to 0.4087 before Adaptive Variance Control is introduced, illustrating how newly added dimensions learn orthogonal (less redundant) features while penalizing unhelpful information in higher dimensions.
Adaptive Variance Control (AVC): AVC then provides an additional and sometimes substantial performance jump by preventing degenerate embedding distributions and refining the variance structure across dimensions. In Table 4 (MB at 192 dimensions, 2-bit), AVC lifts the nDCG@10 from 0.4087 to 0.4327, closing much of the gap to full-precision performance. Likewise, for MiniLM at 96 dimensions (Table 6), cumulative additions of Matryoshka Loss and other regularizations yield 0.3590 at 2-bit—a marked improvement over 0.2428 (*Thresholds Only*)—and further rise to 0.3712 with AVC. These incremental gains underscore the synergy among Matryoshka Loss, Orthogonality, Information Bottleneck, and AVC in enabling robust performance, even under aggressive compression settings.
Trainable FFN Transform: Another integral component of our framework is the Trainable FFN Transform, which underpins both quantization and Matryoshka training. As seen in Table 6, adding the FFN Transform immediately after *Thresholds Only* substantially boosts nDCG@10 from 0.2428 to 0.2860 at 2-bit, a relative improvement of about 17.8%. Similarly, at 1.5-bit quantization, performance

jumps from 0.2330 to 0.2570. This learnable transformation effectively redistributes and enhances features, making them more resilient to the subsequent discrete binning process. It ensures that critical semantic information is captured in earlier dimensions for Matryoshka representation, acting as a "pre-processor" that optimally arranges the embedding space for both quantization and nesting.

Quantization Loss: Alongside these components, we employ a dedicated Quantization Loss that penalizes embedding values lying near threshold boundaries, thus reducing ambiguity in the quantization process. For example, in Table 6 at 2-bit, moving from *+ Trainable FFN Transform* (nDCG@10 = 0.2860) to *+ Quantization Loss* (0.3304) results in a 4.4-point absolute improvement, over 15% relative gain. By discouraging embeddings from clustering around threshold edges, the Quantization Loss ensures that discrete bins remain well-separated, minimizing quantization errors. This mechanism is pivotal for maintaining high performance under low-bit or hybrid quantization schemes.

Performance Degradation with Reduced Dimensions: While there is an expected decrease in performance when reducing the embedding dimensions, our approach mitigates this effect significantly. By concentrating the most informative features in the early dimensions, the model retains essential semantic relationships. This is evident from the modest performance drop between the higher and lower dimensions, demonstrating the robustness of our methods.

5.5 Storage Efficiency and Retrieval Speed

Table 7 compares the storage requirements of different quantization levels using 768-dimensional vectors using our proposed methods. Our methods offer substantial storage savings compared to full-precision embeddings. Moreover, the bitwise operations (`XOR` + `POPCOUNT`) accelerate retrieval, particularly for large-scale corpora.

Table 7. Storage comparison for different embedding formats (768-dimensional vector)

Format	Bits/dim	Vector Size	1M Vectors	Savings (%)	Relative Performance
FP32	32	3072 B	3.07 GB	0%	100%
FP16	16	1536 B	1.54 GB	50%	
Int8	8	768 B	768 MB	75%	
2-bit (3-bit expanded)	3	288 B	288 MB	90.6%	96.35%
1.5-bit (2-bit expanded)	2	192 B	192 MB	93.8%	89.73%
1-bit	1	96 B	96 MB	96.9%	80.74%
Hybrid (25% each level)	1.625	156 B	156 MB	94.9%	95.07%

The use of bitwise operations for similarity computation significantly boosts retrieval speed due to optimized CPU instructions. Our methods achieve faster retrieval times compared to traditional floating-point computations, making them suitable for real-time applications.

5.6 Visualization of Embeddings

We visualize the embeddings using t-SNE plots to illustrate how the structure is preserved after quantization. Figure 6 shows the embeddings from the MB model with 2-bit quantization.

Fig. 6. t-SNE visualization of MB embeddings with 2-bit quantization.

The visualization demonstrates that the quantized embeddings maintain the overall structure and grouping of the data, indicating effective preservation of semantic relationships.

5.7 Discussion

Our experiments confirm that QAMA effectively reduces storage requirements and improves retrieval speeds without significant loss in accuracy. The Matryoshka Representation Learning and the advanced quantization techniques contribute to maintaining high performance even with aggressive compression. The ablation studies highlight the importance of each component in our training framework. Notably, the Matryoshka Loss and the Orthogonality Regularization play crucial roles in enabling smaller embeddings to capture essential information. Furthermore, the Hybrid Quantization approach balances the trade-off between storage efficiency and performance by allocating higher precision to dimensions with more information content. This adaptive strategy ensures that critical semantic relationships are preserved.

6 CONCLUSION

Summarize your contributions and the significance of your findings in advancing the field of information retrieval.

REFERENCES

[1] Alexander Bondarenko, Maik Fröbe, Meriem Beloucif, Lukas Gienapp, Yamen Ajjour, Alexander Panchenko, Chris Biemann, Benno Stein, Henning Wachsmuth, Martin Potthast, and Matthias Hagen. 2020. Overview of Touché 2020: Argument Retrieval: Extended Abstract. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 11th International Conference of the CLEF Association, CLEF 2020, Thessaloniki, Greece, September 22–25, 2020, Proceedings* (Thessaloniki, Greece). Springer-Verlag, Berlin, Heidelberg, 384–395. https://doi.org/10.1007/978-3-030-58219-7_26

[2] Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. 2016. A Full-Text Learning to Rank Dataset for Medical Information Retrieval. *Proceedings of the 38th European Conference on Information Retrieval*. http://www.cl.uni-heidelberg.de/~riezler/publications/papers/ECIR2016.pdf

[3] Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. 2020. SPECTER: Document-level Representation Learning using Citation-informed Transformers. In *ACL*.

[4] Thomas Diggelmann, Jordan Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2020. CLIMATE-FEVER: A Dataset for Verification of Real-World Climate Claims. arXiv:2012.00614 [cs.CL]

[5] Faegheh Hasibi, Fedor Nikolaev, Chenyan Xiong, Krisztian Balog, Svein Erik Bratsberg, Alexander Kotov, and Jamie Callan. 2017. DBpedia-Entity v2: A Test Collection for Entity Search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Shinjuku, Tokyo, Japan) *(SIGIR '17)*. Association for Computing Machinery, New York, NY, USA, 1265–1268. https://doi.org/10.1145/3077136.3080751

[6] Doris Hoogeveen, Karin M. Verspoor, and Timothy Baldwin. 2015. CQADupStack: A Benchmark Data Set for Community Question-Answering Research. In *Proceedings of the 20th Australasian Document Computing Symposium (ADCS)* (Parramatta, NSW, Australia) *(ADCS '15)*. ACM, New York, NY, USA, Article 3, 8 pages. https://doi.org/10.1145/2838931.2838934

[7] Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. 2017. First Quora Dataset Release: Question Pairs. In *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, Perth, Australia.

[8] Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. 2022. Matryoshka Representation Learning. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 30233–30249. https://proceedings.neurips.cc/paper_files/paper/2022/file/c32319f4868da7613d78af9993100e42-Paper-Conference.pdf

[9] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics* 7 (2019), 452–466. https://doi.org/10.1162/tacl_a_00276

[10] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. MTEB: Massive Text Embedding Benchmark. *arXiv preprint arXiv:2210.07316* (2022). https://doi.org/10.48550/ARXIV.2210.07316

[11] 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. *CoRR* abs/1611.09268 (2016). arXiv:1611.09268 http://arxiv.org/abs/1611.09268

[12] Zach Nussbaum, John X. Morris, Brandon Duderstadt, and Andriy Mulyar. 2024. Nomic Embed: Training a Reproducible Long Context Text Embedder. arXiv:2402.01613 [cs.CL]

[13] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. http://arxiv.org/abs/1908.10084

[14] Kirk Roberts, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, Kyle Lo, Ian Soboroff, Ellen M. Voorhees, Lucy Lu Wang, and William R. Hersh. 2021. Searching for Scientific Evidence in a Pandemic: An Overview of TREC-COVID. *CoRR* abs/2104.09632 (2021). arXiv:2104.09632 https://arxiv.org/abs/2104.09632

[15] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. https://openreview.net/forum?id=wCu6T5xFjeJ

[16] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a Large-scale Dataset for Fact Extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, Marilyn Walker, Heng Ji, and Amanda Stent (Eds.). Association for Computational Linguistics, New Orleans, Louisiana, 809–819. https://doi.org/10.18653/v1/N18-1074

[17] Henning Wachsmuth, Shahbaz Syed, and Benno Stein. 2018. Retrieval of the Best Counterargument without Prior Topic Knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Iryna Gurevych and Yusuke Miyao (Eds.). Association for Computational Linguistics, Melbourne, Australia, 241–251. https://doi.org/10.18653/v1/P18-1023

[18] David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or Fiction: Verifying Scientific Claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 7534–7550. https://doi.org/10.18653/v1/2020.emnlp-main.609

[19] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MINILM: deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) *(NIPS '20)*. Curran Associates Inc., Red Hook, NY, USA, Article 485, 13 pages.

[20] Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. 2024. Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference. arXiv:2412.13663 [cs.CL] https://arxiv.org/abs/2412.13663

[21] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.). Association for Computational Linguistics, Brussels, Belgium, 2369–2380. https://doi.org/10.18653/v1/D18-1259

A RESEARCH METHODS

A.1 Part One

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi malesuada, quam in pulvinar varius, metus nunc fermentum urna, id sollicitudin purus odio sit amet enim. Aliquam ullamcorper eu ipsum vel mollis. Curabitur quis dictum nisl. Phasellus vel semper risus, et lacinia dolor. Integer ultricies commodo sem nec semper.

A.2 Part Two

Etiam commodo feugiat nisl pulvinar pellentesque. Etiam auctor sodales ligula, non varius nibh pulvinar semper. Suspendisse nec lectus non ipsum convallis congue hendrerit vitae sapien. Donec at laoreet eros. Vivamus non purus placerat, scelerisque diam eu, cursus ante. Etiam aliquam tortor auctor efficitur mattis.

B ONLINE RESOURCES

Nam id fermentum dui. Suspendisse sagittis tortor a nulla mollis, in pulvinar ex pretium. Sed interdum orci quis metus euismod, et sagittis enim maximus. Vestibulum gravida massa ut felis suscipit congue. Quisque mattis elit a risus ultrices commodo venenatis eget dui. Etiam sagittis eleifend elementum.

Nam interdum magna at lectus dignissim, ac dignissim lorem rhoncus. Maecenas eu arcu ac neque placerat aliquam. Nunc pulvinar massa et mattis lacinia.