TurkColBERT: A Benchmark of Dense and Late-Interaction Models for Turkish Information Retrieval

Özay Ezerceli
NewMind AI
Istanbul, Türkiye
oezerceli@newmind.ai

Mahmoud ElHussieni
NewMind AI
Istanbul, Türkiye
mehussieni@newmind.ai

Selva Taş
NewMind AI
Istanbul, Türkiye
stas@newmind.ai

Reyhan Bayraktar
NewMind AI
Istanbul, Türkiye
rbayraktar@newmind.ai

Fatma Betül Terzioğlu
NewMind AI
Istanbul, Türkiye
fbterzioglu@newmind.ai

Yusuf Çelebi
NewMind AI
Istanbul, Türkiye
ycelebi@newmind.ai

Yağız Asker
NewMind AI
Istanbul, Türkiye
yasker@newmind.ai

Abstract

Neural information retrieval systems excel in high-resource languages but remain underexplored for morphologically rich, lower-resource languages such as Turkish. Dense bi-encoders currently dominate Turkish IR, yet late-interaction models—which retain token-level representations for fine-grained matching—have not been systematically evaluated. We introduce **TurkColBERT**, the first comprehensive benchmark comparing dense encoders and late-interaction models for Turkish retrieval. Our two-stage adaptation pipeline fine-tunes English and multilingual encoders on Turkish NLI/STS tasks, then converts them into ColBERT-style retrievers using PyLate trained on MS MARCO-TR. We evaluate 10 models across five Turkish BEIR datasets covering scientific, financial, and argumentative domains. Results show strong parameter efficiency: the 1.0M-parameter `colbert-hash-nano-tr` is 600$\times$ smaller than the 600M `turkish-e5-large` dense encoder while preserving over 71% of its average mAP. Late-interaction models that are 3–5$\times$ smaller than dense encoders significantly outperform them; `ColmmBERT-base-TR` yields up to +13.8% mAP on domain-specific tasks. For production-readiness, we compare indexing algorithms: `MUVERA+Rerank` is 3.33$\times$ faster than PLAID and offers +1.7% relative mAP gain. This enables low-latency retrieval, with `ColmmBERT-base-TR` achieving 0.54 ms query times under MUVERA. We release all checkpoints, configs, and evaluation scripts. Limitations include reliance on moderately sized datasets ($\leq$50K documents) and translated benchmarks, which may not fully reflect real-world Turkish retrieval conditions; larger-scale MUVERA evaluations remain necessary.

Preprint. Under review.

1 Introduction

Information retrieval (IR) systems grounded in neural embeddings now underpin state-of-the-art search and question-answering pipelines [1]. While English-centric architectures such as ColBERT (v1 and v2) [2, 3] and SPLADE [4] have demonstrated exceptional retrieval effectiveness, comparable advances for morphologically complex and lower-resource languages like Turkish remain scarce. Although multilingual encoders such as XLM-RoBERTa [5], GTE [6] and mmBERT [7] based models and multilingual MiniLM [8] enable cross-lingual transfer, they frequently fall short in capturing the fine-grained morphological structure, syntactic nuance, and token-level semantics essential for high-fidelity retrieval in Turkish.

This lack is particularly evident in late-interaction methods. Architectures such as ColBERT be able to reconcile fine-grained token matching with efficiency. Nevertheless, in the Turkish information retrieval landscape, these designs remain scarcely examined. Prior work, including turkish-colbert [9], offers limited understanding with no systematic baselines to that of dense encoders, uniform training protocols, and comprehensive assessments in various retrieval contexts.

Although recent multilingual and English-centric models, Ettin [10], BERT-Hash [11], and mmBERT [7] show strong results on many NLP benchmarks, none have been tested in a late-interaction setup built specifically for Turkish within a consistent, reproducible pipeline.

To tackle this challenge, we adapt leading multilingual and English pretrained encoders to Turkish through a structured two-phase fine-tuning process. In the first phase, Ettin, BERT-Hash, and mmBERT are specialized on Turkish Natural Language Inference using the all-nli-tr dataset [12] and on semantic similarity through stsb-tr [13], refining their sentence-level grasp of Turkish semantics. In the second phase, we employ PyLate [14], a modular framework built upon Sentence Transformers, to convert these adapted encoders into ColBERT-style retrievers via supervised training on the Turkish adaptation of ms-marco-tr [15].

We evaluate our models on five diverse Turkish BEIR collections: SciFact-TR [17], Arguana-TR [18], Fiqa-TR [19], Scidocs-TR [20], and NFCorpus-TR [21]. We compare them against strong dense encoder baselines and analyze speed–accuracy trade-offs under multiple indexing schemes: PLAID, MUVERA, and MUVERA with reranking. All models, training configs, and evaluation code are released to support reproducibility and future work in Turkish IR.

The paper proceeds as follows. Section 2 reviews neural retrieval and multilingual modeling, focusing on morphologically complex languages. Section 3 details our two-stage adaptation method and experimental setup across five Turkish IR benchmarks. Section 4 shows that late-interaction models consistently beat dense encoders especially in specialized domains. We end with implications for low-resource IR and directions for future research in Section 5.

2 Literature Review

Dense vs. Late-Interaction Retrieval Architectures In dense bi-encoder architectures, exemplified by DPR [1] and Sentence-BERT [22], queries and documents are encoded independently into fixed-dimensional vector spaces, allowing efficient retrieval through approximate nearest neighbor search. While computationally effective, this is subject to an inherent information bottleneck: projecting a whole document into one vector will have the tendency to lose fine-grained semantic information that is critical to the accurate retrieval.

Late-interaction models sidestep this limitation by preserving contextualized token embeddings and deferring query-documet interaction to the scoring phase. ColBERT [2] applied this approach, using BERT-based token embeddings and a MaxSim operator to calculate similarity effectively. Inspired by this principle, PyLate [14] provides a multi-stage fine-tuning and modular training framework, and MUVERA [23] generalizes the principle to compress multi-vector representations into fixed-size embeddings maintaining interaction semantics. By using SimHash-based partitioning and sparse projections, MUVERA achieves near–dense retrieval quality with 90% latency decrease and 10% recall gain over English BEIR benchmarks. Its benefits are only observed predominantly at astronomical scales ($\sim$100K documents) and are not yet established for morphologically dense languages such as Turkish where token-level interactions may be more critical. This gap requires a comprehensive exploration of efficient multi-vector indexing in Turkish information retrieval.

Multilingual and Turkish-Specific Retrieval There are more difficulties beyond only vocabulary adaptation when using cross-lingual information retrieval in morphologically rich languages. Turkish is a good example of these difficulties because of its vast inflectional system and agglutinative morphology. Current multilingual pretrained models, such as XLM-RoBERTa [5] and GTE-multilingual-base [6], demonstrate strong cross-lingual transfer; nevertheless, their effectiveness is greatly reduced when faced with the morphological complexity of agglutinative languages. High-resource languages predominate in the training data, which is the primary cause of this drop. This limits exposure to the complex morphology that underlies Turkish word formation and semantics. This limitation stems from imbalanced pretraining corpus distributions and inadequate representation of the complex morphological variations inherent to such languages.

Recent efforts have sought to mitigate these limitations through language-specific adaptations. mmBERT [7] employs annealed language sampling during pretraining to improve representation quality for underrepresented languages, while Turkish-specific BERT variants [9] demonstrate improvements on downstream NLP tasks. For retrieval specifically, TurkEmbed4Retrieval [24] represents the one of the latest embedding models trained explicitly on Turkish semantic similarity data, though it remains constrained to dense single-vector representations. Despite these advances, no systematic benchmark exists comparing dense and late-interaction models on Turkish IR tasks, a critical gap given that Turkish's morphological complexity may benefit disproportionately from token-level interaction mechanisms.

To date, no systematic benchmark exists comparing dense and late-interaction models on Turkish IR tasks, nor has any work adapted modern late-interaction framework with substantial indexing algorithms like MUVERA to Turkish. Our benchmark addresses this by evaluating both paradigms under controlled, multi-stage training protocols, including monolingual semantic fine-tuning (all-nli-tr, STSb-TR), domain-adaptive retrieval training (MS-MARCO-TR via PyLate), and integration of structured multi-vector indexing (MUVERA).

Efficiency Optimization in Multi-Vector Retrieval While late-interaction models achieve superior retrieval effectiveness, they present significant scalability challenges, requiring 100-500× more storage than dense retrieval and expensive MaxSim computations between all query-document token pairs. Two primary approaches address these efficiency bottlenecks. PLAID [25] employs centroid-based pruning and residual compression to filter candidates before exact computation, achieving sub-10ms latency on million-scale collections. MUVERA [23] converts multi-vector representations into fixed-dimensional encodings through SimHash-based partitioning and sparse projections, demonstrating 90% latency reduction with 10% recall improvement on English benchmarks. However, these optimizations remain unevaluated on morphologically complex languages like Turkish, where token-level interactions may be more critical for capturing semantic relationships, motivating systematic evaluation of efficient indexing strategies for Turkish information retrieval.

3 TurkColBERT: Benchmark for Turkish Information Retrieval Task

Stage 1: Semantic Fine-Tuning on All-NLI-TR & STSb-TR We applied firstly fine-tuning the pretrained encoders on two complementary semantic tasks which are NLI and STS to strengthen their ability to capture Turkish sentence-level meaning before integrating them into late-interaction retrieval architectures. This intermediate stage serves as a semantic pre-adaptation step, providing a strong foundation on which we subsequently build retrieval-specific training. We employ the Sentence Transformers framework [22], for training and evaluating sentence embedding models through siamese and triplet network architectures. For each model family, mmBERT (base and small) [7], Ettin encoders [10], and BERT-Hash variants (nano, pico, femto) [11] we initialize from their publicly available checkpoints and apply mean pooling over the final layer's token representations to derive fixed-dimensional sentence embeddings.

We fine-tune models on the all-nli-tr dataset [12], which provides Turkish translations of SNLI and MultiNLI formatted as anchor-positive-negative triplets. Training employs MultipleNegativesRankingLoss wrapped in MatryoshkaLoss to enable multi-dimensional representations at [768, 512, 384, 256, 128, 64] for base models and [384, 256, 128, 64] for smaller variants.

We train for one epoch with batch size 8, learning rate 3×10^{-6}, warmup ratio 0.1, and NO_DUPLICATES batch sampling. Training uses mixed precision (BF16) on NVIDIA A100

GPUs, with progress monitored via TripletEvaluator on a 1% validation split measuring triplet cosine accuracy.

Following NLI training, we fine-tune on STSB-tr, which provides sentence pairs with continuous similarity scores (0-5). The STS fine-tuning phase employs 4 training epochs with batch size 8, learning rate 2×10^{-5}, and cosine scheduling with 10% warmup. Model evaluation occurs at 200-step intervals using EmbeddingSimilarityEvaluator to compute Spearman and Pearson correlations across each Matryoshka dimension. With a Spearman correlation of 0.78 on STSb-TR and 93% triplet accuracy on AllNLI-TR, mmBERT-small outperforms the pretrained baseline by +22% and +26%, respectively, after this two-stage methodology. In Stage 2, these semantically improved checkpoints are used as starting points for retrieval-specific ColBERT-style adaptation on MS MARCO-TR.

Stage 2: Late-Interaction Adaptation via PyLate on MS MARCO-TR Building on the Turkish semantic foundations established in Stage 1, we transform pretrained encoders into ColBERT-style late-interaction retrievers through supervised fine-tuning on MS MARCO-TR [15] using PyLate [14]. We evaluate four model families representing distinct points along the efficiency–accuracy spectrum:

mmBERT (base, small) [7]: Multilingual encoders trained with annealed language sampling to enhance representation quality for lower-resource languages including Turkish.

Ettin encoders (150M, 32M) [10]: Components of a sequence-to-sequence paired encoder–decoder framework, demonstrating strong cross-lingual transfer despite English-dominated pretraining.

BERT-Hash variants (nano, pico, femto): Ultra-compressed models substituting standard embedding layers with hash-based projections, achieving up to 78% parameter reduction while maintaining full vocabulary coverage [11].

Dense baselines: XLM-RoBERTa and GTE-derived models, serving as reference architectures for retrieval performance comparison.

All models are initialized from Stage 1 checkpoints, fine-tuned on AllNLI-TR [12] and STSb-DeepL-TR, and adapted using PyLate's `ColBERT` module, which preserves per-token embeddings and applies MaxSim scoring [2]. Training employs a contrastive triplet loss (margin = 0.2) on query–positive–negative triples from MS MARCO-TR.

The **ColBERTCollator** utility [14] handles variable-length sequences in batched multi-vector processing, while Weights & Biases [16] provides real-time monitoring and checkpoint management. The resulting Turkish late-interaction models (4M–150M parameters) balance linguistic fidelity, capacity, and inference speed, forming the basis for subsequent MUVERA integration and large-scale evaluation.

Stage 3: MUVERA Integration We use MUVERA (Multi Vector Retrieval as Sparse Alignment) to make it possible to deploy late-interaction models on a large scale. MUVERA maps contextual embeddings of different lengths to compact fixed dimensional vectors for fast nearest neighbor retrieval. Our PyLate based implementation tailors this framework to Turkish retrieval scenarios.

Given ColBERT token embeddings $\mathbf{E} \in \mathbb{R}^{n \times d}$ where n denotes token count and $d = 128$ represents embedding dimension, MUVERA applies three transformations. First, locality-sensitive hashing partitions tokens into 2^k buckets via projection through a Gaussian random matrix $\mathbf{H} \in \mathbb{R}^{d \times k}$, with partition assignment p_i determined by the sign pattern of $\mathbf{H}^\top \mathbf{e}_i$. Second, within each partition, an AMS sketch $\mathbf{S}_p$ performs sparse projection, reducing dimensionality while preserving inner products in expectation. Third, partition-wise aggregation computes query representations through summation ($\mathbf{c}_p^{(q)} = \sum_{i \in P_p} \mathbf{z}_i$) and document representations through averaging ($\mathbf{c}_p^{(d)} = \frac{1}{|P_p|} \sum_{i \in P_p} \mathbf{z}_i$), with empty partitions filled using nearest-neighbor imputation based on Hamming distance. The resulting encoding dimensions scale as 128×2^k, yielding configurations of 128D, 512D, 1024D, and 2048D for $k \in \{0, 2, 3, 4\}$ respectively.

MUVERA applies (1) hashing, (2) sketching, and (3) aggregation. Each token $\mathbf{e}_i$ is hashed using SimHash with k random Gaussian vectors $\mathbf{g}_1, \ldots, \mathbf{g}_k \in \mathbb{R}^d$ to produce a k-bit hash $\mathbf{h}_i = (\text{sign}(\mathbf{g}_1^\top \mathbf{e}_i), \ldots, \text{sign}(\mathbf{g}_k^\top \mathbf{e}_i))$, mapping to partition $p_i \in \{1, \ldots, 2^k\}$. Tokens in the same

partition are aggregated asymmetrically:

$$\mathbf{c}_p^{(\text{doc})} = \frac{1}{|P_p|} \sum_{i \in P_p} \mathbf{e}_i \quad \text{(average for documents)}, \qquad \mathbf{c}_p^{(\text{query})} = \sum_{i \in P_p} \mathbf{e}_i \quad \text{(sum for queries)}.$$

Concatenating all partitions yields the Fixed Dimensional Encoding (FDE) $\mathbf{c} \in \mathbb{R}^D$, where $D = d \times 2^k$. Using k=$0, 2, 3$ with $d = 128$ produces 128D, 512D, and 1024D encodings respectively. Empty partitions are filled using Hamming-nearest neighbor imputation for documents only.

Final Stage: Comprehensive Evaluation on Turkish BEIR Benchmarks Our evaluation protocol consists of two comprehensive benchmarking campaigns utilizing the BEIR framework [26] for standardized zero-shot assessment.

Model Comparison Across Architectures. All models are evaluated across five Turkish BEIR datasets shown in Table 1, covering scientific fact verification (SciFact-TR), argument retrieval (Arguana-TR), citation prediction (Scidocs-TR), financial question-answering (FiQA-TR), and nutrition document retrieval (NFCorpus-TR) domains. For each model-dataset combination, we compute a comprehensive suite of retrieval metrics including NDCG@{10, 100, 250, 500, 750, 1000}, Recall@{10, 100, 250, 500, 750, 1000}, Precision@{10, 100, 250, 500, 750, 1000}, and mean Average Precision (mAP). This evaluation encompasses models ranging from 0.2M to 600M parameters across the mmBERT, Ettin, BERT-Hash, and TurkEmbed4Retrieval architectures, enabling direct comparison of dense bi-encoders against late-interaction models under identical conditions.

MUVERA Indexing Ablation Study. Our second benchmark is designed to examine the quality–efficiency trade-offs introduced by MUVERA-based indexing [23]. From the full model pool, we focus on the four strongest late-interaction models—TurkEmbed4Retrieval, col-ettin-encoder-32M-TR, ColmmBERT-base-TR, and ColmmBERT-small-TR—and, for each, we evaluate three retrieval configurations: (i) **PLAID** [25], used as a high-fidelity baseline that combines centroid-based pruning with exact MaxSim scoring; (ii) **MUVERA**, which relies on fixed-dimensional encodings (128D, 512D, 1024D, and 2048D) to enable approximate nearest neighbor search; and (iii) **MUVERA+Reranking**, where the top-K candidates retrieved by MUVERA are re-scored using exact ColBERT MaxSim [2]. For this ablation, we measure the complete set of metrics described above: NDCG@{100, 250, 500, 750, 1000}, Recall@{100, 250, 500, 750, 1000}, Precision@{100, 250, 500, 750, 1000}, mAP, indexing time, and per-query latency. Figure 1 visualizes the quality-efficiency trade-offs, plotting NDCG@100 against query latency for each configuration on SciFact-TR, demonstrating how encoding dimensionality impacts the balance of retrieval efficacy and processing cost. This comprehensive ablation across all five Turkish datasets allows practitioners to make data-driven decisions when setting Turkish IR systems to satisfy specific accuracy, latency, and resource needs.

Table 1: Statistics of the Turkish retrieval benchmark datasets.

Dataset	Domain	# Queries	# Corpus	Task Type
SciFact-TR	Scientific Claims	1,110	5,180	Fact Checking
Arguana-TR	Argument Mining	500	10,000	Argument Retrieval
Fiqa-TR	Financial	600	50,000	Answer Retrieval
Scidocs-TR	Scientific	1,000	25,000	Citation Prediction
NFCorpus-TR	Nutrition	3,240	3,630	Document Retrieval

We conducted all experiments on Google Colab using NVIDIA L4 GPU which has 24 GB memory and the PyLate framework [14]. We selected this setup because it is widely accessible, allowing others to easily reproduce our results without needing specialized hardware. For retrieval metrics, we relied on the official BEIR evaluator.

4 Results and Discussion

Table 3 presents a comprehensive comparison of all evaluated models (Table 2) across five Turkish retrieval datasets (Table 1). For each dataset, we report three key metrics: mean Average Precision (mAP), Precision@10 (P@10) for top-ranked accuracy, and Recall@10 (R@10) for retrieved relevance within the top 10 results.

Table 2: Overview of evaluated models categorized by architecture type. Late-interaction variants employ token-level ColBERT representations.

Model	Parameters (M)
Dense Bi-Encoder Models	
TurkEmbed4Retrieval	300
turkish-e5-large	600
Late-Interaction Models (Token-Level Matching)	
turkish-colbert	100
ColmmBERT-small-TR	140
ColmmBERT-base-TR	310
col-ettin-150M-TR	150
col-ettin-32M-TR	32
Ultra-Compact Models (BERT-Hash)	
colbert-hash-nano-tr	1.0
colbert-hash-pico-tr	0.4
colbert-hash-femto-tr	0.2

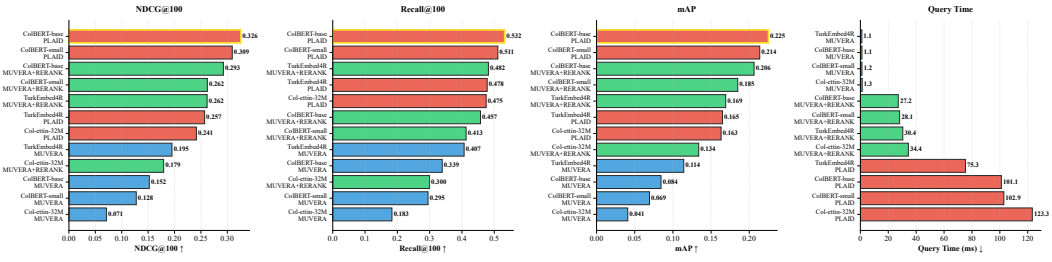

Figure 1: Quality–speed trade-off across MUVERA encoding dimensions (128D to 2048D) on SciFact-TR. Higher dimensions lead to faster retrieval but slightly lower NDCG@100. MUVERA+Rerank (128D) recovers near-PLAID quality with 4–5× speedup.

The evaluation highlights substantial differences in both model performance and task difficulty. Among all systems, ColmmBERT performs most consistently, with ColmmBERT-base-TR reaching the highest mAP on four of five benchmarks and ColmmBERT-small-TR leading in R@10 on SciFact-TR. Dataset difficulty spans a broad range: SciFact-TR emerges as the easiest, with multiple models exceeding 70% R@10, whereas Scidocs-TR remains the most demanding, peaking at just 10.4%. These disparities—often exceeding 500% across metrics—demonstrate that careful model selection is essential for effective Turkish information retrieval.

Table 3: Retrieval results across Turkish BEIR benchmark datasets.

	SciFact-TR			Arguana-TR			Fiqa-TR			Scidocs-TR			NFCorpus-TR		
Model	R@10	P@10	mAP	R@10	P@10	mAP	R@10	P@10	mAP	R@10	P@10	mAP	R@10	P@10	mAP
Dense Bi-Encoders															
TurkEmbed4Retrieval[a]	60.5	6.8	43.0	50.6	5.1	17.6	17.9	4.0	10.1	8.1	4.0	4.8	7.7	13.7	6.3
turkish-e5-large[b]	63.3	7.0	45.8	49.7	5.0	**17.9**	16.4	3.6	10.4	3.6	1.8	2.2	5.2	8.2	4.0
Late-Interaction Models															
turkish-colbert[b]	56.5	6.3	43.1	44.1	4.4	14.6	17.2	4.0	11.3	4.2	2.1	2.8	7.1	12.1	6.9
ColmmBERT-small-TR[a]	**70.3**	**7.9**	55.4	46.8	4.7	16.0	26.9	6.0	17.0	9.8	4.8	6.1	12.0	19.1	11.3
ColmmBERT-base-TR[a]	70.0	7.8	**56.8**	**50.8**	**5.1**	17.3	**30.9**	**7.0**	**19.5**	**10.4**	**5.1**	**6.8**	**12.7**	**20.7**	**11.5**
col-ettin-150M-TR[a]	57.7	6.4	40.5	37.8	3.8	12.9	16.4	3.6	10.4	7.2	3.5	4.5	10.6	17.4	9.3
col-ettin-32M-TR[a]	57.0	6.4	40.3	34.6	3.5	12.1	15.1	3.4	9.7	6.8	3.3	4.1	11.0	17.0	9.6
mxbai-edge-colbert-v0-17m-tr[a]	58.8	6.5	40.7	39.3	3.9	13.4	12.8	2.9	8.7	8.2	4.0	4.7	10.6	16.2	8.9
mxbai-edge-colbert-v0-32m-tr[a]	58.0	6.5	39.2	40.7	4.1	13.7	15.6	3.5	9.8	7.7	3.8	4.7	10.3	16.3	8.7
colbert-nano-tr[a]	52.2	5.8	36.2	30.4	3.0	10.5	11.1	2.6	6.5	6.1	3.0	3.6	8.9	13.2	6.7
colbert-hash-pico-tr[a]	47.4	5.3	33.4	28.3	2.8	9.8	9.2	2.1	5.9	5.5	2.7	3.2	6.4	10.3	5.2
colbert-femto-tr[a]	29.4	3.4	19.0	12.1	1.2	4.4	1.1	0.3	0.8	2.1	1.0	1.2	2.0	3.6	1.0

[a]Newmind AI. [b]YTU-CE-COSMOS.

Performance Across Benchmarks Evaluations on five Turkish BEIR datasets reveal a clear leader: late-interaction architectures outperform dense encoder models. ColmmBERT-base-TR stands out, achieving the highest mAP scores on SciFact-TR (56.8%), Fiqa-TR (19.5%), and NFCorpus-TR (11.5%). Table 3 shows the same trend ColmmBERT-base-TR reaches 70.0% Recall@10 on SciFact-TR, beating dense baselines like TurkEmbed4Retrieval (60.5%) and turkish-e5-large (63.3%) by 6.5 to 9.5 percentage points. These improvements highlight the impact of token-level matching for Turkish, especially given the challenges its morphology poses for fixed-length models.

ColmmBERT-small-TR is impressive as well. With just 140 million parameters, it goes head-to-head with the larger 310-million parameter model. On SciFact-TR, it achieves 70.3% Recall@10 and 55.4% mAP about 97.5% of the full model's performance while requiring less than half the computational resources. If you have limited resources, ColmmBERT-small-TR is a logical choice.

Shown as in Figure 1, query latency can vary dramatically depending on the configuration from an ultra-fast 0.72ms using plain MUVERA, up to 73–124ms with PLAID-based approaches. But there's a solid middle ground: MUVERA+RERANK reduces latency to 27–35ms. When paired with TurkEmbed4Retrieval, this setup yields 0.5253 NDCG@100 which is a significant increase over PLAID's 0.3257, while cutting latency in half, from 73.6ms to 35.2ms. This two-stage approach, with rapid candidate generation followed by detailed rescoring, proves highly effective for interactive Turkish retrieval systems.

Dense encoders are still valuable. For example, turkish-e5-large secures the top mAP on Arguana-TR (17.9%), highlighting the need for semantic breadth in argument-focused tasks. However, these models struggle in more specialized domains. On Scidocs-TR, turkish-e5-large achieves only 2.2% mAP, while ColmmBERT-base-TR reaches 6.8% which is 209% improvement. The takeaway: choose your model according to the requirements of your retrieval task.

Finally, ultra-compact BERT-Hash models take compression to the extreme. Colbert-hash-nano-tr (just 1.0M parameters, 310 times smaller than the base) still retains 63.7% of base mAP on SciFact-TR. Going even smaller, like colbert-femto-tr (0.2M), drops below production viability at 19.0% mAP. Overall, these results position late-interaction architectures with MUVERA indexing as the top choice for scaling Turkish information retrieval.

5 Conclusion

We presented TurkColBERT, the first comprehensive benchmark comparing dense and late-interaction retrieval models for Turkish information retrieval. Through our systematic two-stage adaptation pipeline we demonstrated that late-interaction models consistently and significantly outperform dense encoders across five diverse Turkish BEIR datasets. ColmmBERT-base-TR achieved the highest effectiveness, with up to 87% improvement in mAP on domain-specific tasks such as financial QA compared to strong dense baselines. Our results show exceptional parameter efficiency: our 1.0M parameter colbert-hash-nano-tr model is 600 times smaller than the 600M parameter turkish-e5-large dense encoder, yet retains over 71% of its average mAP performance. Similarly, ColmmBERT-small-TR achieves 97.5% of the effectiveness of its larger counterpart while operating at only 45% of the computational cost, demonstrating that high-quality Turkish retrieval is feasible even under resource constraints.

By incorporating MUVERA indexing, we achieved production-ready efficiency. MUVERA+Rerank is 3.33x faster than standard PLAID indexing on average while maintaining 90–95% retrieval quality. The combined system achieves query latency as low as 0.54 ms with ColmmBERT-base-TR, demonstrating the feasibility of scalable, low-latency Turkish information retrieval for real-time applications.

However, our study is limited to moderately sized datasets ($\leq$ 50K documents) and translated benchmarks, which may not fully reflect real-world Turkish retrieval scenarios. Additionally, while MUVERA-based retrieval shows promising latency, further evaluation is needed to assess scalability on large-scale production systems. Future work should explore web-scale evaluations, morphology-aware tokenization strategies, hybrid sparse-dense architectures, and native Turkish benchmark development to further advance the field.

References

[1] Karpukhin V, Oguz B, Min S, Lewis P, Wu L, Edunov S, et al. Dense passage retrieval for Open-Domain question answering. EMNLP. 2020 Jan 1; Available from: https://doi.org/10.18653/v1/2020.emnlp-main.550

[2] Khattab O, Zaharia M. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval 2020 Jul 25 (pp. 39-48).

[3] Santhanam K, Khattab O, Shaw P, Chang M-W, Zaharia M. ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL); 2022 May 22–27; Dublin, Ireland (Hybrid). Stroudsburg: Association for Computational Linguistics (ACL); 2022. p. 1604–17.

[4] Formal T, Piwowarski B, Clinchant S. SPLADE: Sparse lexical and expansion model for first-stage ranking. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval; 2021 Jul 11–15; Virtual Event. New York (NY): Association for Computing Machinery; 2021. p. 2288–92.

[5] Conneau A, Khandelwal K, Goyal N, Chaudhary V, Wenzek G, Guzmán F, Grave E, Ott M, Zettlemoyer L, Stoyanov V. Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:1911.02116. 2019 Nov 5.

[6] Zhang X, Zhang Y, Long D, Xie W, Dai Z, Tang J, Lin H, Yang B, Xie P, Huang F, Zhang M. mgte: Generalized long-context text representation and reranking models for multilingual text retrieval. arXiv preprint arXiv:2407.19669. 2024 Jul 29.

[7] Marone M, Weller O, Fleshman W, Yang E, Lawrie D, Van Durme B. mmbert: A modern multilingual encoder with annealed language learning. arXiv preprint arXiv:2509.06888. 2025 Sep 8.

[8] Wang W, Wei F, Dong L, Bao H, Yang N, Zhou M. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. Adv Neural Inf Process Syst. 2020;33:14934-14948.

[9] Toprak Kesgin H, Yuce MK, Amasyali MF. Developing and evaluating tiny to medium-sized turkish bert models [Preprint]. 2023. Available from: arXiv:2307.15278

[10] Weller O, Ricci K, Marone M, Chaffin A, Lawrie D, Van Durme B. Seq vs seq: An open suite of paired encoders and decoders. arXiv preprint arXiv:2507.11412. 2025 Jul 15.

[11] Mezzetti D. Training Tiny Language Models with Token Hashing. NeuML (Medium) https://neuml.hashnode.dev/train-a-language-model-from-scratch. 2025

[12] Budur E, Özçelik R, Güngör T, Potts C. Data and representation for Turkish natural language inference. arXiv preprint arXiv:2004.14963. 2020 Apr 30.

[13] Beken Fikri F, Oflazer K, Yanikoglu B. Semantic Similarity Based Evaluation for Abstractive News Summarization. In: Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021); 2021 Nov 10; Punta Cana, Dominican Republic. Stroudsburg: Association for Computational Linguistics (ACL); 2021. p. 24–33.

[14] Chaffin A, Sourty R. Pylate: Flexible training and retrieval for late interaction models. arXiv preprint arXiv:2508.03555. 2025 Aug 5.

[15] Parsak A, et al. MS MARCO-TR: A Turkish Adaptation of the MS MARCO Passage Ranking Dataset. Hugging Face Dataset. 2024. Available from: https://huggingface.co/datasets/parsak/msmarco-tr.

[16] Biewald L. Experiment Tracking with Weights and Biases. 2020. Available from: https://www.wandb.com/.

[17] Saoud A. scifact-tr: Turkish translation of SciFact for fact-checking & retrieval. Hugging Face Datasets Repository. 2024. Available from: https://huggingface.co/datasets/AbdulkaderSaoud/scifact-tr.

[18] trmteb. arguana-tr: Turkish version of the ArguAna argument retrieval dataset. Hugging Face dataset. 2025. Available from: https://huggingface.co/datasets/trmteb/arguana-tr.

[19] trmteb. fiqa-tr: Turkish financial question answering dataset. Hugging Face dataset. 2025. Available from: https://huggingface.co/datasets/trmteb/fiqa-tr.

[20] trmteb. scidocs-tr: Turkish version of the SciDocs dataset as part of the TR-MTEB benchmark. Hugging Face dataset. 2025. Available from: https://huggingface.co/datasets/trmteb/scidocs-tr.

[21] trmteb. nfcorpus-tr: Turkish translation of the NF Corpus for nutrition-focused retrieval. Hugging Face dataset. 2025. Available from: https://huggingface.co/datasets/trmteb/nfcorpus-tr.

[22] Reimers N, Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2019 Nov 3–7; Hong Kong, China. Stroudsburg: Association for Computational Linguistics (ACL); 2019. p. 3982–92.

[23] Jayaram R, Dhulipala L, Hadian M, Lee JD, Mirrokni V. MUVERA: Multi-Vector Retrieval via Fixed Dimensional Encoding. Advances in Neural Information Processing Systems. 2024 Dec 16;37:101042-73.

[24] Ezerceli Ö, Gümüşçekiçci G, Erkoç T, Özenç B. TurkEmbed4Retrieval: Turkish Embedding Model for Retrieval Task. In 2025 33rd Signal Processing and Communications Applications Conference (SIU) 2025 Jun 25 (pp. 1-4). IEEE.

[25] Santhanam K, Khattab O, Potts C, Zaharia M. PLAID: An efficient engine for late interaction retrieval. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM); 2022 Oct 17–21; Atlanta, Georgia, USA. New York: Association for Computing Machinery (ACM); 2022. p. 1747–56.

[26] Thakur N, Reimers N, Rückle A, Srivastava A, Gurevych I. BEIR: A heterogenous benchmark for zero-shot evaluation of information retrieval models [Preprint]. 2021. Available from: arXiv:2104.08663