Adapting Pretrained Embedding Models to Turkish via Token Remapping and Distillation

M. Ali Bayram
Department of Computer Engineering
Yıldız Technical University
Istanbul, Turkey
ali.bayram@std.yildiz.edu.tr

Abstract

Sentence embeddings are a foundational component for semantic search, clustering, classification with embedding-based features, and retrieval-augmented generation. This paper presents *embeddingmagibu-152m*, a Turkish-focused sentence embedding model that produces 768-dimensional ℓ_2-normalized vectors and supports an extended 2,048-token context window, exceeding the 512-token limit common in earlier BERT-based Turkish encoders. Instead of full pretraining, an efficient three-stage adaptation pipeline is introduced: (1) train a Turkish SentencePiece BPE tokenizer with a $2^{16} = 65{,}536$ vocabulary, (2) clone a teacher embedding model while preserving transformer backbone weights and initializing a compatible embedding table for the new vocabulary via token-id mapping and mean composition, and (3) perform offline embedding distillation from a released dataset of precomputed teacher vectors using a cosine objective. The resulting student model contains 152M parameters and trains in approximately four hours on a single A100 GPU by avoiding online teacher inference during training. Empirically, Pearson/Spearman correlations of 0.7512/0.7305 are obtained on STSbTR, surpassing the teacher model (0.7391/0.7194). On TR-MTEB, mean scores of 69.68% on the 15-task subset and 62.57% on the full 24-task benchmark are reported, providing a competitive efficiency–quality trade-off relative to substantially larger multilingual baselines. To facilitate reproducibility and downstream use, artifacts are released including model weights and tokenizer files, exported evaluation results, an interactive demo, a benchmark results explorer, and accompanying cloning and distillation tooling.

1 Introduction

Dense text embeddings have become foundational for modern NLP applications including semantic search, document clustering, duplicate detection, and retrieval-augmented generation [20; 16]. Recent multilingual models such as E5 and EmbeddingGemma [28; 26] deliver high general performance, but they must allocate capacity across many languages and carry large vocabularies, which can be suboptimal for a single language like Turkish. Recent analyses also show that tokenizer design materially affects downstream behavior, motivating language-specific adaptation for morphologically rich languages [23].

For Turkish, deployment often requires both high monolingual performance and an extended context window for document-level retrieval and indexing. Existing options are either large multilingual models or older monolingual BERT variants with limited 512-token context windows. This paper introduces *embeddingmagibu-152m*, a Turkish-focused sentence embedding model in the Sentence-Transformers format [22] with a maximum sequence length of 2,048 tokens and 768-dimensional normalized outputs. The model is designed to be efficient in parameter count (152M) while maintaining an extended context window.

Rather than training from scratch, a high-capacity multilingual teacher is adapted to Turkish using an efficient three-stage pipeline. First, a Turkish SentencePiece BPE tokenizer is trained with a $2^{16} = 65{,}536$ vocabulary on the Cosmos Turkish Corpus [8] to better reflect Turkish morphology. Second, the teacher is cloned while preserving the transformer backbone weights and a compatible token embedding table is initialized for the new vocabulary via token-id mapping and embedding composition. Third, offline embedding distillation is performed by matching precomputed teacher vectors for 300,000 examples using a cosine similarity objective.

A key challenge addressed by this pipeline is that changing the tokenizer fundamentally alters the vocabulary size and token identities, making the original token embedding table incompatible. The cloning procedure addresses this by mapping each new token to one or more teacher tokens and composing their embeddings, preserving semantic information while adapting to the new vocabulary.

An end-to-end packaging and evaluation ecosystem is provided to encourage practical use and comparison of Turkish embedding models: (i) model releases on Hugging Face and Ollama for local deployment (Appendix A), (ii) a Hugging Face Space that provides a TR-MTEB benchmark results explorer and leaderboard-style interface covering a broad set of Turkish embedding models (Appendix A), (iii) a Hugging Face Space with interactive tools for similarity, retrieval, clustering, and embedding generation (Appendix A), (iv) a PyPI package for cloning Transformers and SentenceTransformers models to new tokenizers (`transformer-cloner`; Appendix A), (v) a distillation training package with configurable objectives and hyperparameters (`distil-trainer`; Appendix A), and (vi) released artifacts that enable offline training and auditing, including a dataset of precomputed teacher embeddings and exported benchmark outputs (Appendix A).

The paper is organized as follows. Section 2 reviews related work on Turkish embeddings, tokenizer adaptation, and distillation. Section 3 describes the tokenizer training, vocabulary transfer via weight-preserving cloning, and offline distillation objective. Sections 4–6 present the evaluation setup (STSbTR and TR-MTEB), main results, and analysis of training dynamics. Sections 7–8 discuss limitations, ethical considerations, and reproducibility details, followed by conclusions in Section 9.

2 Related Work

Turkish-specific sentence embedding models have recently been evaluated under unified benchmarks. TR-MTEB provides a comprehensive evaluation suite covering retrieval, semantic textual similarity, clustering, and classification tasks for Turkish and establishes baselines for Turkish-focused embedding models [4; 3]. TurkEmbed demonstrates that training on native Turkish NLI and STS data yields consistent improvements on Turkish sentence similarity and retrieval tasks [10]. Large-scale foundation models such as TabiBERT extend this line of work by training transformer encoders at scale for Turkish, paired with unified benchmarking efforts [25].

Tokenizer adaptation for morphologically rich languages is an active area of study. Turkish presents challenges for subword tokenization due to its agglutinative morphology, which can lead to fragmented subword sequences in multilingual vocabularies. Practical adaptation strategies include mapping new tokens to sequences of original tokens and composing their embeddings to initialize a compatible embedding table (Appendix A). Recent cross-lingual analyses of morphological alignment in tokenizers highlight systematic differences in subword coverage and motivate language-specific tokenizer tuning for morphologically rich languages [1]. Recent studies further analyze the impact of language-specific tokenizers on model behavior and show that tokenizer design materially influences downstream performance and efficiency [23]. Efficient tokenizer adaptation methods that combine vocabulary extension with pruning have also been proposed for transferring pretrained models to new domains or languages while controlling vocabulary growth [19]. Hybrid tokenization approaches that mix subword and linguistically motivated units offer an additional avenue to balance fragmentation and coverage [2].

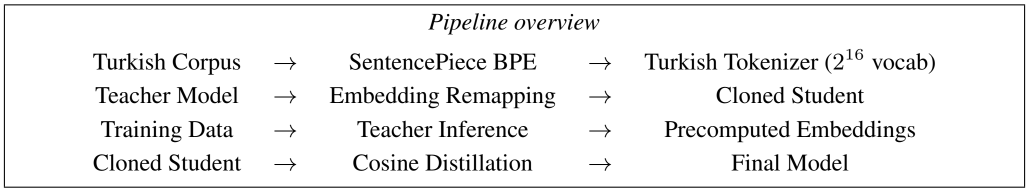
Vocabulary transfer methods explicitly address the embedding-matrix mismatch introduced by replacing a tokenizer. WECHSEL initializes target-language subword embeddings using multilingual static word vectors to align new tokens with semantically related source-language tokens [15]; in contrast, lightweight surface-form mapping approaches (including the mean-composition strategy used here) rely on composing embeddings of the teacher-tokenization of the same string.

Embedding distillation and language adaptation transfer semantic representations from a larger teacher to a smaller or language-specialized student. Sentence-level distillation was demonstrated by aligning multilingual student embeddings with English teacher embeddings using parallel data, enabling multilingual sentence representations from monolingual teachers [21]. Precomputing teacher embeddings and distilling from stored vectors (Appendix A) further reduces training cost, making language-specific adaptation pipelines more practical for resource-constrained settings. Beyond distillation, recent embedding models emphasize multi-functionality and multilinguality through self-distillation and unified training objectives [5]. Weakly supervised contrastive pretraining has also been shown to produce strong general-purpose embeddings across tasks [27], providing a common foundation for teacher models in distillation pipelines.

Embedding benchmarks and cross-lingual evaluation frameworks provide broader context. MTEB provides a broad multi-task evaluation suite for embeddings [16], while MMTEB extends evaluation to multilingual settings [9]. Language-specific benchmark extensions such as MTEB-French [6], PL-MTEB [18], and ruMTEB [24] highlight the value of localized evaluation suites and contextualize TR-MTEB as a Turkish-specific counterpart. Long-context retrieval models such as mGTE extend multilingual embedding approaches to longer inputs and reranking settings [29]. Sparse mixture-of-experts embedding models explore scaling efficiency by activating subsets of parameters for retrieval tasks [17]. Complementary evaluation work studies cross-lingual robustness using adversarial examples generated by LLMs, providing additional stress tests beyond standard benchmarks [14].

3 Method

This section describes the end-to-end pipeline used to build *embeddingmagibu-152m*: tokenizer training, model cloning with embedding remapping, teacher embedding precomputation, and embedding distillation. The pipeline is designed to retain the teacher's semantic space while reducing parameters via a Turkish-specific vocabulary. Figure 1 illustrates the overall workflow.



Pipeline overview

Turkish Corpus → SentencePiece BPE → Turkish Tokenizer (2^{16} vocab)
Teacher Model → Embedding Remapping → Cloned Student
Training Data → Teacher Inference → Precomputed Embeddings
Cloned Student → Cosine Distillation → Final Model

Figure 1: End-to-end pipeline for building *embeddingmagibu-152m*. The teacher model is used only during embedding precomputation, enabling efficient offline distillation.

Tokenizer training uses SentencePiece [13] with the BPE (Byte Pair Encoding) algorithm. The vocabulary size is set to $2^{16} = 65{,}536$ tokens, balancing between coverage of Turkish morphological patterns and embedding table parameter efficiency.

The tokenizer is trained on the Cosmos Turkish Corpus [8], a large-scale Turkish pretraining dataset. According to the dataset card, this corpus contains approximately 15 billion tokens from diverse sources including web text, with URL-based deduplication applied to reduce redundancy. The corpus is released under the CC-BY-4.0 license.

The choice of vocabulary size represents a trade-off: a smaller vocabulary reduces the embedding table size (and thus total model parameters) but may result in longer token sequences for the same text. With 2^{16} tokens, the embedding table for 768-dimensional embeddings contains $65{,}536 \times 768 \approx 50\text{M}$ parameters.

The teacher embedding model is EmbeddingGemma with 300M parameters [12; 26]. EmbeddingGemma is derived from the Gemma 3 architecture and produces 768-dimensional embedding vectors. It supports input sequences up to 2,048 tokens and includes prompt templates for query/document distinction in retrieval tasks.

The student model follows the SentenceTransformers format with a Gemma3TextModel backbone initialized from the teacher, mean pooling over token representations with `include_prompt=True` for prompt-aware encoding, two linear projections $768 \to 3072 \to 768$ without bias terms or nonlinear activations (Identity), and final ℓ_2 normalization to produce unit-length embedding vectors.

The maximum sequence length is maintained at 2,048 tokens, matching the teacher. The final embedding dimension is 768, compatible with many downstream applications.

Changing the tokenizer fundamentally alters the vocabulary: the new Turkish tokenizer has different token identities than the teacher's original tokenizer. This makes the teacher's token embedding table incompatible with the student. The approach preserves transformer backbone weights (attention, feedforward, layer normalization) while recomputing a new embedding table through token-id mapping. For each token j in the target (Turkish) vocabulary, the teacher tokenizer encoding of the same surface form is identified. This produces a mapping $\pi : j \mapsto (i_1, \ldots, i_k)$, where $(i_1, \ldots, i_k)$ is the sequence of teacher token IDs that corresponds to target token j. Given the mapping, the new embedding E'_j for target token j is initialized by combining the corresponding teacher embeddings:

$$E'_j = \text{Compose}(E_{i_1}, E_{i_2}, \ldots, E_{i_k}) \tag{1}$$

where E_i denotes the teacher embedding for token i. The composition strategy can be uniform averaging (MEAN), weighted averaging (WEIGHTED), or selection of a specific position (FIRST, LAST). Mean composition is used:

$$E'_j = \frac{1}{k} \sum_{m=1}^{k} E_{i_m} \tag{2}$$

This initialization avoids random token embeddings, reduces the embedding-table parameter count when moving from the teacher's larger vocabulary to a 65K-token student vocabulary, and preserves the transformer backbone weights exactly.

The cloning procedure is implemented in `transformer-cloner` (Appendix A), which provides utilities for building token-id mappings and executing the embedding transfer.

Running the teacher model at every training step is computationally expensive. To enable efficient training, teacher embeddings are precomputed for the training corpus and stored as a Hugging Face dataset.

The `TeacherEmbeddingsGenerator` class from `distil-trainer` (Appendix A) provides a systematic approach to embedding precomputation. The generator supports multiple output types: `final` (final sentence embedding after all layers including Dense projections), `pre_dense` (output before the Dense layer, if present), and `hidden_states` (per-layer transformer hidden states for layer-wise distillation).

The generator processes text in configurable batches, supports bf16/fp16 precision for efficient inference, and can optionally use `torch.compile()` and Flash Attention for acceleration. For large datasets, the `generate_and_push` method provides checkpoint-like behavior by pushing updates to HuggingFace Hub every n examples.

For this work, teacher embeddings are computed for 300,000 examples and stored with the columns `url`, `text`, `teacher_embedding_final` (768-dimensional final embedding), and `teacher_embedding_pre_dense` (pre-Dense embedding). The resulting dataset (Appendix A) is approximately 5–10GB depending on serialization and compression, eliminating the need to run the 300M-parameter teacher during training iterations.

The student is trained to match the teacher's embedding space using a cosine similarity objective. Let $t_i \in \mathbb{R}^d$ be the precomputed teacher embedding and $s_i \in \mathbb{R}^d$ the student embedding for input x_i. Using ℓ_2-normalized embeddings $\hat{t}_i = t_i / \|t_i\|_2$ and $\hat{s}_i = s_i / \|s_i\|_2$, the cosine distillation loss is:

$$\mathcal{L}_{\text{cos}} = \frac{1}{N} \sum_{i=1}^{N} \left(1 - \hat{s}_i^\top \hat{t}_i\right) \tag{3}$$

This loss is minimized when student and teacher embeddings are perfectly aligned (cosine similarity of 1) and maximized when they are orthogonal (cosine similarity of 0).

The distillation training uses the following hyperparameters, as specified in the model card (Appendix A): the final teacher embeddings (`teacher_embedding_final`) are distilled using the cosine loss in Equation 3 for one epoch with batch size 256 and learning rate 5×10^{-5}. A warmup ratio of 0.01, weight decay of 0.01, maximum gradient norm 1.0, and bf16 precision are used, with gradient checkpointing and `torch.compile` enabled. Checkpoints are saved every 100 steps.

Training is executed on a single NVIDIA A100 80GB GPU. The complete distillation process takes approximately four hours, demonstrating the efficiency of offline distillation compared to online approaches that would require running both teacher and student at each step.

Training progress is tracked using Weights & Biases (Appendix A). Figure 2 shows the training loss and learning rate curves from the distillation run.

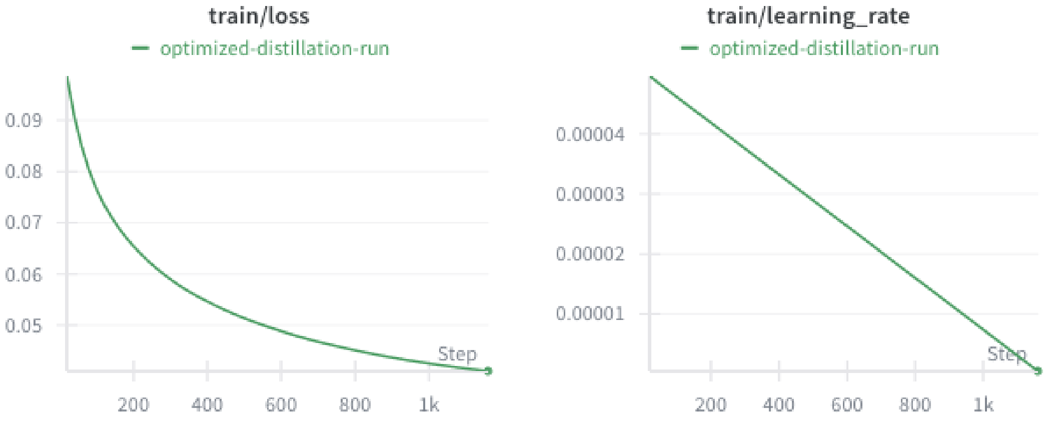


Figure 2: Training curves from the Weights & Biases distillation run. Left: cosine distillation loss decreasing from ∼0.09 to ∼0.05 over 1,000 steps. Right: learning rate schedule with warmup followed by linear decay from 5×10^{-5} to zero.

The training logs indicate rapid early optimization (loss drops from 0.09 to 0.07 within the first 200 steps) followed by steady convergence to approximately 0.05 by step 1,000. Across checkpoints, STSbTR correlation improves consistently; the student surpasses the teacher's STSbTR performance (Pearson 0.7391) approximately halfway through training and reaches 0.7512 by the final checkpoint, a 1.6% relative improvement.

The checkpoint progression (detailed in Section 6) demonstrates that a single epoch of distillation is sufficient to not only match but exceed the teacher's semantic similarity performance on Turkish benchmarks.

4 Experiments

Datasets include the Cosmos Turkish Corpus [8] for tokenizer training and a 300,000-example subset with precomputed teacher embeddings stored as a Hugging Face dataset (Appendix A); the schema and storage details are described in the Method section. Semantic textual similarity is evaluated on

STSbTR [11], the Turkish translation of the STS Benchmark. The STSbTR train split contains 5,749 sentence pairs with human similarity judgments. The repository includes checkpoint-by-checkpoint evaluation logs in `commit_results_sts.json` (tracking 15 model versions with 14 successful evaluations) and cross-model comparisons in `commit_results_sts_accross_models.json`. The TR-MTEB benchmark [3; 4] provides comprehensive evaluation across multiple task types for Turkish embeddings. Results are reported on both a 15-task subset (used in the EmbeddingGemma evaluation context) and the full 24-task benchmark. Per-task results are provided in `mteb_turkish_benchmark_15_tasks.csv` and `mteb_turkish_benchmark_24_tasks.csv`. A TR-MTEB results explorer and leaderboard-style interface covering a broad set of Turkish embedding models is provided as a Hugging Face Space (Appendix A), alongside a model demo space for interactive analysis (Appendix A).

Evaluation protocol follows standard STS practice by computing Pearson and Spearman correlation coefficients between model-predicted similarity scores and gold similarity labels. Model similarity is computed as the cosine similarity between normalized sentence embeddings. All STS evaluations use the train split of STSbTR with 5,749 samples. For TR-MTEB, task-specific metrics are reported as defined by the benchmark and the arithmetic mean is computed across tasks. The benchmark covers semantic textual similarity (STS), classification (sentiment, topic, and irony), clustering, retrieval, pair classification (e.g., XNLI and SnliTr), and bitext mining (WMT16).

Representative multilingual and Turkish-focused embedding models are included as baselines, as recorded in the evaluation artifacts. These include intfloat/multilingual-e5-large-instruct (72.77% on the 15-task subset), intfloat/multilingual-e5-large (72.28%), ytu-ce-cosmos/turkish-e5-large [7], google/embeddinggemma-300m (teacher; 70.97%) [12], selmanbaysan/turkish-embedding-model-fine-tuned (70.47%), alibaba-NLP/gte-multilingual-base (69.76%), and sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2.

5 Results and Discussion

Table 1 summarizes semantic textual similarity results on STSbTR. The *embeddingmagibu-152m* model achieves Pearson and Spearman correlations of 0.7512 and 0.7305 respectively, surpassing the EmbeddingGemma teacher (0.7391/0.7194). This result indicates that embedding-space distillation combined with Turkish tokenizer adaptation can yield a student model that exceeds teacher performance on Turkish semantic similarity. The improvement is plausibly attributable to Turkish-optimized tokenization, which provides better subword coverage for Turkish morphology compared to the teacher's multilingual vocabulary.

Table 1: STSbTR results (Pearson/Spearman correlation). Higher is better. The proposed model surpasses the teacher while using fewer parameters.

Model	Params	Pearson	Spearman
intfloat/multilingual-e5-large-instruct	560M	0.8275	0.8129
trmteb/turkish-embedding-model-fine-tuned	110M	0.8215	0.8061
ytu-ce-cosmos/turkish-e5-large	560M	0.8090	0.7906
sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2	118M	0.7884	0.7659
embeddingmagibu-152m (ours)	152M	0.7512	0.7305
google/embeddinggemma-300m (teacher)	300M	0.7391	0.7194

The gap to the top-performing models (multilingual-e5-large-instruct and Turkish-e5-large) indicates room for improvement. However, these models are significantly larger and were trained with more computational resources. The proposed model provides a favorable trade-off between performance and efficiency.

Results are reported on both the 15-task subset (Table 2) and the full 24-task benchmark (Table 3). On the 15-task subset used in the EmbeddingGemma evaluation context, *embeddingmagibu-152m* achieves an average score of 69.68%, slightly below the EmbeddingGemma teacher (70.97%). The student approaches but does not match the teacher on this aggregated metric, though it surpasses the teacher on STSbTR specifically.

Table 2: TR-MTEB 15-task subset average scores. The proposed model achieves competitive performance with half the parameters of the teacher.

Model	Params	Avg. (%)
intfloat/multilingual-e5-large-instruct	560M	72.77
intfloat/multilingual-e5-large	560M	72.28
ytu-ce-cosmos/turkish-e5-large	560M	72.22
google/embeddinggemma-300m (teacher)	300M	70.97
selmanbaysan/turkish-embedding-model-fine-tuned	110M	70.47
embeddingmagibu-152m (ours)	152M	69.68

On the full 24-task TR-MTEB benchmark, *embeddingmagibu-152m* achieves 62.57%, outperforming some models with similar or larger parameter counts. Notably, it surpasses the Turkish fine-tuned embedding model (62.17%) while providing a 2,048-token context window.

Table 3: TR-MTEB 24-task full benchmark average scores.

Model	Params	Avg. (%)
intfloat/multilingual-e5-large	560M	65.59
ytu-ce-cosmos/turkish-e5-large	560M	64.84
intfloat/multilingual-e5-large-instruct	560M	64.72
alibaba-NLP/gte-multilingual-base	305M	63.25
intfloat/multilingual-e5-base	278M	63.00
embeddingmagibu-152m (ours)	152M	62.57
selmanbaysan/turkish-embedding-model-fine-tuned	110M	62.17

Table 4 presents per-task results for detailed analysis. The model performs particularly well on WMT16BitextMining (96.33%), QuoraRetrievalTR (93.95%), TurkishNewsCategoryClassification (91.12%), and Turkish75NewsClassification (90.00%). Weaker performance is observed on SCIDOCSTR (2.71%), NFCorpusTR (8.68%), and FiQA2018TR (36.96%), suggesting that domain- and genre-specific fine-tuning may be beneficial for specialized retrieval settings.

These results suggest that while the model provides competitive general-purpose Turkish embeddings, domain-specific fine-tuning may be beneficial for specialized applications.

Table 4: Per-task TR-MTEB results (full 24-task benchmark) for *embeddingmagibu-152m*.

Task	Score	Task	Score
WMT16BitextMining	96.33	QuoraRetrievalTR	93.95
TurkishNewsCategoryClassification	91.12	Turkish75NewsClassification	90.00
TQuadRetrieval	84.63	SciFactTR	80.82
TurkishMovieSentimentClassification	75.19	STSbTR	73.79
SquadTRRetrieval	70.57	XNLI	67.84
TurkishColumnWritingClustering	63.80	TurkishProductSentimentClassification	63.52
TSTimelineNewsCategoryClassification	62.91	TurkishOffensiveLanguageClassification	61.50
TurkishAbstractCorpusClustering	61.11	THYSentimentClassification	58.42
SnliTr	55.56	MSMarcoTRRetrieval	53.07
TurkishIronyClassification	53.50	CQADupstackGamingRetrievalTR	50.54
ArguAnaTR	45.11	FiQA2018TR	36.96
NFCorpusTR	8.68	SCIDOCSTR	2.71

The results indicate that high-quality Turkish embeddings can be obtained with modest computational resources. The complete training process requires approximately 4 hours on a single A100 GPU, compared to days or weeks for training from scratch. The offline distillation strategy-precomputing teacher embeddings before training-is a key contributor to this efficiency. The observation that the student surpasses the teacher on STSbTR (Table 1) is notable; a plausible explanation is that Turkish-optimized tokenization provides better subword representations for Turkish text, whereas the teacher's multilingual vocabulary may yield suboptimal segmentations for Turkish morphology. While the model achieves competitive results, a gap remains to the best-performing models on TR-MTEB, reflecting the trade-off between model size, training compute, and performance. The approach prioritizes efficiency and accessibility while maintaining utility for Turkish NLP applications.

6 Ablations and Analysis

This section discusses the design choices in the proposed pipeline and analyzes training dynamics based on the checkpoint evaluation logs.

By reducing the vocabulary size to 65,536, the embedding table is reduced and the total parameter count is brought down to approximately 152M while preserving a 2,048-token context window. This enables long-document retrieval use cases without the heavy computational footprint of 300M+ parameter multilingual models. The checkpoint evaluation logs in `commit_results_sts.json` reveal the training dynamics across 14 successful checkpoint evaluations. Table 5 shows STSbTR performance at selected checkpoints.

Table 5: STSbTR performance progression during training. The student starts below the teacher (0.7391 Pearson) and surpasses it by checkpoint 7. Performance continues improving with diminishing gains.

Checkpoint	Time	Pearson	Spearman	vs Teacher
1 (initial)	2025-12-31 09:51	0.6536	0.6310	−11.6%
2	2026-01-01 18:22	0.6637	0.6377	−10.2%
3	2026-01-01 18:42	0.7077	0.6853	−4.2%
4	2026-01-01 19:13	0.7281	0.7057	−1.5%
5	2026-01-01 19:43	0.7355	0.7145	−0.5%
6	2026-01-01 20:14	0.7374	0.7168	−0.2%
7	2026-01-01 20:45	0.7455	0.7255	+0.9%
8	2026-01-01 21:16	0.7467	0.7266	+1.0%
9	2026-01-01 21:47	0.7505	0.7292	+1.5%
10	2026-01-01 22:17	0.7494	0.7284	+1.4%
11	2026-01-01 22:48	0.7513	0.7303	+1.7%
12	2026-01-01 23:19	0.7510	0.7302	+1.6%
13	2026-01-01 23:50	0.7512	0.7305	+1.6%
14 (final)	2026-01-02 00:12	0.7512	0.7305	+1.6%

The training progression suggests that embedding remapping provides an effective initialization: even the initial checkpoint (before significant training) achieves 0.6536 Pearson. Performance improves rapidly in the early checkpoints (0.6637 to 0.7281, a relative gain of 9.7%), the student exceeds the teacher around checkpoint 7 (teacher Pearson 0.7391), and gains continue through the final checkpoint with diminishing returns.

The cloning procedure uses mean composition to combine teacher embeddings when a target token maps to multiple source tokens. Alternative strategies include weighted mean (e.g., by token frequency or position), first/last-token selection, or a learned initialization that starts from the mean and is subsequently tuned. The mean strategy provides a reasonable initialization without introducing additional hyperparameters. The checkpoint progression shows that even straightforward mean composition enables effective knowledge transfer.

The method uses $2^{16} = 65{,}536$ tokens, reducing the embedding table from the teacher's larger vocabulary. This choice affects the embedding-table parameter count (approximately 50M parameters for $65{,}536 \times 768$), token sequence length (smaller vocabularies can increase sequence length), and morphological coverage (balancing fine-grained versus coarse-grained tokenization for an agglutinative language). A systematic study varying vocabulary size (32K, 64K, 128K) would clarify these trade-offs for Turkish.

Distillation uses the teacher's final normalized embeddings (`teacher_embedding_final`). The precomputed dataset also includes `teacher_embedding_pre_dense` for alternative experiments. Potential targets include pre-dense embeddings (before the projection layers), layer-wise distillation that matches intermediate representations, or attention-based distillation that matches attention patterns. The final embedding target provides the most direct alignment with the teacher's semantic space for downstream tasks.

The training logs show that a single epoch is sufficient for this distillation setup. Loss decreases throughout training with the student improving from 0.6536 to 0.7512 Pearson. The final checkpoints show diminishing returns (0.7510 → 0.7512 → 0.7512), suggesting convergence.

The available artifacts do not provide controlled ablations sufficient to draw causal conclusions. Specifically, only one vocabulary size (65K) is evaluated, only final embeddings are used as the distillation target, only cosine loss is explored (alternatives include MSE and contrastive losses), and training is conducted for a single epoch. Future work should systematically vary these factors to understand their individual contributions.

7 Limitations

Several limitations should be noted. The current artifact set does not include systematic ablations over tokenizer size, distillation targets, or training data composition, so controlled experiments are needed to quantify the contribution of each component. Evaluation is limited to STSbTR and TR-MTEB as provided in the released artifacts; additional evaluations on domain-specific corpora (e.g., legal, medical, e-commerce), in-production retrieval systems, or cross-lingual settings are not covered. The tokenizer is trained on a broad Turkish web corpus, which may not optimally represent specialized domains, and deployments in legal, medical, or highly colloquial contexts may require domain-specific adaptation. Although the model supports 2,048-token inputs, performance on long documents is not extensively evaluated; very long content may require chunking and aggregation strategies (e.g., hierarchical pooling) beyond the simple mean pooling used here. The mean-composition initialization is not compared against alternative tokenizer adaptation methods such as vocabulary extension with pruning, semantic initialization methods like WECHSEL [15], or hybrid tokenization strategies [19; 2]; controlled comparisons would clarify the trade-offs between initialization quality and adaptation cost. Mean composition also ignores polysemy and context: a surface-form token may map to teacher subwords whose embeddings reflect multiple senses, and averaging cannot disambiguate which sense is relevant. The training recipe focuses on matching teacher embeddings for single texts and does not include supervised contrastive training on Turkish NLI/IR data, which may limit ceiling performance compared to models trained with task-specific objectives. The approach depends on the quality and characteristics of the EmbeddingGemma teacher; limitations or biases in the teacher model may propagate to the student, and alternative or ensemble teachers could potentially improve results. While the model is optimized for Turkish, performance may degrade on code-mixed text (Turkish-English) or other Turkic languages, and cross-lingual capabilities from the teacher may not be fully preserved after Turkish-focused tokenization. While high-level training procedures and artifacts are released, some implementation details (e.g., exact random seeds, data shuffling) may affect exact reproducibility, and comparable compute resources may not be accessible to all researchers. The model is designed for embedding generation and is not suitable for open-ended text generation or instruction-following tasks.

8 Reproducibility

Comprehensive information is provided to facilitate reproduction of results. The model weights are available on Hugging Face as `magibu/embeddingmagibu-152m`, and the precomputed embeddings dataset is released as `alibayram/cosmos-corpus-0-05-with-embeddings` (Appendix A). For local deployment, the model is also distributed as a GGUF package via Ollama (Appendix A). Released artifacts include tokenizer files (`tokenizer.model`, `tokenizer.json`, `tokenizer_config.json`, `special_tokens_map.json`), model configuration (`config.json`, `modules.json`, `sentence_bert_config.json`), module configurations and weights (`1_Pooling/`, `2_Dense/`, `3_Dense/`), and the model weights (`model.safetensors`). Evaluation outputs include STSbTR checkpoint evaluations (`commit_results_sts.json`), cross-model comparisons (`commit_results_sts_accross_models.json`), and TR-MTEB exports for the 15-task and 24-task benchmarks (`mteb_turkish_benchmark_15_tasks.csv` and `mteb_turkish_benchmark_24_tasks.csv`). Training logs are available via the Weights & Biases run (Appendix A). Interactive inspection is supported via the Hugging Face Spaces referenced in Section 4, including a benchmark results explorer and an interactive demo.

The following packages are required:

```
pip install -U sentence-transformers datasets sentencepiece
pip install -U transformer-cloner distil-trainer
```

Pipeline steps are as follows. First, train a SentencePiece BPE tokenizer on Cosmos Turkish Corpus [8]:

```
import sentencepiece as spm
spm.SentencePieceTrainer.train(
    input=’cosmos_corpus.txt’,
    model_prefix=’turkish_bpe’,
    vocab_size=65536,
    model_type=’bpe’
)
```

Second, clone the teacher model with the new tokenizer using `transformer-cloner` (Appendix A):

```
from transformer_cloner import TransformerCloner
cloner = TransformerCloner(
    source_model=’google/embeddinggemma-300m’,
    target_tokenizer=’./turkish_bpe.model’
)
cloner.clone(output_path=’./cloned_model’)
```

Third, generate the teacher embeddings dataset using `distil-trainer` (Appendix A):

```
from distil_trainer.data import TeacherEmbeddingsGenerator
generator = TeacherEmbeddingsGenerator(
    teacher_model=’google/embeddinggemma-300m’
)
generator.generate(
    dataset=’cosmos_subset’,
    output_path=’./embeddings_dataset’
)
```

Fourth, train the student model (Appendix A):

```
from distil_trainer import EmbeddingDistillationTrainer
trainer = EmbeddingDistillationTrainer(
    student_model=’./cloned_model’,
    embeddings_dataset=’./embeddings_dataset’,
    target_type=’final’,
    loss=’cosine’,
    batch_size=256,
    learning_rate=5e-5,
    num_epochs=1,
    precision=’bf16’
)
trainer.train()
```

Hardware requirements include a single NVIDIA A100 80GB GPU (or an equivalent GPU with ≥40GB VRAM). Distillation takes approximately 4 hours, and the precomputed-embeddings dataset requires approximately 5–10GB of storage depending on serialization and compression. Complete training hyperparameters as specified in the model card (Appendix A):

9 Conclusion

embeddingmagibu-152m is a Turkish-focused sentence embedding model with an extended 2,048-token context window and 768-dimensional outputs. The three-stage pipeline—Turkish tokenizer training, weight-preserving model cloning with embedding remapping, and offline distillation from precomputed teacher embeddings—provides an efficient approach to language-specific model adaptation.

Empirically, the student surpasses its EmbeddingGemma teacher on STSbTR (0.7512/0.7305 vs. 0.7391/0.7194 Pearson/Spearman), indicating that Turkish-optimized tokenization combined with distillation can improve performance over a multilingual teacher on Turkish semantic similarity. On TR-MTEB, the model achieves competitive performance (69.68% on 15 tasks and 62.57% on 24 tasks) while using 152M parameters. More broadly, offline distillation from precomputed embeddings reduces training cost relative to online approaches.

Future work includes controlled ablations over vocabulary size, distillation targets, and training data scale; comparisons with alternative tokenizer adaptation methods [19; 2]; extending the approach to other Turkic languages; domain-specific fine-tuning (e.g., legal, medical, or technical text); and integration with retrieval-augmented generation pipelines for Turkish.

In addition to the model, supporting infrastructure is released for adoption and benchmarking, including open-source cloning and distillation tooling, precomputed embedding datasets for offline training, and Hugging Face Spaces for interactive exploration and benchmark result inspection.

A Supplementary Artifacts

The following non-peer-reviewed artifacts support reproducibility and reuse:

- **Model card (Hugging Face)**: https://huggingface.co/magibu/embeddingmagibu-152m
- **Precomputed embeddings dataset**: https://huggingface.co/datasets/alibayram/cosmos-corpus-0-05-with-embeddings
- **GGUF package (Ollama)**: https://ollama.com/alibayram/embeddingmagibu-152m
- **Interactive demo (Hugging Face Space)**: https://huggingface.co/spaces/magibu/embeddingmagibu-152m
- **TR-MTEB results explorer (Hugging Face Space)**: https://huggingface.co/spaces/magibu/mteb-turkish
- **distil-trainer (code)**: https://github.com/malibayram/distil-trainer
- **transformer-cloner (code)**: https://github.com/malibayram/transformer-cloner
- **Weights & Biases run**: https://api.wandb.ai/links/alibayram-ytu/srxzzhof

References

[1] Catherine Arnett, Marisa Hudspeth, and Brendan O'Connor. Morphscore: Evaluating morphological awareness of tokenizers across languages. In *Proceedings of the Tokenization Workshop at ICML 2025*, 2025. URL https://arxiv.org/abs/2507.06378. Accepted to the Tokenization Workshop at ICML 2025.

[2] M. Ali Bayram, Ali Arda Fincan, Ahmet Semih Gümüş, Sercan Karakaş, Banu Diri, Savaş Yıldırım, and Demircan Çelik. Tokens with meaning: A hybrid tokenization approach for nlp. https://arxiv.org/abs/2508.14292, 2025. Preprint.

[3] Selman M. Baysan. Mteb-tr: Turkish massive text embedding benchmark. https://github.com/selmanbaysan/mteb_tr.

[4] Selman M. Baysan and Tunga Güngör. Tr-mteb: A comprehensive benchmark and embedding model suite for turkish sentence representations. https://aclanthology.org/2025.findings-emnlp.471/, 2025.

[5] Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.137. URL https://aclanthology.org/2024.findings-acl.137/.

[6] Mathieu Ciancone, Imene Kerboua, Marion Schaeffer, and Wissam Siblini. Mteb-french: Resources for french sentence embedding evaluation and analysis. https://arxiv.org/abs/2405.20468, 2024. Preprint.

[7] YTU CE Cosmos. Turkish-e5-large: E5 model enhanced for turkish with multi-positive contrastive learning. https://huggingface.co/ytu-ce-cosmos/turkish-e5-large.

[8] YTU CE Cosmos. Cosmos turkish corpus v1.0. https://huggingface.co/datasets/ytu-ce-cosmos/Cosmos-Turkish-Corpus-v1.0, 2024.

[9] Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemiński, Genta Indra Winata, Saba Sturua, Saiteja Utpala, Mathieu Ciancone, Marion Schaeffer, Gabriel Sequeira, Diganta Misra, Shreeya Dhakal, Jonathan Rystrøm, Roman Solomatin, Ömer Çağatan, Akash Kundu, Martin Bernstorff, Shitao Xiao, Akshita Sukhlecha, Bhavish Pahwa, Rafał Poświata, Kranthi Kiran GV, Shawon Ashraf, Daniel Auras, Björn Plüster, Jan Philipp Harries, Loïc Magne, Isabelle Mohr, Mariya Hendriksen, Dawei Zhu, Hippolyte Gisserot-Boukhlef, Tom Aarsen, Jan Kostkan, Konrad Wojtasik, Taemin Lee, Marek Šuppa, Crystina Zhang, Roberta Rocca, Mohammed Hamdy, Andrianos Michail, John Yang, Manuel Faysse, Aleksei Vatolin, Nandan Thakur, Manan Dey, Dipam Vasani, Pranjal Chitale, Simone Tedeschi, Nguyen Tai, Artem Snegirev, Michael Günther, Mengzhou Xia, Weijia Shi, Xing Han Lù, Jordan Clive, Gayatri Krishnakumar, Anna Maksimova, Silvan Wehrli, Maria Tikhonova, Henil Panchal, Aleksandr Abramov, Malte Ostendorff, Zheng Liu, Simon Clematide, Lester James Miranda, Alena Fenogenova, Guangyu Song, Ruqiya Bin Safi, Wen-Ding Li, Alessia Borghini, Federico Cassano, Hongjin Su, Jimmy Lin, Howard Yen, Lasse Hansen, Sara Hooker, Chenghao Xiao, Vaibhav Adlakha, Orion Weller, Siva Reddy, and Niklas Muennighoff. Mmteb: Massive multilingual text embedding benchmark. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=zl3pfz4VCV. Accepted at ICLR 2025.

[10] Özay Ezerceli, Gizem Gümüşçekiçci, Tuğba Erkoç, and Berke Özenç. Turkembed: Turkish embedding model on natural language inference & sentence text similarity tasks. In *2025 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pages 1–6. IEEE, September 2025. doi: 10.1109/ASYU67174.2025.11208511. URL https://doi.org/10.1109/ASYU67174.2025.11208511.

[11] Figen Fikri. Stsb-tr: Turkish sts benchmark. https://huggingface.co/datasets/figenfikri/stsb_tr.

[12] Google. Embeddinggemma-300m model card. https://huggingface.co/google/embeddinggemma-300m.

[13] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing, 2018.

[14] Andrianos Michail, Simon Clematide, and Rico Sennrich. Examining multilingual embedding models cross-lingually through LLM-generated adversarial examples. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 2161–2170, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.115. URL https://aclanthology.org/2025.findings-emnlp.115/.

[15] Benjamin Minixhofer, Fabian Paischer, and Navid Rekabsaz. Wechsel: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models. https://aclanthology.org/2022.naacl-main.293/, 2022.

[16] Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. MTEB: Massive text embedding benchmark. In Andreas Vlachos and Isabelle Augenstein, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.148. URL https://aclanthology.org/2023.eacl-main.148/.

[17] Zach Nussbaum and Brandon Duderstadt. Training sparse mixture of experts text embedding models. https://arxiv.org/abs/2502.07972, 2025. Preprint.

[18] Rafał Poświata, Sławomir Dadas, and Michał Perełkiewicz. Pl-mteb: Polish massive text embedding benchmark. https://arxiv.org/abs/2405.10138, 2024. Preprint.

[19] Taido Purason, Pavel Chizhov, Ivan P. Yamshchikov, and Mark Fishel. Teaching old tokenizers new words: Efficient tokenizer adaptation for pre-trained models. https://arxiv.org/abs/2512.03989, 2025. Preprint.

[20] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410/.

[21] Nils Reimers and Iryna Gurevych. Making monolingual sentence embeddings multilingual using knowledge distillation. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.365. URL https://aclanthology.org/2020.emnlp-main.365/.

[22] SentenceTransformers. https://www.sbert.net/. https://www.sbert.net/.

[23] Jean Seo, Jaeyoon Kim, SungJoo Byun, and Hyopil Shin. How does a language-specific tokenizer affect llms? https://arxiv.org/abs/2502.12560, 2025. Preprint.

[24] Artem Snegirev, Maria Tikhonova, Maksimova Anna, Alena Fenogenova, and Aleksandr Abramov. The Russian-focused embedders' exploration: ruMTEB benchmark and Russian embedding model design. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 236–254, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.12. URL https://aclanthology.org/2025.naacl-long.12/.

[25] Melikşah Türker, A. Ebrar Kızıloğlu, Onur Güngör, and Susan Üsküdarlı. Tabibert: A large-scale modernbert foundation model and unified benchmarking framework for turkish. https://arxiv.org/abs/2512.23065, 2025. Preprint.

[26] Henrique Schechter Vera, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, Sara Smoot, Iftekhar Naim, Joe Zou, Feiyang Chen, Daniel Cer, Alice Lisak, Min Choi, Lucas Gonzalez, Omar Sanseviero, Glenn Cameron, Ian Ballantyne, Kat Black, Kaifeng Chen, Weiyi Wang, Zhe Li, Gus Martins, Jinhyuk Lee, Mark Sherwood, Juyeong Ji, Renjie Wu, Jingxiao Zheng, Jyotinder Singh, Abheesht Sharma, Divyashree Sreepathihalli, Aashi Jain, Adham Elarabawy, AJ Co, Andreas Doumanoglou, Babak Samari, Ben Hora, Brian Potetz, Dahun Kim, Enrique Alfonseca, Fedor Moiseev, Feng Han, Frank Palma Gomez, Gustavo Hernández Ábrego, Hesen Zhang, Hui Hui, Jay Han, Karan Gill, Ke Chen, Koert Chen, Madhuri Shanbhogue, Michael Boratko, Paul Suganthan, Sai Meher Karthik Duddu, Sandeep Mariserla, Setareh Ariafar, Shanfeng Zhang, Shijie Zhang, Simon Baumgartner, Sonam Goenka, Steve Qiu, Tanmaya Dabral, Trevor Walker, Vikram Rao, Waleed Khawaja, Wenlei Zhou, Xiaoqi Ren, Ye Xia, Yichang Chen, Yi-Ting Chen, Zhe Dong, Zhongli Ding, Francesco Visin, Gaël Liu, Jiageng Zhang, Kathleen Kenealy, Michelle Casbon, Ravin Kumar, Thomas Mesnard, Zach Gleicher, Cormac Brick, Olivier Lacombe, Adam Roberts, Qin Yin, Yunhsuan Sung, Raphael Hoffmann, Tris Warkentin, Armand Joulin, Tom Duerig, and Mojtaba Seyedhosseini. Embeddinggemma: Powerful and lightweight text representations. https://arxiv.org/abs/2509.20354, 2025. Preprint.

[27] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. https://arxiv.org/abs/2212.03533, 2022. Preprint.

[28] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Multilingual e5 text embeddings: A technical report. https://arxiv.org/abs/2402.05672, 2024. Preprint.

[29] Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. mGTE: Generalized long-context text representation and reranking models for multilingual text retrieval. In Franck Dernoncourt, Daniel Preoţiuc-Pietro, and Anastasia Shimorina, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1393–1412, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-industry.103. URL https://aclanthology.org/2024.emnlp-industry.103/.