

# Tokens with Meaning: A Hybrid Tokenization Approach for Turkish

Summary for PhD Thesis

Ali Bayram

February 2026

## Abstract

This document provides a comprehensive summary of the paper "Tokens with Meaning: A Hybrid Tokenization Approach for Turkish," which proposes MFT (Morphology-First Tokenizer), a linguistically informed hybrid tokenizer for Turkish. The tokenizer combines dictionary-driven root/affix segmentation with phonological normalization and a BPE fallback, achieving 90.29% Turkish Token Percentage and 85.80% Pure Token Percentage on TR-MMLU dataset. Controlled downstream experiments with sentence embedding models show that MFT provides substantial gains: +16.8 percentage points on STSb-TR and +5.7 points on MTEB-TR compared to baseline BPE tokenizers.

## Contents

<b>1</b>	<b>Introduction and Motivation</b>	<b>3</b>
1.1	The Turkish Language Challenge . . . . .	3
1.2	Research Contributions . . . . .	3
<b>2</b>	<b>Methodology</b>	<b>3</b>
2.1	Tokenizer Architecture . . . . .	3
2.1.1	Dictionary Construction . . . . .	3
2.1.2	Encoding Algorithm . . . . .	4
2.1.3	Decoding Algorithm . . . . .	4
2.1.4	Roundtrip Reconstruction Accuracy . . . . .	5
2.2	Tokenization Efficiency . . . . .	5
<b>3</b>	<b>Tokenization Quality Evaluation</b>	<b>5</b>
3.1	Evaluation Metrics . . . . .	5
3.2	TR-MMLU Benchmark Results . . . . .	6
<b>4</b>	<b>Downstream Task Evaluation</b>	<b>6</b>
4.1	Experimental Setup . . . . .	6
4.2	Semantic Textual Similarity (STSb-TR) . . . . .	7
4.3	MTEB-TR Benchmark . . . . .	7
4.4	TurBLiMP Linguistic Evaluation . . . . .	9
4.5	Training Stability . . . . .	9

<b>5 Discussion and Analysis</b>	<b>10</b>
5.1 Why Morphology-First Tokenization Works . . . . .	10
5.2 Token Count Trade-off . . . . .	10
5.3 Model Architecture Compatibility . . . . .	11
<b>6 Conclusion</b>	<b>11</b>

# 1 Introduction and Motivation

Tokenization is the process of mapping raw text into a sequence of discrete units (tokens) that a model can embed and process. While subword tokenization methods such as Byte Pair Encoding (BPE), WordPiece, and Unigram have become standard for transformer-based models, their behavior is not neutral for morphologically rich languages.

## 1.1 The Turkish Language Challenge

Turkish exhibits rich suffix morphology and systematic morphophonological alternations:

- **Vowel harmony:** Suffix allomorphs such as *-lAr* (plural) and *-dAn/-tAn* (ablative) realize the same grammatical morpheme under different phonological contexts
- **Consonant alternations:** Forms like *kitap* → *kitabı* (p→b before a vowel) create predictable surface variants
- **Agglutinative structure:** A single word like *kitaplarımızdan* ("from our books") contains the root *kitap* plus four suffixes

Standard BPE tokenizers treat these variants as unrelated units, inflating vocabulary redundancy and reducing semantic coherence.

## 1.2 Research Contributions

This work makes three main contributions:

1. A **morphology-first hybrid tokenizer** for Turkish with an explicit decoder for lossless reconstruction
2. **Quantitative tokenization quality evaluation** on TR-MMLU against widely used tokenizers
3. **Controlled downstream comparisons** across four matched embedding models to assess whether improved morpheme alignment translates into better sentence representations

# 2 Methodology

## 2.1 Tokenizer Architecture

MFT follows a three-stage pipeline: preprocessing, morphology-first dictionary lookup, and subword fallback.

### 2.1.1 Dictionary Construction

The vocabulary consists of three components:

Table 1: MFT vocabulary composition.

Component	Entries	Token IDs	ID Range
Roots	22,231	20,000	0–19,999
Suffixes	177 forms	72	20,000–20,071
BPE tokens	12,696	12,696	20,072–32,767
<b>Total</b>	—	<b>32,768</b>	—

**Root Dictionary:** The root dictionary comprises approximately 22,000 roots extracted from large-scale Turkish corpora. Surface variants map to a single canonical root ID through phonological normalization.

**Affix Dictionary:** 177 suffix surface forms are consolidated into 72 abstract affix IDs. Allomorphs serving identical grammatical functions share IDs (e.g., *-ler/-lar* → PL).

**BPE Fallback:** 12,696 BPE tokens handle out-of-vocabulary terms and foreign words.

### 2.1.2 Encoding Algorithm

The encoder uses Trie-based data structures for  $O(n)$  prefix matching. For each position, it finds all prefix matches in three tries (roots, suffixes, BPE) and selects the best match using a priority-weighted scoring system.

---

**Algorithm 1** MFT Encoding Algorithm

---

```

1: Input: Raw text string  $S$ 
2: Output: Sequence of Token IDs  $T$ 
3: Build Tries:  $\mathcal{T}_{\text{roots}}$ ,  $\mathcal{T}_{\text{suffixes}}$ ,  $\mathcal{T}_{\text{bpe}}$ 
4: for each word  $w$  in  $S.\text{split}('')$  do
5:   if  $w[0]$  is uppercase then
6:      $T.\text{append}(\text{ID}_{\text{uppercase}})$ 
7:      $w \leftarrow \text{TurkishLowercase}(w)$                                  $\triangleright I \rightarrow i, I \rightarrow 1$ 
8:   end if
9:    $w \leftarrow ' ' + w$                                                $\triangleright$  Prepend space for word boundary
10:   $pos \leftarrow 0$ 
11:  while  $pos < |w|$  do
12:     $\text{substr} \leftarrow w[pos :]$ 
13:     $R \leftarrow \mathcal{T}_{\text{roots}}.\text{FindAllPrefixes}(\text{substr})$                  $\triangleright \{(id, len, chars)\}$ 
14:     $B \leftarrow \mathcal{T}_{\text{bpe}}.\text{FindAllPrefixes}(\text{substr})$ 
15:     $S \leftarrow \mathcal{T}_{\text{suffixes}}.\text{FindAllPrefixes}(\text{substr})$ 
16:     $\text{best} \leftarrow \text{SelectBest}(R, B, S, \text{substr})$                        $\triangleright$  Priority: roots > bpe > suffix
17:     $T.\text{append}(\text{best}.id)$ 
18:     $pos \leftarrow pos + \text{best}.len$ 
19:  end while
20: end for
21: Return  $T$ 

```

---

**Priority Scoring:** For each candidate match, the score is computed as  $\text{chars} \times 10$ . Additionally, BPE and suffix matches receive a lookahead bonus if the remainder starts with a valid suffix ( $\geq 2$  chars). Among equal scores, priority favors: roots (0) > BPE (1) > suffixes (2).

### 2.1.3 Decoding Algorithm

The decoder reconstructs surface forms by applying Turkish phonological rules based on context. Each suffix ID maps to multiple allomorphic surface forms, and the decoder selects the appropriate variant.

---

#### Algorithm 2 MFT Decoding Algorithm

---

```

1: Input: Sequence of Token IDs  $T$ 
2: Output: Reconstructed text string  $S$ 
3: parts  $\leftarrow \emptyset$ 
4:  $i \leftarrow 0$ 
5: while  $i < |T|$  do
6:    $id \leftarrow T[i]$ 
7:   if  $id = \text{ID}_{\text{uppercase}}$  then
8:     next  $\leftarrow \text{ReverseLookup}(T[i + 1])$ 
9:     parts.append(TurkishCapitalize(next))            $\triangleright i \rightarrow \dot{I}, 1 \rightarrow I$ 
10:     $i \leftarrow i + 2$ ; continue
11:   end if
12:   candidates  $\leftarrow \text{ReverseLookup}(id)$             $\triangleright$  List of surface forms
13:   if  $|\text{candidates}| > 1$  then
14:     if  $id \in [20000, 20071]$  then                       $\triangleright$  Suffix range
15:       ctx  $\leftarrow \text{GetVowelContext}(\text{parts})$            $\triangleright$  Look back for vowel
16:       surface  $\leftarrow \text{SelectSuffix}(id, \text{ctx}, T, i)$ 
17:     else                                          $\triangleright$  Root with alternations
18:       surface  $\leftarrow \text{SelectRoot}(id, T, i)$ 
19:     end if
20:   else
21:     surface  $\leftarrow \text{candidates}[0]$ 
22:   end if
23:   parts.append(surface)
24:    $i \leftarrow i + 1$ 
25: end while
26: Return parts.join("")
```

---

#### Phonological Rules Applied:

- **Vowel harmony:** Suffix vowels selected based on last vowel of context (front/back: e,i,ö,ü vs a,ı,o,u)
- **Consonant assimilation:**  $d \rightarrow t$  after voiceless consonants (f,s,t,k,ç,ş,h,p)
- **Vowel narrowing:**  $e \rightarrow i$  before progressive suffix -yor (e.g., *de*  $\rightarrow$  *diyor*)
- **Consonant softening:**  $p \rightarrow b, k \rightarrow$  before vowel-initial suffixes (e.g., *kitap*  $\rightarrow$  *kitabı*)
- **Buffer consonants:** y/n inserted between vowels (e.g., *okuma* + ACC  $\rightarrow$  *okumayı*)

### 2.1.4 Roundtrip Reconstruction Accuracy

Word-level roundtrip accuracy on 500 randomly sampled words: **99.2% exact-match accuracy** (496/500 words). The 0.8% failures arise from inherent phonological ambiguity in Turkish where multiple surface realizations are linguistically valid.

## 2.2 Tokenization Efficiency

Table 2: Tokenization efficiency comparison (1,000 texts, 653K words).

Tokenizer	Time (ms)	Tokens	Tok/Word	Tok/Char
MFT	1,935	1,899,670	<b>2.91</b>	0.356
Tabi	1,544	1,298,725	1.99	0.244
Mursit	1,655	1,187,418	1.82	0.223
Cosmos	1,620	1,186,834	1.82	0.223

MFT produces approximately  $1.5 \times$  more tokens because it segments at morpheme boundaries rather than optimizing for compression. This trade-off yields improved downstream performance.

## 3 Tokenization Quality Evaluation

### 3.1 Evaluation Metrics

Following the protocol of Bayram et al. (2025), we evaluate tokenization quality using:

- **Turkish Token Percentage (TR%)**: Proportion of tokens corresponding to valid Turkish words or morphemes
- **Pure Token Percentage (Pure%)**: Proportion of tokens fully aligned with unambiguous root/affix boundaries

TR% and Pure% are computed using an independent morphological validator with curated lexical resources external to the tokenizer under evaluation.

### 3.2 TR-MMLU Benchmark Results

Table 3: Performance on TR-MMLU dataset.

Metric	Value
Vocabulary Size	32,768
Total Token Count	707,727
Processing Time (s)	0.6714
Unique Token Count	11,144
Turkish Token Count	10,062
Turkish Token Percentage (TR%)	<b>90.29%</b>
Pure Token Count	9,562
Pure Token Percentage (Pure%)	<b>85.80%</b>

The proposed tokenizer substantially outperforms all competing tokenizers:

- google/gemma-2-9b: TR% = 40.96%, Pure% = 28.49%
- meta-llama/Llama-3.2-3B: TR% = 45.77%, Pure% = 31.45%
- CohereForAI/aya-expanse-8b: TR% = 53.48%

## 4 Downstream Task Evaluation

### 4.1 Experimental Setup

To isolate the effect of tokenization from pre-training data, we conduct controlled experiments with randomly initialized models:

Table 4: Embedding distillation experiment setup.

Component	Specification
Student architecture	google/embeddinggemma-300m
Initialization	Random weights with fixed seed 42
Vocabulary size	32,768 (matched across all models)
Teacher model	intfloat/multilingual-e5-large-instruct
Training objective	Cosine embedding loss
Context length	2,048 tokens
Batch size / LR	$256 / 5 \times 10^{-5}$
Schedule	Two-phase: 100-step warmup + 1 epoch
Hardware	NVIDIA H100 80GB

Four models are compared: MFT, Tabi, Mursit, and Cosmos—all using independently trained BPE tokenizers with matched vocabulary sizes.

### 4.2 Semantic Textual Similarity (STSb-TR)

Table 5: STS benchmark correlations.

Model	Split	Pearson	Spearman	Time (s)
MFT	test ( $n = 1379$ )	<b>50.37</b>	<b>49.35</b>	10.68
Mursit	test ( $n = 1379$ )	43.94	43.75	7.67
Cosmos	test ( $n = 1379$ )	38.90	38.02	7.52
Tabi	test ( $n = 1379$ )	33.58	33.24	8.18
<b>MFT Advantage over Tabi</b>		<b>+16.79</b>	<b>+16.11</b>	—

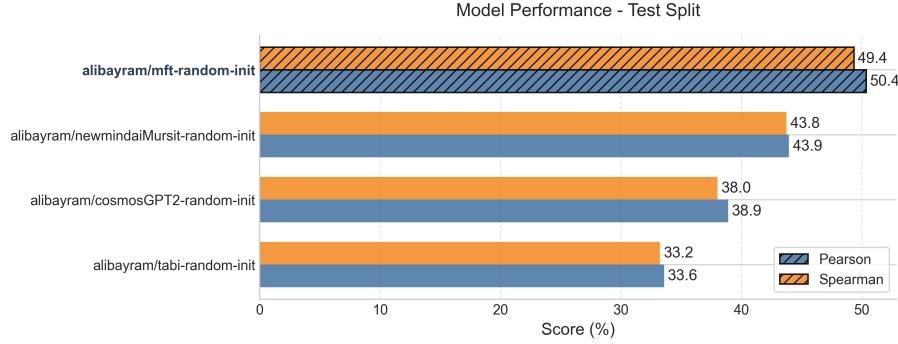


Figure 1: STS benchmark test split performance comparison.

### 4.3 MTEB-TR Benchmark

Table 6: MTEB-TR category averages.

Category	MFT	Cosmos	Mursit	Tabi
BitextMining	<b>1.53</b>	1.08	1.26	1.39
Classification	<b>58.81</b>	57.52	57.71	56.52
Clustering	66.30	<b>66.45</b>	65.39	65.71
Other	<b>4.94</b>	1.58	2.19	1.89
Pair Classification	<b>50.25</b>	47.94	46.89	47.43
Retrieval	<b>28.94</b>	20.10	21.12	18.46
STS	<b>49.36</b>	38.04	43.75	33.24
<b>Overall Average</b>	<b>38.99</b>	34.43	34.98	33.33

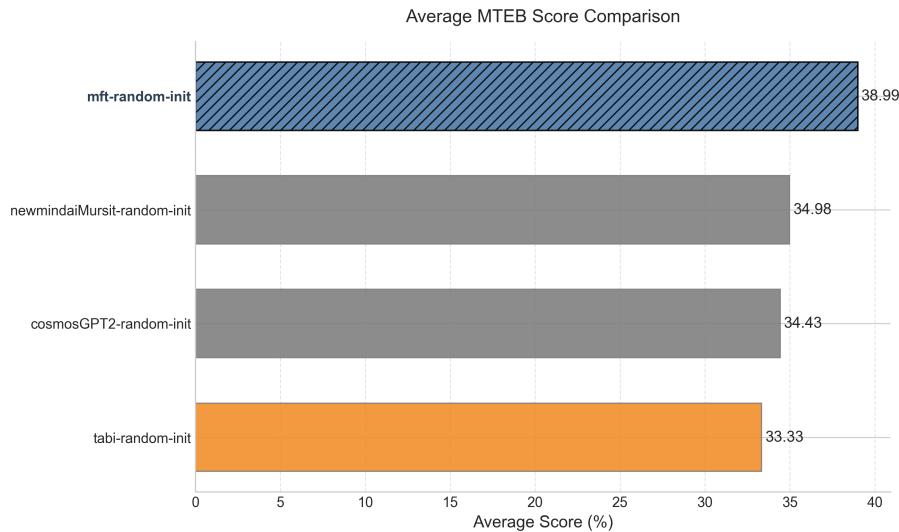


Figure 2: Average MTEB-TR scores across all tasks.

Table 7: Detailed MTEB-TR task-level performance (selected tasks).

Task	MFT	Cosmos	Mursit	Tabi
TurkishNewsCategoryClassification	<b>85.40</b>	83.32	80.52	79.08
TurkishProductSentimentClassification	<b>54.34</b>	51.39	51.85	52.10
QuoraRetrievalTR	<b>63.01</b>	46.44	49.24	46.98
TQuadRetrieval	<b>43.46</b>	29.97	29.48	26.30
MSMarcoTRRetrieval	<b>12.84</b>	6.07	6.13	4.83
STSbTR	<b>49.36</b>	38.04	43.75	33.24

MFT demonstrates substantial advantages in *STS* and *Retrieval* tasks, supporting the hypothesis that morphology-aware segmentation improves semantic embedding quality.

#### 4.4 TurBLiMP Linguistic Evaluation

Table 8: TurBLiMP sensitivity scores by linguistic phenomenon.

Linguistic Phenomenon	Mursit	MFT	Cosmos	Tabi
Ellipsis	98.0%	<b>99.2%</b>	97.7%	93.0%
Scrambling	97.7%	<b>98.0%</b>	97.3%	<b>98.0%</b>
Determiners	94.9%	<b>96.5%</b>	93.4%	94.1%
Relative Clauses	89.1%	<b>96.1%</b>	86.3%	93.0%
Suspended Affixation	89.5%	<b>93.4%</b>	83.2%	91.0%
Subject Verb Agreement	84.8%	<b>89.5%</b>	79.7%	86.3%
Island Effects	79.7%	<b>84.0%</b>	76.2%	<b>84.0%</b>

MFT yields higher similarity between minimal pairs across most linguistic phenomena, suggesting that morphology-first segmentation produces more stable semantic representations.

## 4.5 Training Stability

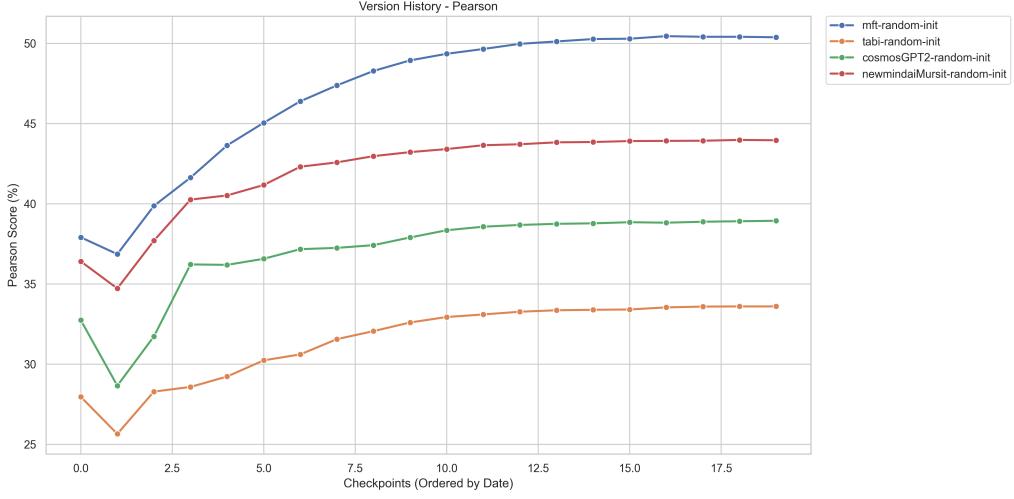


Figure 3: Pearson correlation across model revisions.

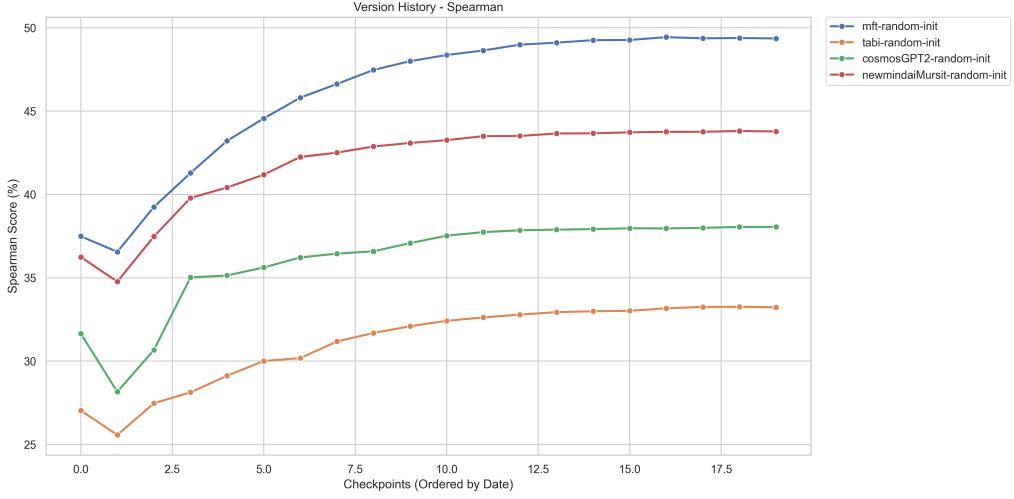


Figure 4: Spearman correlation across model revisions.

All three baseline tokenizers (Mursit, Cosmos, Tabi)—which are independently trained BPE tokenizers from different research groups—show consistent relative ordering across all benchmarks. This cross-tokenizer consistency provides strong evidence that performance differences reflect genuine tokenizer-induced inductive bias rather than random initialization variance.

## 5 Discussion and Analysis

### 5.1 Why Morphology-First Tokenization Works

The substantial downstream gains (+16.8 pp on STS) can be attributed to several factors:

1. **Semantic coherence:** Each token represents a meaningful linguistic unit (root or affix) rather than an arbitrary subword

2. **Embedding reuse:** Phonologically normalized representations allow the same root embedding to be shared across surface variants
3. **Grammatical compositionality:** Explicit affix tokens enable the model to learn compositional semantics of Turkish morphology

## 5.2 Token Count Trade-off

MFT produces 2.91 tokens per word compared to 1.99 for baseline BPE tokenizers. While this increases sequence length by approximately 46%, the morpheme-aligned representations yield improved sample efficiency during training:

- **STSb-TR:** +16.8 percentage points improvement
- **MTEB-TR Retrieval:** +10.5 percentage points improvement
- **TurBLiMP Relative Clauses:** +9.8 percentage points improvement

The trade-off favors semantic quality over sequence efficiency.

## 5.3 Model Architecture Compatibility

The 99.2% roundtrip reconstruction accuracy enables MFT to be used across all transformer architectures:

- **Encoder-only models** (BERT-style): Exact reconstruction not required
- **Encoder-decoder models** (translation, summarization): Near-lossless encoding
- **Decoder-only models** (GPT-style generation): 99.2% ensures correct outputs

## 6 Conclusion

This work presented MFT, a linguistically informed morphology-first hybrid tokenizer for Turkish. Key findings:

1. **Tokenization quality:** 90.29% TR% and 85.80% Pure% on TR-MMLU, substantially exceeding general-purpose tokenizers
2. **Semantic similarity:** +16.79 percentage points on STSb-TR compared to Tabi baseline
3. **Retrieval tasks:** +10.5 percentage points on MTEB-TR Retrieval average
4. **Reconstruction:** 99.2% word-level roundtrip accuracy

These results demonstrate that morphology-first tokenization provides a stronger inductive bias for learning Turkish semantic representations from scratch, validating the motivation that linguistically informed tokenization is essential for morphologically rich languages.

## Key Figures and Tables Summary

Table 9: Summary of all experimental results.

Benchmark	Metric	MFT	Best Baseline
TR-MMLU	TR%	90.29%	53.48% (Aya)
TR-MMLU	Pure%	85.80%	31.45% (LLaMA)
STSb-TR	Pearson	50.37%	43.94% (Mursit)
STSb-TR	Spearman	49.35%	43.75% (Mursit)
MTEB-TR	Average	38.99%	34.98% (Mursit)
MTEB-TR	Retrieval Avg	28.94%	21.12% (Mursit)
Roundtrip	Exact Match	99.2%	—