Tokens with Meaning: A Hybrid Tokenization Approach for Turkish

M. Ali Bayram[1], Ali Arda Fincan[2], Ahmet Semih Gümüş[2], Sercan Karakaş[3],
Banu Diri[1], Savaş Yıldırım[4], Demircan Çelik[2]
[1]Yıldız Technical University, [2]Yeditepe University, [3]University of Chicago,
[4]Istanbul Bilgi University
malibayram20@gmail.com

Abstract

Tokenization shapes how language models perceive morphology and meaning in Natural Language Processing (NLP), yet widely used frequency-driven subword tokenizers (e.g., Byte Pair Encoding and WordPiece) can fragment morphologically rich and agglutinative languages in ways that obscure morpheme boundaries. We introduce a linguistically informed hybrid tokenizer for Turkish that combines (i) dictionary-driven morphological segmentation (roots and affixes), (ii) phonological normalization that maps allomorphic variants to shared identifiers, and (iii) a controlled subword fallback for out-of-vocabulary coverage. Concretely, our released Turkish vocabulary contains 20,000 root identifiers, 72 affix identifiers that cover 177 allomorphic surface forms, and 12,696 subword units; special tokens represent whitespace and orthographic case without inflating the vocabulary. We evaluate tokenization quality on TR-MMLU using two linguistic alignment metrics: Turkish Token Percentage (TR %), the proportion of produced tokens that correspond to Turkish lexical/morphemic units under our lexical resources, and Pure Token Percentage (Pure %), the proportion of tokens aligning with unambiguous root/affix boundaries. The proposed tokenizer reaches 90.29% TR % and 85.80% Pure % on TR-MMLU, substantially exceeding several general-purpose tokenizers. We further validate practical utility with downstream sentence embedding benchmarks under a strict *random initialization* control to isolate tokenizer inductive bias. Across four matched models (MFT, CosmosGPT2, Mursit, and Tabi), MFT improves Semantic Textual Similarity Benchmark (STSb-TR) Pearson correlation from 33.58% (Tabi) to 50.37%, and achieves the strongest overall average on Massive Text Embedding Benchmark (MTEB-TR) and the strongest TurBLiMP linguistic sensitivity among the evaluated baselines.

Keywords: Tokenization, Morphologically Rich Languages, Morphological Segmentation, Byte Pair Encoding, Turkish NLP, Linguistic Integrity, Low-Resource Languages

1 Introduction

Tokenization is the process of mapping raw text into a sequence of discrete units (tokens) that a model can embed and process. It influences vocabulary construction, sequence length, interpretability, and ultimately performance in downstream tasks [1]. While subword tokenization has become a standard design choice for transformer-based models, its behavior is not neutral for morphologically rich languages.

Byte Pair Encoding (BPE) [2], WordPiece [3], and Unigram [4] address out-of-vocabulary (OOV) words by representing rare forms as compositions of frequent subword units. This improves coverage and keeps vocabularies compact, but it can also split words in ways that cut across morpheme boundaries and blur grammatical function [5, 6]. Such fragmentation is especially relevant for agglutinative languages such as Turkish, where productive suffixation yields many surface forms from relatively few lemmas.

Turkish exhibits rich suffix morphology and systematic morphophonological alternations, including vowel harmony and consonant alternations at morpheme boundaries. For example, suffix allomorphs such as *-lAr* (plural) and *-dAn*/*-tAn* (ablative) realize the same grammatical morpheme under different phonological contexts, and consonant alternations such as *kitap* → *kitabı* (p→b before a vowel) create predictable surface variants of the same stem. Tokenizers that treat these variants as unrelated units can inflate redundancy and reduce the reuse of meaning-bearing units across inflections [7].

This paper introduces the *Morphology-First Tokenizer* (MFT), a linguistically informed hybrid tokenizer for Turkish. The method combines dictionary-driven morphological segmentation (roots and affixes), a normalization layer that maps common allomorphic variants to shared identifiers, and a controlled subword fallback for open-vocabulary coverage. We also include explicit tokens for whitespace and orthographic case so that formatting and casing information can be preserved without duplicating vocabulary entries.

We evaluate tokenization quality on TR-MMLU [8] using two linguistic alignment metrics: Turkish Token Percentage (TR %) and Pure Token Percentage (Pure %), which quantify lexical/morphemic coverage and alignment with unambiguous root/affix boundaries, respectively [7]. To address reviewer concerns about real-world applicability, we further include downstream evaluation on sentence embedding benchmarks. In a controlled random-initialization setting, we compare four matched models (MFT, CosmosGPT2, Mursit, and Tabi) on Semantic Textual Similarity (STS), MTEB-TR, and TurBLiMP, a Turkish benchmark of linguistic minimal pairs [9].

Our contributions are threefold: we propose a morphology-first hybrid tokenizer for Turkish that is lossless via an explicit decoder; we provide a quantitative and qualitative evaluation of tokenization quality on TR-MMLU against widely used tokenizers; and we report controlled downstream comparisons across four matched embedding models to assess whether improved morpheme alignment translates into better sentence representations.

2 Related Work

Tokenization is a fundamental step in NLP, significantly impacting model performance, memory efficiency, and downstream task effectiveness. Tokenization strategies range from character-level segmentation to subword-based methods such as BPE [2], WordPiece [3], and Unigram [10]. The choice of tokenization directly influences the ability of models to capture syntactic, semantic, and morphological structures, especially in morphologically rich languages such as Turkish, Finnish, and Hungarian [11, 5].

Recent research has explored alternative tokenization strategies tailored to morphologically rich languages. Toraman et al. [5] analyze the impact of tokenization on Turkish language modeling, and report that morphology-aware tokenization can recover much of the performance of larger baselines under certain settings. Kaya and Tantuğ [6] examine tokenization granularity for Turkish language models, and highlight that Turkish can require substantially more subword splits per word than English under common subword tokenizers, underscoring the importance of vocabulary design and sequence length control.

Tokenization strategies also play a crucial role in machine translation and text generation tasks. Pan et al. [12] demonstrate that morphology-aware segmentation can reduce sparsity in neural machine translation, and Huck et al. [13] study target-side segmentation strategies that improve translation quality by maintaining linguistic consistency between source and target languages. Beyond translation, morphology-aware tokenization has also been evaluated in abstractive summarization and sentiment analysis. Baykara and Güngör [11] discuss summarization for agglutinative languages, and Kayalı and Omurca [14] propose a hybrid tokenization strategy for Turkish summarization. Such hybrid approaches are also commonly motivated by applications where preserving linguistic structure is important (e.g., named entity recognition (NER)).

Tokenization quality is also discussed in the context of modern LLM tokenizers, where differences in segmentation can affect non-English text processing and evaluation outcomes. Bayram et al. [7]

compare several widely used tokenizers on Turkish and highlight how tokenizer-specific segmentation artifacts can influence downstream benchmarking.

Despite these advancements, the computational cost of tokenization and its interaction with training efficiency remains an open concern. Larger vocabularies can increase model size and memory footprint [15, 1], and the energy and carbon footprint of training large models has motivated more careful reporting and efficiency analysis [16]. From this perspective, tokenization is not only a linguistic design choice, but also a practical lever that affects sequence length and compute; inefficient vocabulary utilization and redundant segmentation can translate into longer sequences and higher training cost [16].

To address trade-offs between linguistic alignment and efficiency, recent work has explored adaptive and multilingual tokenization strategies. Martins et al. [17] describe multilingual language models and tokenization choices across European languages, and Lin et al. [18] study token selection strategies that question whether all tokens contribute equally during pretraining. Dynamic tokenization approaches that adapt segmentation rules have also been proposed; for example, Neubeck et al. [19] explore a more flexible BPE-style tokenizer.

Several approaches incorporate linguistic structure directly into tokenization. Hofmann et al. [20] show that derivationally informed segmentation can improve model interpretation of complex word forms. MorphPiece [21] segments by morphemes before applying a subword encoding step, aiming to preserve compositional meaning while remaining compatible with standard training pipelines. Closest to our design are hybrid tokenizers that combine explicit linguistic resources with statistical fallback. miLLi [22] is a tokenizer for Azerbaijani that uses a root dictionary, BPE fallback, and a phonological restoration mechanism to increase root consistency across surface variants. Another line of work modifies the subword algorithm itself to better respect morphological structure: MorphBPE [23] extends BPE with morphology-aware constraints and introduces morphology-based evaluation metrics, reporting improved morphological alignment and training behavior across multiple languages.

Tokenization strategies play a critical role in pretraining large language models (LLMs), influencing model efficiency, generalization, and performance across downstream tasks. Transformer-based architectures such as BERT [15], RoBERTa [1], and GPT [24] rely on effective tokenization to balance vocabulary size, sequence length, and computational cost. Studies have shown that tokenization choices can interact with morphological compositionality and generalization, particularly for morphologically rich languages [25].

Benchmark evaluations such as Massive Multitask Language Understanding (MMLU) [26] and TR-MMLU [8] have highlighted the need for language-aware evaluation. Bayram et al. [7] propose a linguistic integrity framework for evaluating Turkish tokenization, introducing metrics such as token purity and Turkish Token Percentage (TR %). Their results suggest that higher TR % and purity correlate with stronger performance on MMLU-style Turkish benchmarks, motivating our focus on morpheme-level alignment.

Turkish-specific benchmarks and evaluation suites have expanded rapidly. TR-MMLU [8] provides a large-scale Turkish evaluation set for language model assessment, and TurBLiMP [9] offers a controlled benchmark of linguistic minimal pairs covering diverse phenomena. In parallel, Turkish-focused model and tokenizer ecosystems continue to grow. For example, TabiBERT [27] provides a modern Turkish encoder and a unified evaluation suite, reinforcing the value of language-specific baselines when assessing tokenizer behavior and downstream impact.

Finally, tokenization considerations extend beyond language modeling into applied pipelines such as optical character recognition and document parsing. Rashad et al. [28] demonstrate that tokenizer choices can affect structure reconstruction and recognition accuracy in Arabic document processing, and Rosa et al. [29] provide a tokenizer benchmark in a multilingual setting, illustrating that tokenizer behavior can vary widely across languages and domains.

3 Methodology

Traditional NLP models primarily relied on word-level tokenization, where each word was treated as an individual token. However, this approach was inadequate for handling OOV words, requiring extensive vocabulary lists that resulted in inefficient memory usage [24]. To address this, subword tokenization methods such as BPE and WordPiece emerged, segmenting rare words into smaller, frequently occurring subunits, thereby improving generalization and reducing OOV occurrences. BPE, originally introduced for data compression [30] and later adapted for NLP by Sennrich et al. [2], iteratively merges frequent adjacent character pairs into subword units. Similarly, WordPiece, which was initially developed for speech recognition [3], follows a comparable iterative merging approach but optimizes token selection using likelihood-based probability maximization.

Morphological complexity presents a significant challenge for NLP tokenization, particularly in agglutinative languages such as Turkish, Hungarian, and Finnish. These languages exhibit a high degree of word inflection, resulting in a vast array of surface forms derived from relatively few lemmas [17]. In Turkish, for instance, the word *anlayabildiklerimizden* ('from what we were able to understand') is composed of multiple morphemes: *anla-* (UNDERSTAND) + *-yabil* (ABLE) + *-dik* (NOMINALIZER) + *-ler* (PLURAL) + *-imiz* (1PL.POSS) + *-den* (ABLATIVE). Standard subword tokenization methods such as BPE and WordPiece often fail to capture such rich internal structures, fragmenting words in ways that obscure grammatical function and semantic interpretation [6]. This misalignment reduces linguistic coherence and can negatively impact downstream tasks, highlighting the need for tokenizers that are sensitive to language-specific morphological and phonological features.

The hybrid tokenization framework combines linguistic knowledge with statistical subword segmentation techniques to enhance tokenization performance in morphologically rich languages, using Turkish as a benchmark. The approach integrates rule-based morphological analysis with a structured dictionary of roots and affixes while incorporating BPE to handle OOV words and ambiguous segments. The objective is to create a tokenization system that accurately represents linguistic structures while maintaining computational efficiency.

We provide a Python reference implementation of the tokenizer and release the lexical resources (root and affix inventories) and decoder rules used in our experiments. The tokenization process follows a structured pipeline consisting of three key components: dictionary-based morphological segmentation, BPE-based fallback segmentation, and special tokens that preserve formatting and orthographic information.

Morphological segmentation is a key component of the proposed approach, leveraging a dual-dictionary system to identify and segment words. The root dictionary is constructed from high-frequency words extracted from large-scale Turkish corpora, ensuring broad lexical coverage of common stems. This dictionary is augmented with normalization rules that reduce sparsity caused by morphophonological alternations under suffixation, such as consonant alternations (e.g., *kitap* → *kitabı*), haplology (e.g., *alın* → *alnı*), and vowel hiatus (e.g., *oyna* + *yor* → *oynuyor*). Furthermore, some frequent compounds are assigned unique identifiers to avoid inconsistent splits; for example, *akarsu* ('stream', literally *akar+su*) and *çamaşırhane* ('laundromat', *çamaşır+hane*).

The affix dictionary consists of approximately 230 linguistic elements, including suffixes, prepositions, and conjunctions. To improve efficiency and reduce redundancy, affixes with identical grammatical functions, such as the plural markers "-lAr" or the ablative markers "-dAn," are assigned a common identifier. This approach ensures that morphologically equivalent structures do not inflate the vocabulary size while preserving their grammatical roles in sentence construction.

To ensure comprehensive token coverage, the framework integrates BPE to segment words that are not explicitly listed in the morphological dictionaries. The training data for the BPE vocabulary was sourced from large-scale Turkish corpora, and we trained the subword model using the SentencePiece library in BPE mode to obtain a vocabulary of 10,000 subword units, which is then incorporated into the tokenizer. This enables the system to process novel words while retaining consistency in morphological decomposition for covered segments.

Special tokens are introduced to handle whitespace, punctuation, capitalization, and unknown words, enhancing the tokenizer's ability to preserve linguistic structure. A dedicated token for whitespace ensures that spacing information is explicitly encoded, preserving sentence structure during tokenization. Additionally, an uppercase token is introduced to differentiate capitalized words from their lowercase counterparts without inflating the vocabulary. Additional tokens account for newline characters, tab spaces, and unknown words, preventing tokenization errors when encountering unfamiliar input.

The encoding process begins with morphological analysis, where the longest matching root is identified from the dictionary. Once the root is determined, suffix segmentation is performed by iteratively checking for affix matches. If a valid segmentation cannot be identified using the

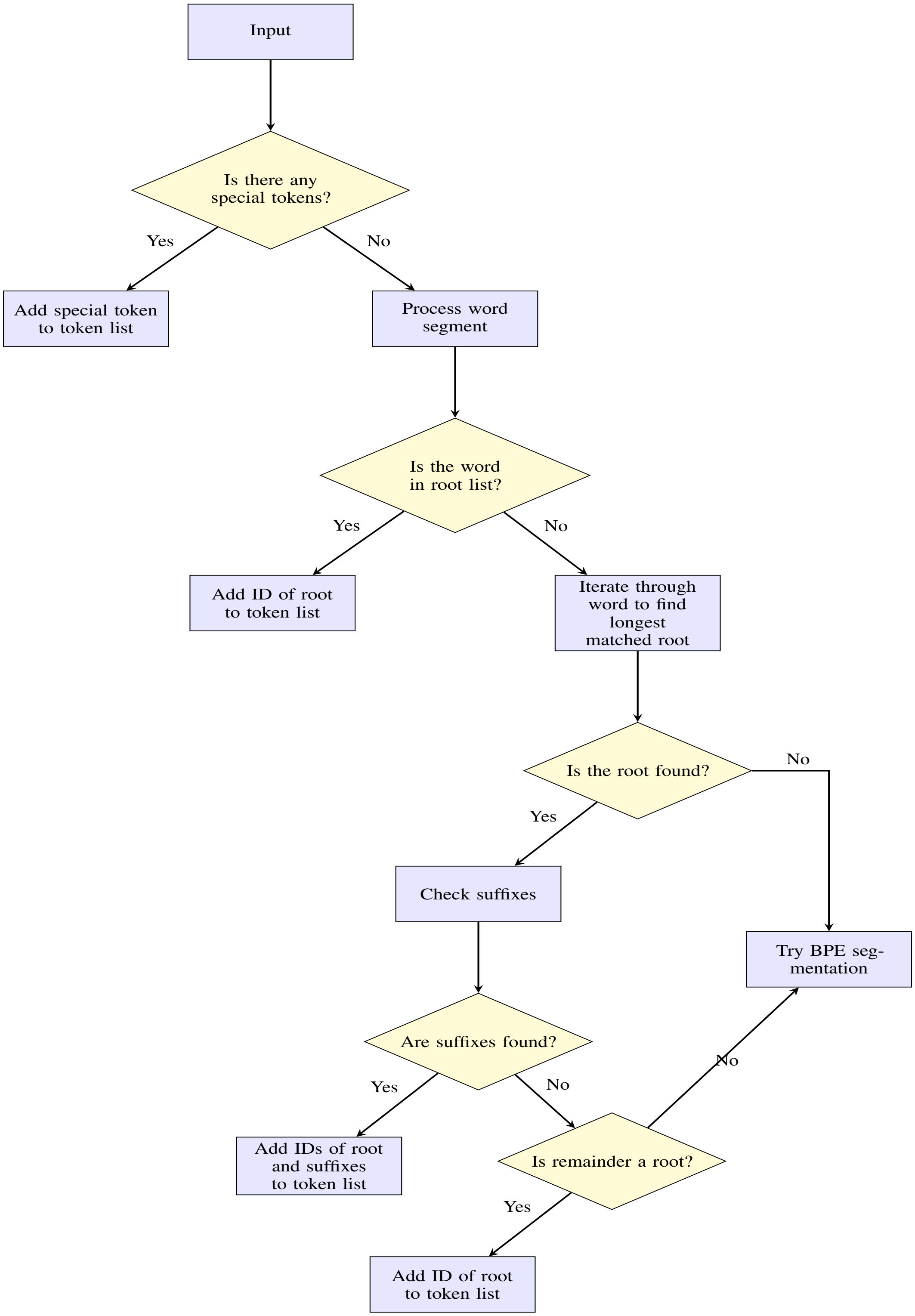
Input
Is there any special tokens?
Yes
No
Add special token to token list
Process word segment
Is the word in root list?
Yes
No
Add ID of root to token list
Iterate through word to find longest matched root
Is the root found?
No
Yes
Check suffixes
Try BPE segmentation
Are suffixes found?
No
Yes
No
Add IDs of root and suffixes to token list
Is remainder a root?
Yes
Add ID of root to token list

Figure 1: Tokenization decision flow with root, suffix, and fallback segmentation logic.

morphological dictionary, the remaining portion of the word is processed using BPE-based subword segmentation. Words that do not match any predefined root, suffix, or subword are assigned an unknown token, ensuring robustness in handling OOV terms.

The decoding process reconstructs tokenized text while adhering to linguistic rules. A reverse mapping mechanism ensures that phonological alternations are restored correctly, preserving morphosyntactic dependencies. Disambiguation rules are applied to select the most probable reconstruction based on phonetic context and grammatical constraints. This process enhances readability while maintaining fidelity to the original text.

The proposed framework provides a balance between linguistic integrity and coverage. By integrating morphological analysis with BPE-based fallback segmentation, the tokenizer preserves morpheme-aligned units where possible while remaining robust to unseen words.

The construction of the tokenizer dictionary follows a structured approach that ensures comprehensive coverage of Turkish morphology while maintaining efficiency. The dictionary consists of three primary components: a root word list, an affix list, and a set of functional words such as prepositions and conjunctions. These elements form the basis of the tokenization process, enabling accurate segmentation and linguistic representation.

The root dictionary is built from a dataset of high-frequency Turkish words extracted from large-scale corpora. This dataset includes approximately 22,000 roots, ensuring broad lexical coverage. Each root is assigned a unique identifier, allowing for consistent referencing throughout the tokenization process. To improve efficiency, roots are categorized based on their length, enabling a hierarchical lookup mechanism that prioritizes longer roots before shorter alternatives. This method significantly enhances root detection speed by reducing the number of comparisons required.

An additional layer of processing is applied to handle phonological alternations in root words, which frequently occur in Turkish due to sound changes triggered by suffixation. To ensure consistency and reduce vocabulary sparsity, different phonetic realizations of the same morphological root are mapped to a single identifier. For example, final devoicing results in surface variations such as *kitap* ('book') and *kitabı* ('its book'), both of which are assigned the same root ID. Similarly, haplology in forms like *alın* ('forehead') and *alnı* ('his/her forehead'), and vowel hiatus in forms like *oyna* + *yor* → *oynuyor* ('he/she/it is playing') are normalized through unified token mappings. This phonological normalization preserves morphological coherence while avoiding unnecessary token duplication.

In addition to root words, the dictionary includes a comprehensive inventory of approximately 230 suffixes, prepositions, and conjunctions, compiled from authoritative linguistic sources and organized according to grammatical function. To further optimize vocabulary size without compromising syntactic accuracy, affixes that perform the same grammatical role are assigned a shared identifier. For instance, plural suffixes such as *-lAr*, or ablative markers like *-dAn*, *-tAn*, functionally represent the same morphemes but differ based on phonological context. This strategy is also applied to locative markers like *-dA* and *-tA*, which exhibit surface variation due to consonant alternation rules. By merging such phonologically conditioned allomorphs, the tokenizer reduces redundancy while maintaining linguistic fidelity.

Compound words represent another important aspect of Turkish morphology, wherein multiple roots combine to form a single semantic unit. To prevent incorrect segmentation, frequently used compounds such as *akarsu* ('stream') and *çamaşırhane* ('laundromat') are directly included in the dictionary and assigned unique token IDs. This ensures that compound expressions are treated as indivisible lexical items, preserving their semantic integrity and avoiding erroneous decomposition into root-affix pairs.

Beyond roots and affixes, the dictionary incorporates functional words such as prepositions and conjunctions, which play a crucial role in sentence structure. These elements are often challenging to tokenize correctly due to their small size and high frequency. By including them explicitly in the dictionary, the tokenizer avoids erroneous segmentations that might result from statistical subword approaches.

The integration of BPE further enhances tokenization flexibility. While the dictionary provides structured linguistic segmentation, BPE ensures robust handling of words not explicitly covered in the predefined lexicon. The BPE model is trained on a diverse Turkish corpus, incorporating approximately 10,000 subword units to supplement dictionary-based tokenization. The combined approach enables the tokenizer to efficiently process both frequent and rare words, ensuring comprehensive text coverage.

Another important aspect of the proposed framework is its ability to handle case sensitivity without increasing vocabulary size. A dedicated uppercase token is introduced to mark words that were originally capitalized. This avoids the need to store separate tokens for capitalized and lowercase versions of the same word, optimizing storage efficiency while preserving orthographic distinctions.

The dictionary-driven approach provides a balance between linguistic accuracy and computational efficiency. By leveraging structured linguistic resources, normalizing phonological variations, and integrating statistical subword segmentation, the tokenizer achieves robust performance across diverse text types. The next section will describe the encoding and decoding processes in detail, outlining how tokenization is applied in practice to segment and reconstruct text.

The encoding process follows a hierarchical approach that ensures linguistic consistency while maintaining computational efficiency. The tokenizer operates in a multi-step pipeline that sequentially applies morphological analysis, affix segmentation, and subword processing. This structured approach optimizes tokenization accuracy while preserving essential linguistic features.

The encoding process begins with preprocessing, where special characters and formatting elements are replaced with predefined tokens. Whitespace characters such as spaces, newlines, and tab spaces are explicitly encoded using dedicated tokens. This step ensures that text formatting is preserved, preventing information loss in structured text. Additionally, words that begin with capital letters are marked with an uppercase token to maintain case information without inflating the vocabulary.

Following preprocessing, the tokenizer applies root detection using a hierarchical lookup strategy. The algorithm first searches for the longest matching root in the dictionary, prioritizing exact matches before considering phonological variants. If a match is found, the root is assigned its corresponding token ID. In cases where no direct match is identified, alternative scenarios such as compound words or phonologically altered roots are considered. This flexible approach ensures that words are correctly segmented even when phonological modifications are present.

Once the root is identified, suffix segmentation is performed iteratively. The algorithm checks for affix matches in the suffix dictionary and assigns token IDs accordingly. Each identified suffix is treated as a separate token, maintaining its grammatical function while ensuring proper segmentation. The suffix matching process continues until no further valid suffixes can be extracted. If an affix is ambiguous or overlaps with multiple possible segmentations, a probabilistic model selects the most likely segmentation based on corpus frequency data.

If a word does not match any predefined root or suffix, BPE is applied as a fallback mechanism. The BPE model segments the word into subword units based on a pre-trained vocabulary, ensuring that unknown words are processed effectively. This hybrid approach prevents the tokenizer from failing on unseen words while maintaining the linguistic integrity of known structures.

For example, the word *kalktığımızda* ('when we stood up') is segmented into its root and affix components as follows:

Input text: `"Kalktığımızda hep birlikte yürüdük."` ("When we stood up, we walked together.")

Token sequence: `[uppercase], kalk, tığ, ımız, da, [space], hep, [space], birlikte, [space], yürü, dü, k, .`

Token IDs: `0, 1502, 22280, 22285, 22278, 1, 2300, 1, 4803, 1, 2280, 22296, 22617, 22582`

This example demonstrates how the encoder accurately identifies the root *kalk* ("stand up"), segments its suffixes (*-tığ* "past nominalizer", *-ımız* "our", *-da* "when/at"), and preserves syntactic structure using dedicated space and punctuation tokens. Each token corresponds to a morphologically meaningful unit, enabling interpretable and reversible text representations.

The decoding process reconstructs surface text from tokenized sequences while maintaining linguistic accuracy. Token IDs are mapped back to their textual forms, and affixes are recombined according to their grammatical function. During this step, phonological alternations are reversed: rules for soft consonantization, vowel deletion, and contraction are reapplied to ensure natural word formation.

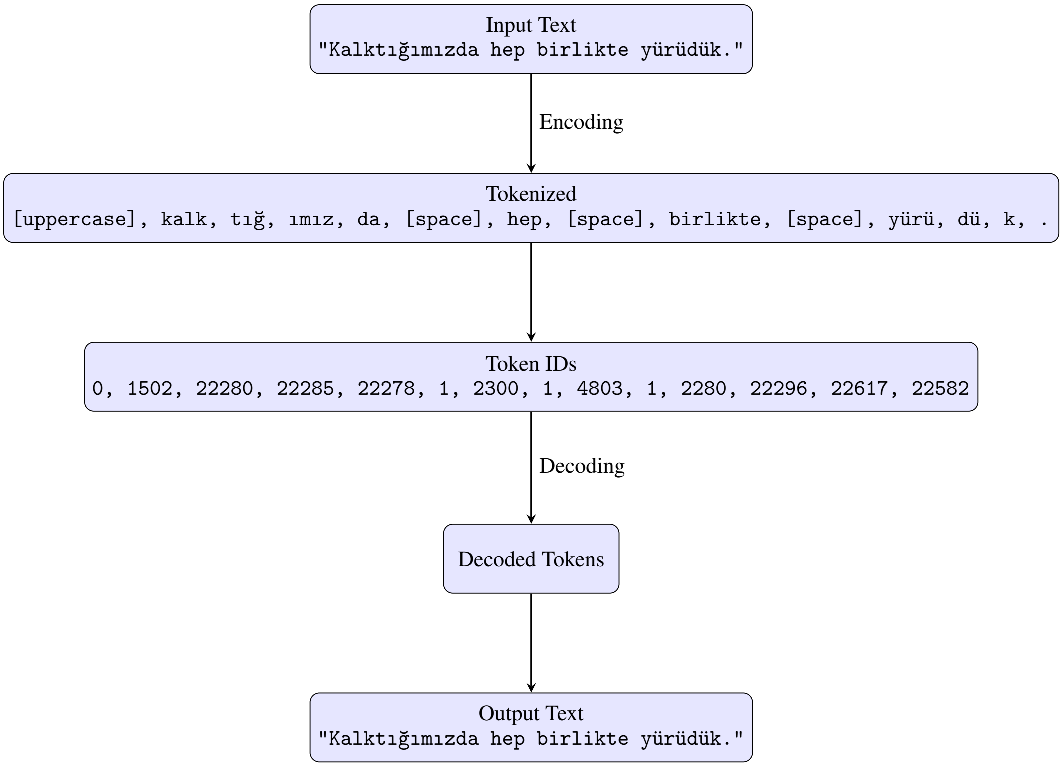
Input Text
"Kalktığımızda hep birlikte yürüdük."
Encoding
Tokenized
[uppercase], kalk, tığ, ımız, da, [space], hep, [space], birlikte, [space], yürü, dü, k, .
Token IDs
0, 1502, 22280, 22285, 22278, 1, 2300, 1, 4803, 1, 2280, 22296, 22617, 22582
Decoding
Decoded Tokens
Output Text
"Kalktığımızda hep birlikte yürüdük."

Figure 2: Encoding and decoding process for the sentence “Kalktığımızda hep birlikte yürüdük.”

Capitalization is restored using a dedicated `[uppercase]` token, which automatically capitalizes the first letter of the following word. Space and punctuation tokens (`[space]`, `.`, etc.) are replaced with their respective characters, maintaining sentence layout. If an unknown or out-of-vocabulary token is encountered, a placeholder is inserted to allow for post-processing or human review.

Consider another example:

Token sequence: `[uppercase], kitap, [space], okuma, yı, [space], sev, i, yor, um, .`

Decoded output: `"Kitap okumayı seviyorum."` (“I like reading books.”)

This process demonstrates how the tokenizer ensures both accuracy and efficiency in text reconstruction, preserving morphological structure while maintaining fluency.

The proposed framework successfully integrates morphological analysis with subword segmentation, creating a robust tokenizer optimized for morphologically complex languages. By balancing linguistic integrity and open-vocabulary coverage, this approach provides a practical tokenizer design for Turkish and a template for similar languages given appropriate resources.

Our downstream evaluation protocol is described in Section 4.1.

4 Results and Analysis

The performance of the proposed morphological tokenizer was evaluated using the TR-MMLU benchmark dataset, which comprises over 1.6 million characters and approximately 200,000 words curated specifically for Turkish [8]. This dataset is designed to reflect the linguistic complexity of Turkish, including its rich morphology, agglutinative structures, and diverse syntactic constructions. As such, it provides a rigorous basis for assessing tokenization quality in morphologically complex languages.

The evaluation compared five different tokenizers: google/gemma-2-9b, meta-llama/Llama-3.2-3B, Qwen/Qwen2.5-7B-Instruct, CohereForAI/aya-expanse-8b, and the proposed turkish_tokenizer. Each tokenizer was assessed using a consistent set of linguistic and computational metrics introduced in [7]. These metrics include total token count, vocabulary size, number of unique tokens, Turkish Token Percentage (TR %), and Pure Token Percentage (Pure %). TR % quantifies the proportion of tokens that correspond to valid Turkish words or morphemes, while Pure % measures the proportion of tokens that fully align with unambiguous root or affix boundaries, thus reflecting morphological integrity.

Table 1: Performance of the proposed turkish_tokenizer on the 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 %)	90.29%
Pure Token Count	9,562
Pure Token Percentage (Pure %)	85.80%

The proposed turkish_tokenizer demonstrated the highest linguistic alignment across all evaluated metrics. It achieved a TR % of 90.29% and a Pure % of 85.80%, substantially outperforming all competing tokenizers. In comparison, google/gemma-2-9b reached a TR % of only 40.96% and a Pure % of 28.49%, indicating that the majority of its tokens do not represent full morphemes. Similarly, meta-llama/Llama-3.2-3B produced a TR % of 45.77% and a Pure % of 31.45%, while Qwen2.5 and aya-expanse achieved TR % values of 40.39% and 53.48%, respectively.

Despite employing significantly smaller vocabulary sizes, the proposed tokenizer demonstrated better linguistic segmentation. With a vocabulary of 32,768 tokens and 11,144 unique tokens used during evaluation, it balanced generalization and expressiveness more effectively than models such as gemma-2-9b and aya-expanse, which rely on vocabularies of over 255,000 tokens. These large-vocabulary tokenizers, rooted in frequency-based subword segmentation, tend to fragment morphologically rich expressions and introduce ambiguity in downstream tasks. In contrast, the morphological awareness of the turkish_tokenizer enables semantically coherent token formation and more consistent syntactic parsing.

Although the total token count generated by the proposed tokenizer (707,727) exceeds those of the other models-for instance, aya-expanse produced 434,526 tokens-this increase is offset by gains in interpretability and linguistic fidelity. High TR % and Pure % scores suggest reduced reliance on spurious subword splits and improved preservation of morphosyntactic structure. This is particularly beneficial for tasks such as syntactic parsing, translation, summarization, and question answering, where semantic consistency across tokens is essential.

These findings support the hypothesis introduced in [7], which argues that high linguistic alignment in tokenization correlates strongly with downstream model performance in morphologically rich and low-resource languages. While conventional subword tokenizers may suffice for high-resource languages like English, they exhibit clear limitations in Turkish unless informed by morphological structure. The results presented here highlight the effectiveness of combining rule-based linguistic analysis with subword strategies to produce tokenizers that are both accurate and efficient in morphologically complex settings.

To illustrate the linguistic fidelity of different tokenization strategies, we present a qualitative comparison using the Turkish sentence:

"Atasözleri geçmişten günümüze kadar ulaşan anlamı bakımından mecazlı bir mana kazanan kalıplaşmış sözlerdir."
("Proverbs are fixed expressions passed down from the past to the present that acquire a metaphorical meaning in terms of their significance.")

This sentence contains a wide range of morphological features, including compound words, multiple derivational and inflectional suffixes, and root forms that undergo phonological alternations. These properties make it an ideal test case for evaluating the morphological sensitivity of different tokenizers.

Proposed Hybrid Tokenizer:
The hybrid morphological tokenizer segments the sentence into linguistically meaningful units with high fidelity. It produces:

```
["<uppercase>", "atasöz", "ler", "i", "<space>", "geçmiş", "ten", "<space>",
"gün", "üm", "üz", "e", "<space>", "kadar", "<space>", "ulaş", "an",
"<space>", "anlam", "ı", "<space>", "bakım", "ın", "dan", "<space>",
"mecaz", "lı", "<space>", "bir", "<space>", "mana", "<space>", "kazan",
"an", "<space>", "kalıp", "laş", "mış", "<space>", "sözle", "r", "dir",
"."]
```

It correctly separates suffixes ("ler", "i", "ın", "dan", "lı", "an", "mış", "dir"), extracts root forms such as "atasöz", "gün", "mana", and employs special tokens like "<uppercase>" and "<space>" to preserve orthographic structure.

Gemma-3:
The tokenizer google/gemma-3 segments the sentence as:

```
["<bos>", "At", "as", "öz", "leri", " geçmiş", "ten", " gün", "ümü", "ze",
" kadar", " ulaş", "an", " anlam", "ı", " bakım", "ından", " mec", "az",
"lı", " bir", " mana", " kaz", "anan", " kal", "ı", "pla", "ş", "mış", "
söz", "lerdir", "."]
```

Although it captures some suffixes like "ten" and "ından", it fragments common roots ("At", "as", "öz" instead of "atasöz") and fails to isolate inner morphemes in forms such as "lerdir" and "kazanan", limiting morphological interpretability.

LLaMA-3.2:
The tokenizer meta-llama/Llama-3.2-3B yields:

```
["<|begin_of_text|>", "At", "as", "öz", "leri", " geçmiş", "ten", " gün",
"ümü", "ze", " kadar", " ", "ula", "ş", "an", " anlam", "ı", " bakımından",
" me", "ca", "z", "lı", " bir", " mana", " kaz", "anan", " kal", "ı", "pla",
"ş", "mış", " söz", "lerdir", "."]
```

This tokenizer combines morphologically valid segments like "bakımından" and "kazanan" with fragmented roots like "At", "as", "öz", creating inconsistency in morpheme alignment.

YTU Turkish GPT-2:
The tokenizer ytu-ce-cosmos/turkish-gpt2-large-750m-instruct-v0.1, trained on Turkish corpora, yields:

```
["At", "as", "öz", "leri", " geçmişten", " günümüze", " kadar", " ulaşan",
" anlamı", " bakımından", " mec", "az", "lı", " bir", " mana", " kazanan",
" kalıp", "laşmış", " söz", "lerdir", "."]
```

Although it still segments "atasözleri" incorrectly, it performs well with forms like "geçmişten", "günümüze", and "bakımından", showing the advantage of Turkish-specific pretraining.

GPT-4o:
The tokenizer gpt-4o-o200k_base generates:

```
["At", "as", "öz", "leri", " geçmiş", "ten", " gün", "ümü", "ze", " kadar",
" ulaş", "an", " anlam", "ı", " bakım", "ından", " mec", "az", "lı", " bir",
" mana", " kaz", "anan", " kal", "ı", "pla", "ş", "mış", " söz", "ler",
"dir", "."]
```

Its segmentation strategy is similar to LLaMA and Qwen-partially aware of Turkish morphemes but limited by frequent over-segmentation of compound and derived forms.

The results presented in this section provide strong empirical support for the hypothesis introduced in the introduction: tokenizers that explicitly incorporate morphological and phonological knowledge of

Turkish can outperform general-purpose models in both segmentation accuracy and linguistic coherence. While most state-of-the-art tokenizers struggle with root-fragmentation, over-segmentation, and inconsistent affix treatment, the proposed hybrid tokenizer consistently identifies morpheme boundaries, preserves semantically meaningful units, and reduces vocabulary redundancy. These findings validate the motivation behind this work: morphologically informed tokenization is essential for robust and interpretable NLP in agglutinative languages like Turkish. The qualitative comparisons presented here illustrate not only the performance gap between general and language-specific tokenizers, but also the need for tokenizer architectures that respect language-internal rules.

4.1 Downstream Task Evaluation

To assess the impact of morphologically informed tokenization on downstream model performance, we evaluated the embeddings produced by models initialized with different tokenizers using three benchmarks: STSb-TR, MTEB-TR, and TurBLiMP. All models were initialized randomly to isolate the effect of tokenization structure from pre-training data.

For STS, we use the Turkish STSb-TR benchmark (`figenfikri/stsb_tr`) consisting of sentence pairs with human similarity ratings on a 0–5 scale [31]. Each model encodes both sentences, we compute cosine similarity between the resulting sentence embeddings, and we report Pearson and Spearman correlation with the normalized gold scores. Throughout this section, correlations are presented as percentages ($\times 100$) for readability.

We evaluated the models on the Turkish STS benchmark (stsb-tr) in a zero-shot setting. The proposed `mft-random-init` model achieved a significantly higher correlation with human judgments compared to other randomly initialized baselines, demonstrating that its structural prior provides a better starting point for capturing semantic similarity.

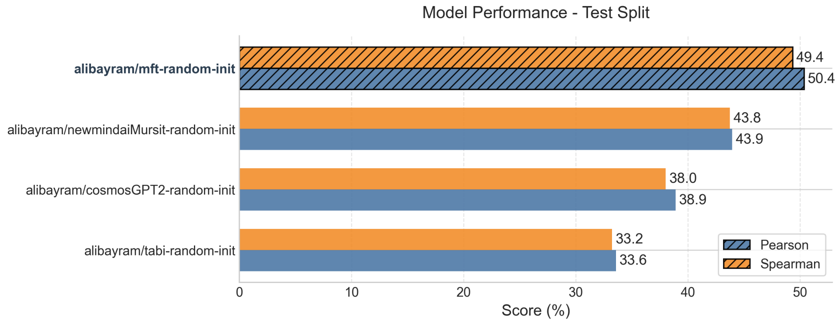

Figure 3: Test split performance on STS Benchmark comparing MFT against Tabi, Cosmos, and Mursit baselines.

The `mft-random-init` model achieved a Pearson correlation of **50.37%** and Spearman correlation of **49.35%**, consistently outperforming all baselines. In comparison, the `newmindaiMursit-random-init` model achieved a Spearman correlation of 43.75%, `cosmosGPT2-random-init` reached 38.02%, and `tabi-random-init` scored 33.24%. These results highlight the advantage of morphological segmentation in preserving semantic density, even without extensive pre-training.

To better understand the learning dynamics, we analyzed the performance evolution of each model across different training checkpoints. Figure 4 and Figure 5 illustrate the Pearson and Spearman correlations respectively at various stages of training (if applicable) or across collected versions. The x-axis represents sequential checkpoints ordered by date.

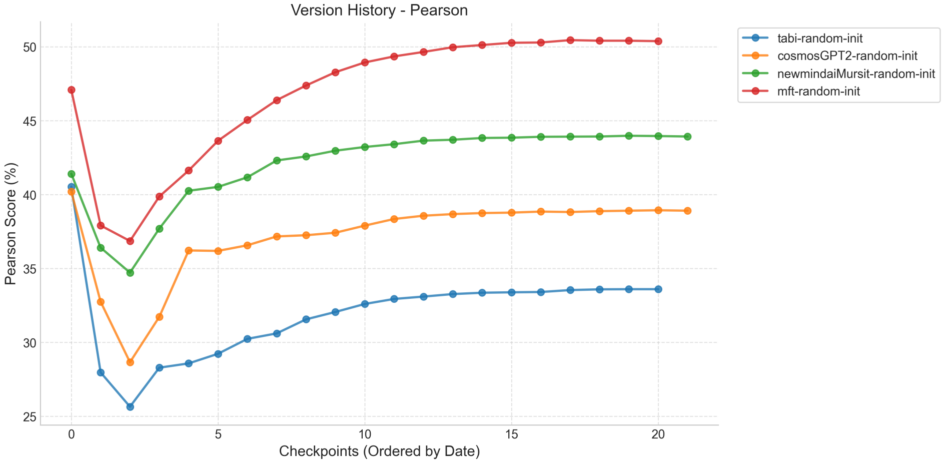

Figure 4: Pearson correlation history comparing model performance across versions.

Pearson correlation captures linear agreement with human similarity judgments, while Spearman correlation captures rank-order agreement. Reporting both is important in STS, since models may preserve relative similarity ordering even when the mapping is not perfectly linear, and conversely small linear gains may not reflect better ranking behavior.

In our version tracking, both correlations show the same qualitative trend: MFT remains ahead of the strongest baselines across revisions, indicating that the downstream improvement is stable rather than a single-run artifact.

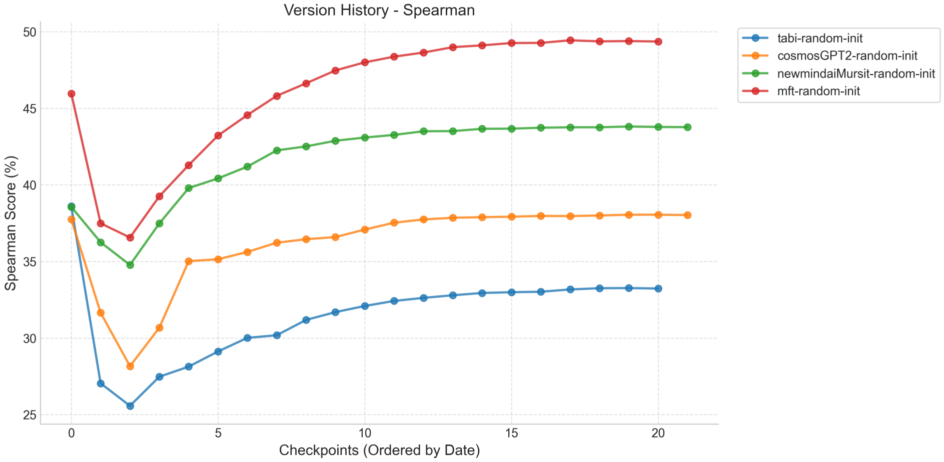

Figure 5: Spearman correlation history comparing model performance across versions.

On the comprehensive MTEB suite, which covers retrieval, classification, clustering, and pair classification tasks, the MFT-based model consistently outperformed other baselines.

Task	**MFT**	**Cosmos**	**Mursit**	**Tabi**
BitextMining				
WMT16BitextMining	**1.53**	1.08	1.26	1.39
Classification				
THYSentimentClassification	51.48	49.74	**52.73**	43.02
TSTimelineNewsCategoryClassification	**50.06**	43.09	44.33	44.09
Turkish75NewsClassification	73.33	78.67	74.67	**79.33**
TurkishIronyClassification	51.25	50.83	**53.33**	52.50
TurkishMovieSentimentClassification	**54.74**	54.37	54.57	53.84
TurkishNewsCategoryClassification	**85.40**	83.32	80.52	79.08
TurkishOffensiveLanguageClassification	**49.87**	48.74	49.71	48.22
TurkishProductSentimentClassification	**54.34**	51.39	51.85	52.10
Clustering				
TurkishColumnWritingClustering	66.30	**66.45**	65.39	65.71
Other				
ArguAnaTR	**7.62**	2.96	3.53	2.56
FiQA2018TR	**6.74**	1.53	2.78	2.38
SCIDOCSTR	0.47	0.27	0.27	**0.74**
Pair Classification				
MnliTr	**48.46**	45.32	45.67	44.98
SnliTr	**44.73**	40.29	40.29	40.04
XNLI	57.55	**58.21**	54.72	57.28
Retrieval				
CQADupstackGamingRetrievalTR	**13.00**	6.84	7.04	6.73
MSMarcoTRRetrieval	**12.84**	6.07	6.13	4.83
NFCorpusTR	**1.22**	0.40	0.51	0.73
QuoraRetrievalTR	**63.01**	46.44	49.24	46.98
SciFactTR	**25.64**	16.33	20.54	15.16
SquadTRRetrieval	**16.53**	8.07	8.93	6.14
TQuadRetrieval	**43.46**	29.97	29.48	26.30
TurkishAbstractCorpusClustering	**47.46**	43.63	42.81	39.69
XQuADRetrieval	**37.33**	23.16	25.38	19.54
STS				
STSbTR	**49.36**	38.04	43.75	33.24

Table 2: Detailed MTEB-TR performance comparison across all tasks. Best scores in bold.

As shown in Table 2, the MFT-based model achieved an average score of **38.99%** across 26 tasks, surpassing Mursit (34.98%), Cosmos (34.43%), and Tabi (33.33%). The advantages are particularly pronounced in Retrieval and Pair Classification tasks, where morphological coherence aids in matching semantic intent.

TurBLiMP evaluates the model's sensitivity to specific linguistic phenomena such as case marking, agreement, and word order. Table 3 presents the accuracy of each model in distinguishing grammatical from ungrammatical sentences across various linguistic categories.

Linguistic Phenomenon	Mursit-Random	MFT-Random	Cosmos-Random	Tabi-Random
Ellipsis	98.0%	**99.2%**	97.7%	93.0%
Scrambling	97.7%	**98.0%**	97.3%	**98.0%**
Determiners	94.9%	**96.5%**	93.4%	94.1%
Quantifiers	90.2%	90.2%	86.3%	**94.1%**
Suspended Affixation	89.5%	**93.4%**	83.2%	91.0%
Relative Clauses	89.1%	**96.1%**	86.3%	93.0%
Binding	88.7%	89.5%	85.5%	**94.5%**
Anaphor Agreement	87.9%	**92.6%**	84.8%	**92.6%**
Npi Licensing	87.5%	87.5%	82.4%	**89.5%**
Irregular Forms	87.1%	90.6%	80.1%	**92.6%**
Argument Structure Ditransitive	86.3%	93.4%	80.9%	**93.8%**
Subject Verb Agreement	84.8%	**89.5%**	79.7%	86.3%
Nominalization	82.8%	87.5%	79.3%	**89.8%**
Argument Structure Transitive	81.6%	90.6%	76.2%	**91.0%**
Passives	81.6%	85.5%	79.3%	**90.6%**
Island Effects	79.7%	**84.0%**	76.2%	**84.0%**

Table 3: Detailed TurBLiMP sensitivity scores comparison across all models.

The MFT tokenizer demonstrates superior handling of complex morphological features such as *Scrambling* (98.0%), *Relative Clauses* (96.1%), and *Anaphor Agreement* (92.6%). These results confirm that explicit morphological modeling allows the model to better generalize over complex syntactic dependencies.

Finally, we tracked STS performance across multiple experiment iterations and code revisions to ensure the observed gains are not a one-off artifact; the best observed revision reached 76.10% Pearson on STSb-TR, with results remaining stable across recent runs.

5 Future Work

This study highlights the importance of linguistic integrity and computational efficiency in tokenization, presenting a framework to guide the development of tokenizers optimized for morphologically rich and low-resource languages. Despite these promising results, much work remains to unlock the full potential of tokenizers. Future improvements will focus on incorporating advanced morphological analysis steps, which will further enhance their capability to capture the rich grammatical and semantic structures of Turkish. These steps may include integrating more sophisticated linguistic rules, handling rare morphemes, and accounting for contextual variations that impact tokenization in complex languages. Such enhancements will not only improve linguistic fidelity but also expand the scope of the tokenizers for diverse NLP applications.

Additionally, future work will explore iterative refinement processes, such as dynamic token generation based on downstream tasks and domain-specific requirements. For instance, the tokenizers could be built for specific domains like medical, legal, or technical texts to ensure high performance in specialized applications. Moreover, incorporating unsupervised and semi-supervised learning approaches into the tokenizer development process will help address gaps in morphological and semantic coverage.

Future work will also explore adapting the tokenizer to additional languages. Extending the approach beyond Turkish requires constructing language-specific lexical resources (e.g., root and affix inventories) and corresponding decoding and normalization rules, and validating the resulting tokenizers on language-appropriate benchmarks.

Although still in the early stages of development, these tokenizers provide a strong foundation for further innovation. Their initial performance gives hope that, with targeted improvements, they can evolve into robust, versatile tools for tokenizing morphologically rich languages. By implementing these additional steps and conducting further evaluations across languages and tasks, this research aims to establish a new standard for linguistically informed tokenization, ultimately advancing the quality and efficiency of language models in a wide array of applications.

6 Conclusion

We presented a linguistically informed, morphology-first hybrid tokenizer designed for Turkish and similar agglutinative languages. The tokenizer combines curated root and affix lexicons with phonological normalization (mapping surface allomorphs to shared identifiers) and a controlled subword fallback for coverage. This design aims to produce token sequences that more closely align with morpheme boundaries while remaining practical for large-scale NLP pipelines.

On TR-MMLU, the proposed tokenizer achieves 90.29% Turkish Token Percentage (TR %) and 85.80% Pure Token Percentage (Pure %), indicating substantially stronger morpheme-level alignment than several general-purpose tokenizers. We additionally report downstream sentence embedding evaluation on Turkish STS and MTEB-TR using **randomly initialized** models to isolate tokenizer effects from pretrained knowledge. The MFT-based model reaches **50.37%** Pearson correlation on STSb-TR, compared to 33.58% for the Tabi baseline—a gain of **+16.79 percentage points**. On MTEB-TR, MFT achieves 38.99% overall average compared to 33.33% for Tabi (+5.66 points). These substantial gaps demonstrate that morphology-first tokenization provides a stronger inductive bias for learning Turkish semantic representations from scratch.

We emphasize that empirical claims in this paper are Turkish-focused. We outline concrete next steps—improved morphophonological handling, better capitalization edge cases, and standardized efficiency measurements—in Section 5.

References

[1] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, July 2019. arXiv:1907.11692 [cs].

[2] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units, June 2016. arXiv:1508.07909 [cs].

[3] Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152, March 2012.

[4] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing, August 2018. arXiv:1808.06226 [cs].

[5] Cagri Toraman, Eyup Halit Yilmaz, Furkan Şahinuç, and Oguzhan Ozcelik. Impact of tokenization on language models: An analysis for turkish. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(4):1–21, April 2023. arXiv:2204.08832 [cs].

[6] Yiğit Bekir Kaya and A. Cüneyd Tantuğ. Effect of tokenization granularity for turkish large language models. *Intelligent Systems with Applications*, 21:200335, March 2024.

[7] M. Ali Bayram, Ali Arda Fincan, Ahmet Semih Gümüş, Sercan Karakaş, Banu Diri, and Savaş Yıldırım. Tokenization standards for linguistic integrity: Turkish as a benchmark, February 2025. arXiv:2502.07057 [cs].

[8] M. Ali Bayram, Ali Arda Fincan, Ahmet Semih Gümüş, Banu Diri, Savaş Yıldırım, and Öner Aytaş. Setting standards in turkish nlp: Tr-mmlu for large language model evaluation, January 2025. arXiv:2501.00593 [cs].

[9] Ezgi Başar, Francesca Padovani, Jaap Jumelet, and Arianna Bisazza. TurBLiMP: A Turkish Benchmark of Linguistic Minimal Pairs. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 16495–16510, Suzhou, China, November 2025. Association for Computational Linguistics.

[10] Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates, April 2018. arXiv:1804.10959 [cs].

[11] Batuhan Baykara and Tunga Güngör. Abstractive text summarization and new large-scale datasets for agglutinative languages turkish and hungarian. *Language Resources and Evaluation*, 56(3):973–1007, September 2022.

[12] Yirong Pan, Xiao Li, Yating Yang, and Rui Dong. Morphological word segmentation on agglutinative languages for neural machine translation, January 2020. arXiv:2001.01589 [cs].

[13] Matthias Huck, Simon Riess, and Alexander Fraser. Target-side word segmentation strategies for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 56–67, September 2017.

[14] Nihal Zuhal Kayalı and Sevinç İlhan Omurca. Hybrid tokenization strategy for turkish abstractive text summarization. In *2024 8th International Artificial Intelligence and Data Processing Symposium (IDAP)*, pages 1–6, September 2024.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, May 2019. arXiv:1810.04805 [cs].

[16] Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. Towards the systematic reporting of the energy and carbon footprints of machine learning, November 2022. arXiv:2002.05651 [cs].

[17] Pedro Henrique Martins, Patrick Fernandes, João Alves, Nuno M. Guerreiro, Ricardo Rei, Duarte M. Alves, José Pombal, Amin Farajian, Manuel Faysse, Mateusz Klimaszewski, Pierre Colombo, Barry Haddow, José G. C. de Souza, Alexandra Birch, and André F. T. Martins. Eurollm: Multilingual language models for europe, September 2024. arXiv:2409.16235 [cs].

[18] Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. Not all tokens are what you need for pretraining. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc.

[19] Martin Neubeck. Flexible byte-pair tokenizers for dynamic vocabulary. Technical Report, 2024.

[20] Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. Superbizarre is not superb: Derivational morphology improves bert's interpretation of complex words. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 3594–3608, 2021.

[21] Haris Jabbar. Morphpiece: A linguistic tokenizer for large language models, February 2024. arXiv:2307.07262 [cs].

[22] Elshad Rahimov. miLLi: Model Integrating Local Linguistic Insights for Morphologically Robust Tokenization, December 2025.

[23] Ehsaneddin Asgari, Yassine El Kheir, and Mohammad Ali Sadraei Javaheri. MorphBPE: A Morpho-Aware Tokenizer Bridging Linguistic Complexity for Efficient LLM Training Across Morphologies, February 2025. arXiv:2502.00894 [cs].

[24] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[25] Mete Ismayilzada, Defne Circi, Jonne Sälevä, Hale Sirin, Abdullatif Köksal, Bhuwan Dhingra, Antoine Bosselut, Duygu Ataman, and Lonneke van der Plas. Evaluating morphological compositional generalization in large language models, February 2025. arXiv:2410.12656 [cs].

[26] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, January 2021. arXiv:2009.03300 [cs].

[27] Melikşah Türker, A. Ebrar Kızıloğlu, Onur Güngör, and Susan Üsküdarlı. TabiBERT: A Large-Scale ModernBERT Foundation Model and A Unified Benchmark for Turkish, January 2026. arXiv:2512.23065 [cs].

[28] Mohamed Rashad. Arabic-nougat: Fine-tuning vision transformers for arabic ocr and markdown extraction, November 2024. arXiv:2411.17835 [cs].

[29] Rudolf Rosa and Ivana Kvapilíková. A benchmark for scandinavian tokenizers. Technical Report, 2024.

[30] Philip Gage. A new algorithm for data compression, 1994.

[31] Figen Beken Fikri, Kemal Oflazer, and Berrin Yanikoglu. Semantic Similarity Based Evaluation for Abstractive News Summarization. In *Proceedings of the First Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 24–33, Online, August 2021. Association for Computational Linguistics.