Tokens with Meaning: A Hybrid Tokenization Approach for NLP

Anonymous ACL submission

Abstract

Tokenization plays a pivotal role in natural language processing (NLP), shaping how textual data is segmented, interpreted, and processed by language models. Despite the success of subword-based tokenization techniques such as Byte Pair Encoding (BPE) and WordPiece, these methods often fall short in morphologically rich and agglutinative languages due to their reliance on statistical frequency rather than linguistic structure. This paper introduces a linguistically informed hybrid tokenization framework that integrates rule-based morphological analysis with statistical subword segmentation to address these limitations. The proposed approach leverages phonological normalization, root-affix dictionaries, and a novel tokenization algorithm that balances morpheme preservation with vocabulary efficiency. It assigns shared identifiers to phonologically variant affixes (e.g., *-ler* and *-lar*) and phonologically altered root forms (e.g., *kitap* vs. *kitabı*), significantly reducing redundancy while maintaining semantic integrity. The framework also incorporates special tokens for whitespace and orthographic case, including an `<uppercase>` token to prevent vocabulary inflation from capitalization. Byte Pair Encoding is integrated to support out-of-vocabulary coverage without compromising morphological coherence. Evaluation on the TR-MMLU benchmark—a large-scale, Turkish-specific NLP benchmark—demonstrates that the proposed tokenizer achieves the highest Turkish Token Percentage (90.29%) and Pure Token Percentage (85.8%) among all tested models. Comparative analysis against widely used tokenizers from models such as LLaMA, Gemma, and OpenAI's GPT reveals that the proposed method yields more linguistically meaningful and semantically coherent tokens. A qualitative case study further illustrates improved morpheme segmentation and interpretability in complex Turkish sentences. Although the implementation focuses on Turkish, the underlying methodology is language-independent and adaptable to other languages. This work contributes to ongoing efforts to improve tokenizer design through linguistic alignment, offering a practical and extensible solution for enhancing both interpretability and performance in multilingual NLP systems.

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

1 Introduction

Tokenization is a foundational step in Natural Language Processing (NLP), directly impacting vocabulary construction, model efficiency, and downstream task performance (Liu et al., 2019). While subword-based methods like Byte Pair Encoding (BPE) (Sennrich et al., 2016) and WordPiece (Schuster and Nakajima, 2012) effectively handle out-of-vocabulary (OOV) words in high-resource languages, they often disregard the linguistic structure of morphologically rich languages. In agglutinative languages like Turkish, Finnish, and Hungarian, words are formed by appending multiple affixes to a root, resulting in complex surface forms. Frequency-based subword models frequently violate morphemic boundaries in these languages, reducing semantic coherence and interpretability (Toraman et al., 2023).

Turkish poses specific challenges due to its agglutinative nature and phonological processes such as vowel harmony and consonant alternation. Words are formed by appending multiple affixes to a root, producing an expansive set of surface forms. For instance, the single word *Avrupalılaştıramadıklarımızdanmışsınızcasına* ("as if you were one of those whom we could not make resemble a European") conveys a meaning that requires an entire sentence in English. Standard tokenizers often treat these variants as distinct or fragment them inconsistently, leading to vocabulary redundancy and poor alignment with linguistic units (Bayram et al., 2025b). Recent benchmarks, such as TR-MMLU (Bayram et al., 2025a), indicate that "token purity"—the alignment of tokens with morphemes—correlates strongly with model performance.

To address these limitations, we introduce a linguistically informed hybrid tokenization framework. Our approach integrates rule-based morphological segmentation with BPE to ensure both linguistic fidelity and broad coverage. Key innovations include: (1) **Phonological Normalization**, mapping surface variants (e.g., *-dAn*, *-tAn*) to unified token IDs; (2) **Orthographic Encoding**, using a special `<uppercase>` token to handle case without vocabulary duplication; and (3) **Hybrid Fallback**, using BPE only for stems not covered by the morphological dictionary.

We evaluate our tokenizer on the TR-MMLU benchmark, demonstrating significantly higher Turkish Token Percentage (TR %) and Pure Token Percentage (Pure %) compared to state-of-the-art models like LLaMA, Gemma, and Qwen. These results validate that morphologically aware tokenization yields more semantically meaningful and syntactically coherent representations, offering a pathway to more efficient and equitable multilingual NLP systems.

2 Related Work

Tokenization significantly impacts model performance, especially in morphologically rich languages (Toraman et al., 2023). While subword methods like BPE (Sennrich et al., 2016) and WordPiece (Schuster and Nakajima, 2012) are standard, they often fail to capture the agglutinative structure of languages like Turkish, Finnish, and Hungarian (Baykara and Güngör, 2022).

Early Turkish NLP relied on rule-based morphological analyzers like Zemberek (Akın and Akın, 2007), which offered precise segmentation but lacked the scalability required for modern LLMs. Recent research has sought to bridge this gap. Toraman et al. (2023) showed that morphological tokenization could recover 97% of BERT's performance with a fraction of the model size. Similarly, Pan et al. (2020) and Huck et al. (2017) demonstrated that morphology-aware segmentation improves Neural Machine Translation (NMT) by reducing data sparsity.

Hybrid approaches combining rule-based and statistical methods have shown promise. Kayalı and Omurca (2024) used a hybrid tokenizer for Turkish NER and summarization, finding benefits in preserving linguistic structure. Jabbar (2024) introduced MorphPiece for English, achieving superior performance with smaller vocabularies. However, challenges remain in balancing vocabulary size, sequence length, and computational efficiency (Henderson et al., 2022; Kaya and Tantuğ, 2024). Our work extends these efforts by introducing a fully hybrid pipeline that integrates phonological normalization directly into the tokenization process.

Despite the theoretical appeal of morphological segmentation, some studies argue that strict linguistic boundaries may not always be optimal for neural models. Kudo (2018) introduced "subword regularization," suggesting that exposing models to multiple segmentations of the same word (including non-canonical ones) can improve robustness and generalization. Similarly, recent large-scale multilingual models often favor larger, data-driven vocabularies that maximize compression rates over linguistic purity. This "byte-premium" hypothesis suggests that minimizing the number of tokens per word is the primary driver of cross-linguistic performance gaps (Martins et al., 2024).

Furthermore, the rise of massive multilingual LLMs presents a dilemma for language-specific tokenization. While a dedicated Turkish tokenizer offers superior alignment for Turkish text, it may not be easily integrated into a model trained on 100+ languages without significantly increasing the combined vocabulary size or complicating the embedding space. Our work acknowledges this tension but argues that for high-stakes or specialized applications in morphologically rich languages, the benefits of semantic coherence and interpretability outweigh the costs of language-specific engineering.

3 Methodology

Traditional subword tokenization methods like BPE (Sennrich et al., 2016) and WordPiece (Schuster and Nakajima, 2012) often fail to capture the rich internal structure of agglutinative languages. For example, the Turkish word *anlayabildiklerimizden* is composed of multiple morphemes (*anla-yabil-dik-ler-imiz-den*). Standard tokenizers fragment such words arbitrarily, obscuring grammatical function. To address this, we propose a hybrid tokenization framework that prioritizes morphological segmentation while retaining BPE as a fallback for robustness.

3.1 Dictionary Construction

The foundation of our hybrid tokenizer is a curated set of morphological dictionaries derived from open-source linguistic resources, primarily the Zemberek NLP framework (Akın and Akın, 2007) and the Turkish Language Association (TDK) data.

1. **Root Dictionary (`kokler.json`):** We extracted approximately 22,000 high-frequency Turkish roots (nouns and verbs) from a large-scale corpus of Turkish web text. These roots were filtered to exclude rare or archaic terms that would unnecessarily inflate the vocabulary. Special control tokens (`<uppercase>`, `<unknown>`, `<pad>`, `<eos>`) were added to support model training.

2. **Suffix Dictionary (`ekler.json`):** This dictionary contains 230 distinct derivational and inflectional suffixes. Crucially, we applied phonological abstraction: suffixes that are phonetically distinct but functionally identical (allomorphs) are mapped to a single canonical ID. For example, the plural suffix forms *-lar* and *-ler* share the same token ID, as do the four variants of the accusative case (*-ı*, *-i*, *-u*, *-ü*).

3. **BPE Vocabulary (`bpe_tokenler.json`):** To ensure full coverage for foreign words, proper names, and rare scientific terms, we trained a Byte Pair Encoding (BPE) model on the same corpus with a vocabulary limit of 10,000 subwords. This serves as a fallback mechanism for any segment not found in the root or suffix dictionaries.

3.2 Encoding Process

The encoding phase converts raw text into a sequence of token IDs using a "Longest Prefix Match" algorithm with a strict priority hierarchy: **Roots** $\gg$ **Suffixes** $\gg$ **BPE**. This design ensures that linguistically valid morphemes are always preferred over statistical subwords.

3.2.1 Handling Special Cases

- **Case Sensitivity:** Unlike standard lowercasing, we preserve case information using a special `<uppercase>` token. This token is inserted immediately before any token that was originally capitalized. This approach allows the model to distinguish between proper nouns (e.g., *Ayşe*) and common nouns (e.g., *ayşe*) without duplicating every word in the vocabulary.

- **Acronyms and CamelCase:** Words with mixed casing or all-uppercase acronyms (e.g., *TBMM*, *iPhone*) are first split into segments based on case transitions. Each segment is then tokenized individually. For example, *HTTPServer* is split into *HTTP* and *Server*, which are then processed by the BPE fallback if they are not in the root dictionary.

- **Compound Words:** Lexicalized compounds (e.g., *hanımeli*, *bilgisayar*) are treated as single roots if they appear in the root dictionary. Novel or transparent compounds are naturally segmented into their constituent roots and suffixes by the longest-prefix match algorithm.

3.3 Decoding and Phonological Resolution

The **Decoding** phase reconstructs text from token IDs using a "Single ID, Multiple Views" principle. Since multiple surface forms (allomorphs) map to a single ID, the decoder must dynamically resolve the correct form based on context. This process is critical for generating natural-sounding Turkish text.

```
Algorithm: Hybrid Encoding
Input: text string T
Output: list of token IDs

1. Split T into words by whitespace.
2. For each word W:
    a. Identify uppercase positions.
    b. Split W into segments (handle camelCase).
    c. For each segment S:
        i.   Check ROOTS for longest prefix match.
             If match: add ID, continue.
        ii.  Check SUFFIXES for longest prefix match.
             If match: add ID, continue.
        iii. Check BPE for longest prefix match.
             If match: add ID, continue.
        iv.  Else: add <unknown> ID.
    d. Insert <uppercase> tokens based on (a).
3. Return list of IDs.
```

Figure 1: Pseudocode for the hybrid encoding process.

3.3.1 Root Resolution (Lookahead)

Roots susceptible to alternation are stored with their canonical form but can be modified based on the following suffix.

- **Vowel Softening:** Roots ending in *k, p, ç, t* may soften to *ğ, b, c, d* when followed by a vowel. For example, the root *kitap* (book) is tokenized as a single ID. If the next token is the accusative suffix *-ı*, the decoder outputs *kitabı* instead of *kitapı*.
- **Vowel Dropping:** Some roots lose a vowel when a suffix is added. For instance, *akıl* (mind) + *-ı* becomes *aklı*. The decoder checks the root type and the incoming suffix to apply this transformation.

3.3.2 Suffix Resolution (Lookbehind)

Suffixes are stored as abstract templates (e.g., *-lAr* for plural) and are instantiated based on the phonological properties of the preceding token.

- **2-Way Vowel Harmony (A-Type):** Suffixes containing *a/e* (e.g., plural *-lar/-ler*) select the vowel based on the last vowel of the previous token.
  - Back vowels (*a, ı, o, u*) → *-lar* (e.g., *arabalar*)
  - Front vowels (*e, i, ö, ü*) → *-ler* (e.g., *evler*)
- **4-Way Vowel Harmony (I-Type):** Suffixes containing high vowels (e.g., accusative *-ı/-i/-u/-ü*) select from four variants based on roundness and backness.
  - *a, ı* → *-ı* (e.g., *kapı-yı*)
  - *e, i* → *-i* (e.g., *kedi-yi*)
  - *o, u* → *-u* (e.g., *okul-u*)
  - *ö, ü* → *-ü* (e.g., *gül-ü*)
- **Consonant Hardening:** Suffixes starting with *c, d, g* (e.g., locative *-da*) harden to *ç, t, k* if the previous token ends in a voiceless consonant (F, S, T, K, Ç, Ş, H, P). For example, *sokak* + *-da* → *sokakta.*

This dynamic resolution allows the vocabulary to remain compact while generating linguistically correct surface forms, effectively decoupling the model's internal representation from the surface complexity of the language.

4 Results and Analysis

We evaluated the proposed tokenizer on the TR-MMLU benchmark (Bayram et al., 2025a), comparing it against five state-of-the-art tokenizers: `google/gemma-2-9b`, `meta-llama/Llama-3.2-3B`, `Qwen/Qwen2.5-7B-Instruct`, `CohereForAI/aya-expanse-8b`, and `microsoft/phi-4`. Metrics include Vocabulary Size, Total Tokens, Unique Tokens, Turkish Token Percentage (TR %), and Pure Token Percentage (Pure %).

Table 1: Performance comparison on the TR-MMLU dataset.

Tokenizer	Vocab	Tokens	Unique	TR %	Pure %
`turkish_tokenizer` (Ours)	32k	707k	11.1k	**90.29**	**85.80**
`google/gemma-2-9b`	256k	-	-	40.96	28.49
`meta-llama/Llama-3.2-3B`	128k	-	-	45.77	31.45
`Qwen/Qwen2.5-7B-Instruct`	152k	-	-	40.39	-
`CohereForAI/aya-expanse-8b`	256k	434k	-	53.48	-

As shown in Table 1, our tokenizer achieves the highest linguistic alignment (90.29% TR %, 85.80% Pure %) despite having a significantly smaller vocabulary (32k vs. 256k). While the total token count is higher (707k vs. 434k for Aya), this reflects a granular segmentation that respects morpheme boundaries rather than arbitrary subword merges.

4.1 Qualitative Analysis

To illustrate the difference in segmentation strategies, we analyzed two sentences with varying morphological complexity.

Example 1: *"Atasözleri geçmişten günümüze kadar ulaşan anlamı bakımın-*

dan mecazlı bir mana kazanan kalıplaşmış sözlerdir."

- **Proposed Tokenizer:** Correctly identifies roots (*atasöz, gün, mana*) and separates suffixes (*-ler, -i, -ten, -üm, -üz, -e*). It preserves the internal structure of complex words like *kalıplaşmış* (*kalıp-laş-mış*).
- **Baselines:** Frequently fragment roots (e.g., *At-as-öz* instead of *atasöz*) or merge distinct morphemes into opaque subwords (e.g., *bakımından* as a single token). This over-segmentation of roots and under-segmentation of affixes hinders the model's ability to generalize across morphologically related forms.

Example 2: *"Çekoslovakyalılaştıramadıklarımızdan mısınız?"* (Are you one of those whom we could not make resemble a Czechoslovakian?)

- **Proposed Tokenizer:** `["Çekoslovakya", "lı", "laş", "tır", "ama", "dık", "lar", "ımız", "dan", " ", "mı", "sınız", "?"]` The tokenizer successfully decomposes this famous agglutinative tongue-twister into its constituent morphemes. The root *Çekoslovakya* is identified, followed by the derivational suffixes *-lı* (from), *-laş* (become), *-tır* (causative), and the negation *-ama*.
- **Gemma-2:** `["Çek", "os", "lo", "vak", "yalı", "laş", "tı", "ra", "ma", "dık", "la", "rı", "mız", "dan", ...]` The baseline fails to recognize the proper noun root and fragments the suffixes into arbitrary syllables (*os, lo, vak*), destroying the semantic compositionality of the word.

4.2 Token Fertility and Efficiency

A key trade-off in tokenization is between vocabulary size and sequence length (fertility). As shown in Table 1, our tokenizer generates approximately 63% more tokens than `aya-expanse` (707k vs. 434k). This increased fertility is an expected consequence of granular morphological segmentation.

- **Vocabulary Efficiency:** Our tokenizer uses a compact vocabulary of 32k, which is 8× smaller than Gemma's 256k. This significantly reduces the embedding layer parameters, potentially lowering the model's memory footprint.
- **Sequence Length:** The higher token count implies longer input sequences for the same text. While this increases the computational cost of attention mechanisms (which scale quadratically with length), it provides the model with a more explicit and regular representation of the language. For agglutinative languages, this trade-off is often beneficial, as the model does not need to memorize millions of surface forms (e.g., *geldim, geldin, geldi...*) as distinct vocabulary items, but can instead learn the compositional rules of the grammar.

These results confirm that frequency-based tokenizers, even with large vocabularies, struggle with the agglutinative structure of Turkish. Our hybrid approach, by contrast, yields tokens that are linguistically meaningful and consistent.

5 Discussion

The results presented in this study highlight a fundamental tension in tokenizer design: the trade-off between vocabulary compactness and morphological fidelity. By prioritizing linguistic structure, our hybrid tokenizer achieves significantly higher alignment with Turkish morphology than standard BPE-based models, but at the cost of increased sequence length.

5.1 The Efficiency-Expressivity Trade-off

Our tokenizer generates approximately 63% more tokens than the Aya-Expanse tokenizer for the same text. In the context of Transformer-based LLMs, where attention complexity scales quadratically with sequence length ($O(N^2)$), this increase implies a higher computational cost during inference. However, this cost must be weighed against the benefits of "expressivity." A model using our tokenizer does not need to learn that *geldim*, *geldin*, and *geldi* are separate entities; it can compositionally derive their meanings from the root *gel-* and the respective suffixes. We hypothesize that this compositional representation could lead to faster convergence during training and better generalization to unseen word forms, effectively shifting the complexity from the vocabulary (memory) to the sequence (compute).

5.2 Implications for Multilingual Models

Current multilingual models (e.g., LLaMA-3, Gemma-2) predominantly use large, shared vocabularies (128k-256k) to cover many languages. While efficient for high-resource languages like English, this approach often treats low-resource, agglutinative languages as "second-class citizens," allocating them fewer dedicated tokens and relying on fragmented subword sequences. Our findings suggest that a modular tokenization approach—where language-specific morphological adapters are swapped in during pretraining or fine-tuning—could be a viable alternative. Instead of a single monolithic vocabulary, a "mixture-of-tokenizers" architecture could allow models to process each language in its most natural structural form.

5.3 Generalizability to Other Languages

While this study focuses on Turkish, the principles of our hybrid framework are directly applicable to other agglutinative languages such as Finnish, Hungarian, and Estonian. These languages share the core properties of rich suffixation, vowel harmony, and extensive compounding. The "Longest Prefix Match" algorithm with phonological abstraction is language-agnostic; adapting it to Finnish, for example, would primarily require replacing the root and suffix dictionaries and defining the specific phonological rules (e.g., vowel harmony groups) for that language. This portability is a key advantage over purely statistical methods, which require retraining on massive corpora to "discover" these rules implicitly.

6 Conclusion

In this study, we introduced a linguistically informed hybrid tokenization framework designed to address the challenges of morphologically rich languages. By integrating rule-based morphological analysis with BPE, our approach preserves morpheme boundaries and minimizes vocabulary redundancy. Empirical evaluations on the TR-MMLU dataset demonstrate that our tokenizer achieves significantly higher linguistic alignment (90.29% TR %, 85.80% Pure %) compared to state-of-the-art multilingual models like LLaMA, Gemma, and Qwen. These results validate that incorporating linguistic structure into tokenization yields more semantically coherent representations.

6.1 Limitations

While our results are promising, this study has several limitations. First, our evaluation relies primarily on intrinsic metrics (TR % and Pure %). Due to computational constraints, we did not pretrain a language model from scratch to empirically verify the impact of our tokenizer on downstream task performance (e.g., perplexity, classification accuracy). Establishing a causal link between token purity and model performance remains a critical next step. Second, our current implementation is in Python, which may not match the inference speed of highly optimized Rust-based tokenizers used in production LLMs. We have not yet conducted rigorous benchmarking of processing time or memory usage. Third, the approach relies on manually curated dictionaries, which may require maintenance to cover evolving language use and neologisms.

6.2 Future Work

Future research will focus on three key areas: (1) **Downstream Evaluation:** Training a small-scale language model (e.g., 100M parameters) using our tokenizer to measure improvements in perplexity and task-specific accuracy compared to standard BPE; (2) **Optimization:** Re-implementing the tokenizer in Rust to ensure it meets the latency requirements of real-time applications; and (3) **Generalization:** Extending the framework to other agglutinative languages such as Finnish and Hungarian to test the cross-linguistic validity of our hybrid approach.

References

Mehmet Dündar Akın and Ahmet Afşin Akın. 2007. Türk Dilleri İçin Açık Kaynaklı Doğal Dil İşleme Kütüphanesi: ZEMBEREK. *elektrik mühendisliği*, 431:38–44.

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

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

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

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

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

Haris Jabbar. 2024. Morphpiece: A linguistic tokenizer for large language models. *arXiv preprint*. ArXiv:2307.07262 [cs].

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

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

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. *arXiv preprint*. ArXiv:1804.10959 [cs].

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

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. 2024. Eurollm: Multilingual language models for europe. *arXiv preprint*. ArXiv:2409.16235 [cs].

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

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

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

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