Tokens with Meaning: A Hybrid Tokenization Approach for NLP

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 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, the process of segmenting text into smaller linguistic units called tokens, is a foundational step in Natural Language Processing (NLP). It has a direct impact on vocabulary construction, model efficiency, semantic interpretation, and the overall performance of downstream tasks such as question answering, sentiment analysis, and machine translation(Liu et al., 2019). While traditional tokenization techniques—such as whitespace or rule-based segmentation—have been commonly used in early NLP systems, they fall short in modeling the complex morphological phenomena of many languages, particularly those that exhibit agglutination, inflectional variation, and phonological alternation.

Subword-based tokenization methods like Byte Pair Encoding (BPE) (Sennrich et al., 2016), WordPiece (Schuster and Nakajima, 2012), and Unigram (Kudo and Richardson, 2018) have become the de facto standard in transformer-based language models such as BERT (Devlin et al., 2019) and GPT (Radford et al., 2019). These methods address the out-of-vocabulary (OOV) problem by segmenting rare words into frequently occurring subword units, thereby balancing vocabulary size and generalization. However, despite their computational strengths, these frequency-based methods often disregard the linguistic structure of words. As a result, morphologically rich languages such as Turkish, Finnish, and Hungarian are frequently segmented in ways that violate morphemic boundaries, reducing

semantic coherence and interpretability (Baykara and Güngör, 2022; Toraman et al., 2023).

Agglutinative languages like Turkish pose specific challenges for tokenization. Words are formed by appending multiple affixes to a root, producing an expansive set of surface forms that differ only in morphological features. Phonological processes such as vowel harmony and consonant alternation further increase the diversity of surface realizations. For instance, plural suffixes like *-lAr* or ablative markers like *-dAn*, *-tAn*, functionally represent the same morphemes but differ based on the phonological context. Similarly, root alternations like *kitap* → *kitab (book)* and *göğüs* → *göğs (chest)* are common in Turkish. Frequency-based subword models fail to account for such variation, resulting in redundant and inconsistent tokenization (Bayram et al., 2025b).

Tokenization approaches that ignore these morphological and phonological nuances lead to increased vocabulary size, fragmented representation of morphosyntactic units, and reduced performance in syntactically dependent tasks. Recent benchmark studies, including TR-MMLU (Bayram et al., 2025a) and a cross-model tokenizer evaluation (Bayram et al., 2025b), have shown that metrics such as Turkish Token Percentage (TR %) and Pure Token Percentage (Pure %) strongly correlate with downstream model performance. These findings underscore the necessity of tokenization strategies that align with linguistic structures.

Token purity plays a critical role in the effectiveness of large language models, particularly when applied to morphologically complex languages like Turkish. Since LLMs are fundamentally statistical pattern learners, the quality and clarity of those patterns directly influence their ability to generalize, reason, and generate coherent outputs. Pure tokens—those that cleanly align with complete morphemes such as roots or affixes—provide semantically and syntactically consistent input signals. This allows models to recognize grammatical structures, identify morphological relationships, and transfer learned behavior across different word forms (e.g., *kitap*, *kitabı*, *kitaplık*). In contrast, impure tokens—subword units that contain partial or blended morphemes—introduce ambiguity into the token stream. Such noise disrupts the alignment between token boundaries and linguistic meaning, hindering the model's ability to learn reliable representations.

Empirical studies have shown that morphologically aware tokenization can significantly improve model performance, generalization, and interpretability. Hofmann et al. (2021) demonstrated that transformer models with derivationally informed vocabularies perform better at interpreting complex word forms, even in English, a language with relatively mild morphological variation. Similarly, Jabbar (2024) introduced MorphPiece, a tokenizer that segments text based on morphemes before applying subword encoding. A GPT-style model trained with this tokenizer achieved superior performance across multiple NLP benchmarks—including language modeling, zero-shot GLUE, and text embedding tasks—despite using only half the training iterations of its BPE-based counterpart. These findings provide strong evidence that token purity, grounded in morphological structure, enhances learning efficiency and leads to more transparent and generalizable language models.

The importance of token purity is analogous to segmentation practices in other machine learning domains. In computer vision, models such as capsule networks (Sabour et al., 2017) and object-centric architectures like Slot Attention (Locatello et al., 2020) show that performance and generalization improve when visual scenes are decomposed into discrete, meaningful entities rather than treated as undifferentiated pixel grids. Capsule networks, for example, represent objects as holistic capsules rather than scattered features, enabling more accurate recognition in complex visual settings. Similarly, Slot Attention learns to bind visual input to abstract object representations, facilitating compositional reasoning and generalization across novel configurations. The same principle applies to language modeling: when token boundaries reflect linguistic structure, the model receives clearer and more interpretable signals. Token purity is thus not merely a linguistic preference—it is a structural requirement for training high-performing, semantically aware language models. This perspective motivates our use of Pure % as a central evaluation metric in this study.

In response to these limitations, this paper introduces a linguistically informed, language-independent tokenization framework that integrates rule-based morphological segmentation with statistical subword modeling. The approach includes several key innovations:

First, phonological normalization is applied so that surface variants of the same morpheme are assigned a unified identifier. This includes mapping affixes with phonological variation triggered by the vowel harmony (e.g., *-dAn*, *-tAn (from)*) and roots with final devoicing (e.g., *kitap* and *kitab (book)*) to shared token IDs. Second, a special token (`<uppercase>`) is used to encode orthographic case distinctions, enabling models to differentiate capitalized tokens without duplicating them in the vocabulary. Third, formatting characters such as space, newline, and tab are explicitly tokenized, preserving the structural integrity of the original text for downstream tasks involving structured documents or layout-sensitive processing. Fourth, a hybrid tokenization algorithm is developed, combin-

ing dictionary-based morphological analysis with Byte Pair Encoding. While morphological segmentation ensures alignment with linguistic units, BPE provides fallback coverage for unknown words, maintaining efficiency and scalability in large corpora.

The proposed tokenizer is evaluated on the TR-MMLU benchmark to test the hypothesis that incorporating linguistic structures—particularly morphological segmentation and phonological normalization—into tokenization can significantly enhance semantic alignment and efficiency in morphologically rich languages. This hypothesis is grounded in prior empirical evidence that linguistic alignment metrics such as Turkish Token Percentage (TR %) and Pure Token Percentage (Pure %) are correlated with downstream performance on MMLU-style benchmarks (Bayram et al., 2025b). Motivated by these findings, this study aims to develop a tokenization strategy that aligns closely with Turkish morphosyntactic structures, minimizes redundancy, and improves interpretability. Empirical results validate this objective: the tokenizer achieved significantly higher TR % and Pure % scores compared to widely used tokenizers such as those from LLaMA, Gemma, and Qwen. These results demonstrate that tokenizers designed with linguistic integrity in mind can yield tokens that are both semantically meaningful and syntactically coherent, without relying on large vocabularies or excessive computational overhead. While the implementation is tailored to Turkish, the underlying methodology is designed to generalize across other languages.

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 Byte Pair Encoding (BPE) (Sennrich et al., 2016), WordPiece (Schuster and Nakajima, 2012), and Unigram (Kudo, 2018). The choice of tokenization directly influences the ability of models to capture syntactic, semantic, and morphological structures, especially in morphologically rich languages like Turkish, Finnish, and Hungarian (Baykara and Güngör, 2022; Toraman et al., 2023).

By the mid-2000s, the development of open-source tools for Turkish and other agglutinative languages began to gain importance. Zemberek (Akın and Akın, 2007) has earned a significant place in the literature as one of the first comprehensive systems developed for the morphological analysis of Turkish. However, since the methods of this period were built on rule-based structures, they remained limited in terms of the scalable representation power and statistical generalization capacity required by modern LLM architectures. Since these tools were not designed suitably for integrated use with models trained on large datasets in particular, statistical tokenization methods have become dominant in practice.

Zemberek-NLP, developed by Akin (2007) (Akın and Akın, 2007), is an open-source NLP framework designed specifically for Turkish and other Turkic languages. This tool offers comprehensive morphological analysis capabilities to handle the complex morphology of agglutinative languages and has become a fundamental tool used in Turkish NLP research for many years.

Recent research has explored alternative tokenization strategies tailored to morphologically rich languages. The morphological tokenizer introduced by Toraman et al. (2023) outperformed conventional subword tokenization techniques, recovering 97% of the performance of larger BERT-based models while reducing model size by a factor of three. Additionally, tokenization granularity has been extensively examined by Kaya and Tantuğ (2024), which found that Turkish requires nearly 2.5 times more subwords per word than English, emphasizing the importance of vocabulary size in achieving optimal model performance.

Tokenization strategies also play a crucial role in machine translation and text generation tasks. Pan et al. (2020) demonstrated that morphology-aware segmentation reduces data sparsity in Neural Machine Translation (NMT) for Turkish-English and Uyghur-Chinese translation models. Additionally, Huck et al. (2017) investigated target-side word segmentation strategies, showing that morphological segmentation improves translation accuracy by maintaining linguistic consistency between source and target languages.

Beyond language modeling and translation, morphological tokenization has been evaluated in abstractive summarization and sentiment analysis tasks. Studies like Baykara and Güngör (2022) revealed that morphology-aware tokenization improves summarization quality by preserving semantic information and reducing information loss. Hybrid tokenization approaches that combine statistical and morphological segmentation have also demonstrated superior performance in multiple NLP tasks, particularly for Named Entity Recognition (NER) and Sentiment Analysis (Kayalı and Omurca, 2024).

Despite these advancements, the computational cost of morphological tokenization remains an open challenge. Expanding the vocabulary size in tokenization increases memory consumption and slows down training times. Liu et al. (2019) and Devlin et al. (2019) highlighted that while larger vocabulary sizes enhance performance in morphologically complex languages, they also contribute to increased model size. Furthermore, energy consumption in large-scale NLP models has become a

growing concern. As discussed by Henderson et al. (2022), optimizing tokenization strategies plays a crucial role in improving resource efficiency and minimizing computational costs.

To address these challenges, recent research has investigated adaptive tokenization methods that dynamically adjust segmentation strategies based on linguistic context. The EuroLLM project (Martins et al., 2024) developed multilingual tokenizers optimized for European languages, incorporating language-specific subword segmentation techniques. Similarly, Lin et al. (2024) proposed a selective tokenization approach that prioritizes semantically meaningful tokens, demonstrating performance improvements in multilingual NLP tasks.

Overall, ongoing research in tokenization strategies continues to evolve, with increasing emphasis on developing efficient, linguistically informed, and adaptive tokenization frameworks. The next section will delve deeper into the role of tokenization in language modeling, pretraining, and benchmark evaluations.

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 (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and GPT (Radford et al., 2019) rely on effective tokenization to balance vocabulary size, sequence length, and computational cost. Studies have shown that inappropriate tokenization choices can introduce biases, degrade semantic coherence, and limit generalization to low-resource languages (Ismayilzada et al., 2025).

A key challenge in tokenization for LLMs is granularity control—striking a balance between excessively fragmented sequences and overly coarse segmentation. A comparative study by Kaya and Tantuğ (2024) analyzed tokenization granularity across English and Turkish, revealing that standard subword tokenization strategies result in Turkish words being split into approximately 2.5 times more subwords than English. This discrepancy affects the efficiency of multilingual models, as Turkish texts require longer sequences to encode the same information.

Benchmark evaluations such as Massive Multitask Language Understanding (MMLU) (Hendrycks et al., 2021) and TR-MMLU (Bayram et al., 2025a) highlight the shortcomings of existing tokenization techniques for morphologically complex languages. The TR-MMLU benchmark, specifically designed to evaluate Turkish NLP models, demonstrated that token purity—the alignment of tokens with linguistic units—correlates strongly with downstream model performance. The findings suggest that tokenization strategies optimized for English may not be directly transferable to Turkish and similar languages, necessitating morphology-aware adaptations.

To address these issues, Bayram et al. (2025b) proposed a novel linguistic integrity framework for evaluating tokenization strategies. This framework introduced token purity and language-specific token percentages (%TR) as critical evaluation metrics, providing a structured approach for assessing how well tokenization preserves morphological structures. Experimental results confirmed that higher %TR values correlate with improved performance on MMLU-style benchmarks, underscoring the importance of preserving language-specific morphemes.

Recent efforts to refine tokenization strategies have included hybrid and domain-adaptive approaches. The ITUTurkBERT system (Kayalı and Omurca, 2024) explored a hybrid tokenization method, combining whitespace segmentation with BPE and Unigram-based subword representations. This method was particularly beneficial for Named Entity Recognition (NER) and abstractive summarization, where preserving linguistic structure is crucial. Similarly, Shakrapani (2024) examined the differences between GPT-4 and GPT-4o, demonstrating that model performance fluctuates depending on tokenization quality, especially in non-English tasks.

Beyond model pretraining, tokenization impacts computational efficiency and energy consumption. Henderson et al. (2022) argued that BPE is suboptimal for pretraining due to inefficient vocabulary utilization, a concern echoed in Henderson et al. (2022). These studies emphasize the need for tokenization techniques that minimize redundancy and optimize training efficiency. Similarly, research on EuroLLM (Martins et al., 2024) has focused on developing multilingual tokenizers that adjust dynamically to different languages, reducing processing overhead while improving semantic coherence.

Despite these advancements, morphological compositionality remains a challenge for LLMs. Ismayilzada et al. (2025) found that state-of-the-art models struggle with morphological productivity, particularly when encountering novel word roots. Their study demonstrated that model performance sharply declines as word complexity increases, a phenomenon that affects agglutinative languages more than English or Chinese. This finding aligns with earlier work by Toraman et al. (2023), which concluded that morphology-aware tokenization improves semantic alignment, model interpretability, and generalization.

The impact of morphological tokenization on NLP pipelines extends beyond text generation and classification. Research in optical character recognition (OCR) and document parsing (Rashad, 2024) has demonstrated that custom tokenization tailored to linguistic structures significantly enhances accuracy. The Arabic-Nougat project, for instance,

introduced a custom tokenizer, Aranizer-PBE-86k, which improved Markdown structure accuracy and character recognition in Arabic OCR tasks.

Further investigations into tokenization adaptation for multilingual models highlight ongoing challenges in cross-linguistic NLP. While standardized tokenization methods enable broad compatibility, they often fail to capture the linguistic diversity of non-English languages. de la Rosa and Arild (2024) established a benchmark for Scandinavian tokenizers, identifying key differences in how tokenization strategies affect language understanding. These findings support the argument that morphology-aware tokenization is essential for low-resource and typologically diverse languages.

Given these insights, tokenization research continues to evolve toward more adaptive, efficient, and linguistically informed models. The next section will explore cutting-edge developments in tokenizer design, including self-learning tokenization, tokenization-free architectures, and the integration of morphological analysis into transformer-based models.

Despite these advancements, morphological segmentation remains underutilized in contemporary LLM architectures. As shown by Ismayilzada et al. (2025), even state-of-the-art LLMs struggle with compositional morphology, particularly when encountering novel root words. Their analysis found that performance declines sharply as morphological complexity increases, with models failing to generalize across different inflected forms. This limitation highlights the need for morphologically informed tokenization that can dynamically adapt to linguistic variations.

The integration of linguistic knowledge into tokenizer design has been further explored through morphological tagging and feature-based tokenization. While standard subword tokenization methods tokenize text without explicit linguistic knowledge, recent studies have experimented with incorporating morphological features directly into tokenization schemes (Bayram et al., 2025b). One such approach involves using morphologically tagged tokens instead of raw subwords, preserving grammatical information that is often lost in statistical segmentation. However, experiments with morphological tagging as tokens have yielded mixed results, as excessive granularity can lead to sequence length expansion, reducing model efficiency (Kaya and Tantuğ, 2024).

An emerging area of interest is dynamic tokenization strategies that adapt based on task requirements. Studies such as Neubeck and van Antwerpen (2024) have introduced more flexible Byte-Pair Tokenizers, capable of dynamically adjusting segmentation rules based on contextual requirements. This marks a shift away from static, pre-defined vocabularies toward more adaptable tokenization approaches that can optimize model performance dynamically.

Despite these advancements, morphological tokenization has yet to become a standard component in mainstream NLP models. While experimental results consistently show that morphology-aware tokenization improves efficiency and accuracy, most large-scale language models still rely on traditional subword segmentation methods. Addressing this gap requires further research into efficient morphological parsing algorithms, lightweight tokenizer architectures, and seamless integration into pretraining pipelines.

In conclusion, tokenization research has evolved significantly from simple whitespace-based segmentation to more sophisticated subword and morphology-aware techniques. However, the limitations of static tokenization—particularly for morphologically rich languages—have spurred interest in self-learning tokenization, hybrid approaches, and tokenization-free architectures. Future research should focus on refining dynamic, language-aware tokenization methods that can enhance NLP models across diverse linguistic contexts, ensuring that tokenization strategies do not become a bottleneck for language model performance.

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 out-of-vocabulary (OOV) words, requiring extensive vocabulary lists that resulted in inefficient memory usage (Radford et al., 2019). 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 (Gage, 1994) and later adapted for NLP by Sennrich et al. (2016), iteratively merges frequent adjacent character pairs into subword units. Similarly, WordPiece, which was initially developed for speech recognition (Schuster and Nakajima, 2012), 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 (Martins et al., 2024). 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 Byte Pair Encoding (BPE) and WordPiece often fail to capture such rich internal structures, fragmenting words in ways that obscure grammatical function and semantic interpretation (Kaya and Tantuğ, 2024). 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.

3.1 System Architecture

At the core of the system lies a hierarchical dataset used to separate words into their morphological components. The first file, `kokler.json`, hosts noun and verb roots in our language, as well as special control tokens critical for the model's operation such as `<uppercase>`, `<unknown>`, `<pad>`, and `<eos>`. The second file, `ekler.json`, hosts Turkish derivational and inflectional suffixes; the most distinguishing feature of this file is that suffixes with the same grammatical function but changing according to sound harmony (e.g., plural suffixes *-lar* and *-ler*) are grouped under a single ID. The third file, `bpe_tokenler.json`, holds subword units created with the *Byte-Pair Encoding* algorithm for rare word parts or foreign-origin structures that do not directly match in the root or suffix dictionaries.

3.2 Encoding (Tokenization) Process

The *Encoding* phase, where text is converted into numerical data the model can understand, is based on the "Longest Prefix Match" algorithm. The process begins by splitting the raw text according to whitespace characters. At this stage, the positions of all uppercase letters are recorded so that uppercase information in the text is not lost, and compound spellings (CamelCase etc.) are separated into segments according to their internal structures.

For each separated word part, the system scans with a specific priority order: First **roots**, then **suffixes**, and finally **BPE tokens** are checked. The longest matching part found is added to the token list; if no match is provided in any database, the `<unknown>` tag is assigned to the part. Additionally, the `<uppercase>` marker is added before segments starting with an uppercase letter in the original text to preserve case information. In the final step, all obtained token objects are converted to their corresponding numerical IDs in the JSON files.

3.3 Decoding (Text Construction and Sound Events)

The section of the system with the most complexity and linguistic intelligence is the *Decoding* phase. This process works on the "Single ID, Multiple Views" principle; that is, the model produces functional IDs, not the surface forms of words. The

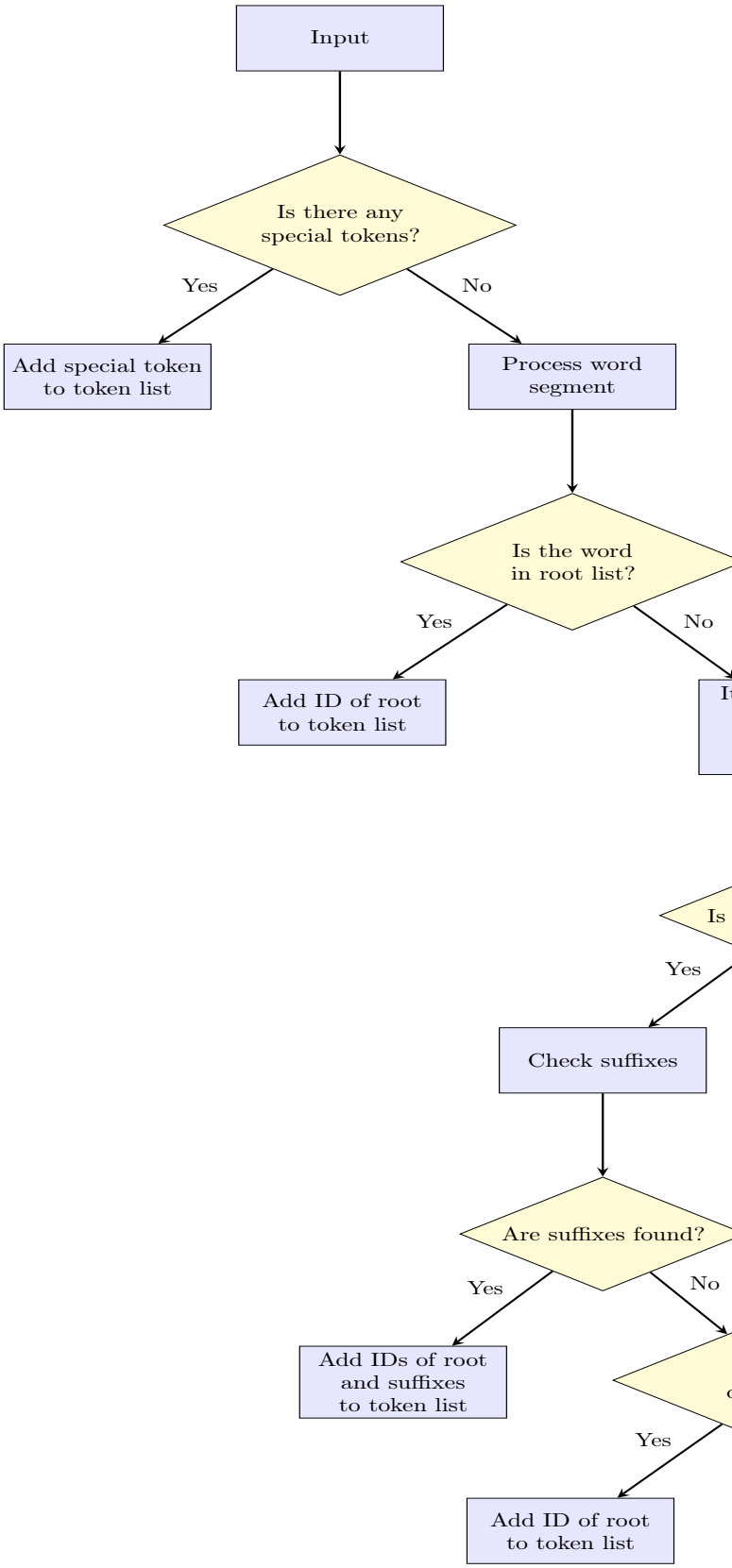

Figure 1: Tokenization decision flow with root, suffix, and fallback segmentation logic.

Decoder, while converting these IDs to text, dynamically resolves Turkish sound events by looking both forward (lookahead) and backward (lookbehind).

In the analysis of roots, the system looks forward, i.e., to the token coming after it. The list of roots is ordered from the form of the word most susceptible to change to the least. For example, the root "sıcak" (hot) hosts `["sıcak", "sıcağ", "sıca"]` variations under a single ID (100). If a suffix starting with a vowel follows this ID (e.g., "-ı"), the system selects the softened form "sıcağ" to form the word "sıcağı". Similarly, if the suffix "-yor" follows the verb "ağla" (cry), the vowel narrowing rule is applied, the "ağlı" form is selected, and the output "ağlıyor" is produced.

In the analysis of suffixes, the system decides by looking backward, i.e., to the token preceding it. Suffixes are shaped according to vowel harmony and consonant hardening rules:

1. **Vowel Harmony:** In suffixes with two variations like the plural suffix, if the last vowel of the previous word is front, "-ler" (meyveler) is selected; if back, "-lar" is selected. In suffixes with four variations (possessive, accusative, derivational suffixes, etc.), the appropriate one from the sounds "ı, i, u, ü" is determined based on whether the previous vowel is unrounded/rounded and front/back. For example, the privative suffix coming after the word "akıl" enters the "ı" harmony and takes the form "akılsız".

2. **Consonant Hardening:** Locative (-da/-de), ablative, or copula suffixes harden if the previous word ends with one of the "FıSTıKÇı ŞaHaP" consonants. For example, the ablative suffix coming after the word "ekmek" becomes "-ten" (ekmekten) instead of "-den"; the copula suffix coming after the word "saat" takes the form "saattir".

3. **Complex Situations:** Structures like the future tense (-acak/-ecek), the denominal noun-making "-lık" or the diminutive suffix "-cık" look at both the previous token (for vowel/consonant harmony) and the next token (for whether softening will occur) simultaneously. For example, while the "-lik" suffix coming to the word "kalem" normally becomes "kalemlik", when a vowel suffix comes after it, it softens and transforms into the "kalemliği" form; the "-cik" suffix coming to the word "kedi" remains as "kedicik".

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

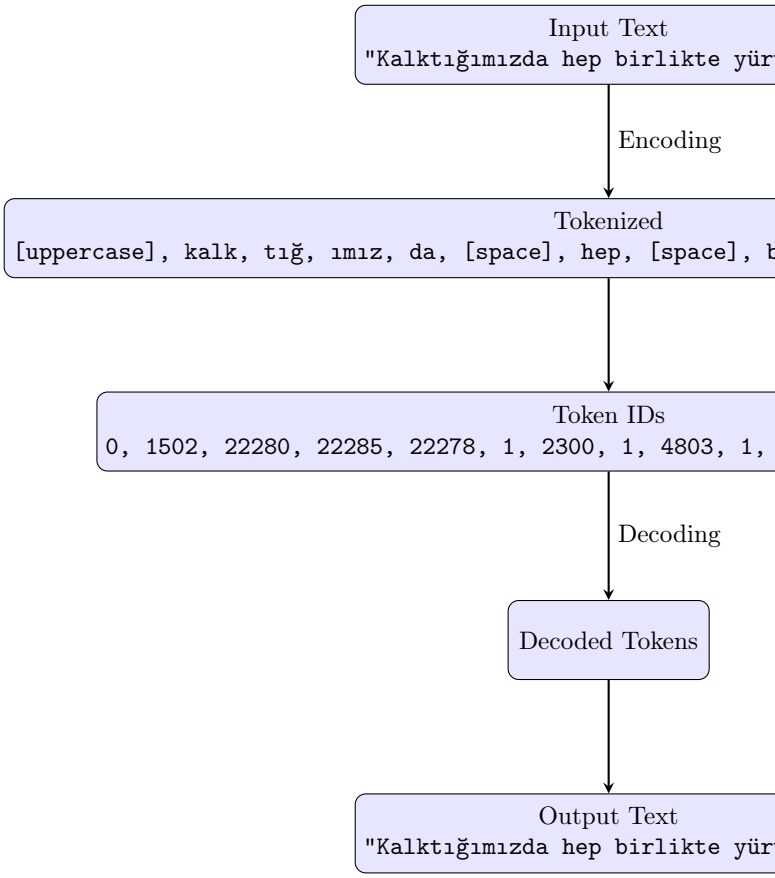

Figure 2: Encoding and decoding process for the sentence “Kalktığımızda hep birlikte yürüdük.”

characters and approximately 200,000 words curated specifically for Turkish (Bayram et al., 2025a). 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 Bayram et al. (2025b). 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 comparison of the proposed `turkish_tokenizer` against baseline models on the TR-MMLU dataset.

Tokenizer	Vocab Size	Total Tokens	Unique Tokens	TR %	Pure %
`turkish_tokenizer` (Ours)	32,768	707,727	11,144	**90.29%**	**85.80%**
`google/gemma-2-9b`	256,000	-	-	40.96%	28.49%
`meta-llama/Llama-3.2-3B`	128,256	-	-	45.77%	31.45%
`Qwen/Qwen2.5-7B-Instruct`	151,936	-	-	40.39%	-
`CohereForAI/aya-expanse-8b`	256,000	434,526	-	53.48%	-

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 Bayram et al. (2025b), 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.

Qwen2.5:
The tokenizer Qwen/Qwen2.5 outputs:
["At", "as", "öz", "leri", " geçmiş", "ten", " gün", "üm", "ü", "ze", " kadar", " ulaş", "an", " anlamı", " bakım", "ından", " me", "ca", "z", "lı", " bir", " mana", " kaz", "anan", " kal", "ı", "pla", "ş", "mış", " söz", "ler", "dir", "."]
While suffixes such as "ten" and "ından" are recognized, the tokenizer introduces redundant splits like "üm", "ü", "ze", reducing the linguistic coherence of the token stream.

Aya-Expanse:
The tokenizer CohereForAI/aya-expanse returns:
["<BOS_TOKEN>", "At", "as", "öz", "leri", " geçmiş", "ten", " günümüze", " kadar", " ulaşan", " anlamı", " bakımından", " mec", "az", "lı", " bir", " mana", " kazanan", " kalı", "pl", "aş", "mış", " söz", "lerdir", "."]
It retains some complete word forms such as "günümüze" and "ulaşan", but still fragments compounds like "kalıplaşmış" and splits the root "atasöz", reducing morphological traceability.

Phi-4:
The tokenizer microsoft/phi-4 produces:
["At", "as", "ö", "z", "leri", " geç", "mi", "ş", "ten", " gün", "üm", "ü", "ze", " kadar", " ", "ula", "ş", "an", " an", "lam", "ı", " bak", "ım", "ından", " me", "ca", "z", "lı", " bir", " mana", " kaz", "anan", " kal", "ı", "pla", "ş", "m", "ış", " sö", "z", "ler", "dir", "."]
This tokenizer over-fragments even basic stems like "geçmiş" into "geç", "mi", "ş" and "anlam" into "an", "lam", increasing token count and reducing interpretability.

YTU Turkish GPT-2:
The tokenizer ytu-ce-cosmos/turkish-gpt2-large-750m-instru 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.

5 Conclusion

In this study, we introduced a linguistically-informed hybrid tokenization framework specifically designed to address the challenges posed by morphologically rich and low-resource languages, with Turkish serving as the primary case study. By integrating rule-based morphological analysis with subword segmentation techniques such as Byte Pair Encoding (BPE), our approach seeks to preserve morpheme boundaries, minimize vocabulary redundancy, and improve syntactic and semantic coherence during tokenization.

Empirical evaluations on the TR-MMLU dataset demonstrated that the proposed `turkish_tokenizer` significantly outperforms existing state-of-the-art tokenizers—including `gemma-2`, `llama-3`, `qwen2.5`, and `aya-expanse`—in both Turkish Token Percentage (TR %) and Pure Token Percentage (Pure %), achieving 90.29% and 85.80%, respectively. These metrics reflect the tokenizer's strong alignment with the linguistic structure of Turkish, a crucial factor for downstream NLP tasks. The tokenizer also exhibited efficient vocabulary utilization with only 32,768 entries and showed robust performance in handling morphosyntactic structures across diverse sentence types.

Qualitative analyses further reinforced the superiority of our approach, revealing that the proposed tokenizer segments text into linguistically meaningful units and accurately preserves suffixes, compound forms, and phonologically altered variants—challenges frequently mishandled by general-purpose, frequency-driven tokenization strategies. The findings presented here reaffirm the thesis proposed in Bayram et al. (2025b), namely that tokenization strategies rooted in linguistic structure are not only desirable but necessary for accurate and efficient language modeling in morphologically complex settings. As NLP continues to evolve toward inclusive, multilingual systems, the development of linguistically aware tokenization methods will be critical for ensuring equity in language technologies.

Future directions include extending this hybrid framework to other agglutinative and typologically diverse languages, refining the morphological rules through semi-supervised learning, and exploring integration with multilingual LLM pretraining pipelines to optimize performance in low-resource language environments.

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].

Javier de la Rosa and Rolv Arild. 2024. Nbailab/tokenizer-benchmark.

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

Philip Gage. 1994. A new algorithm for data compression.

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].

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

Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2021. 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.

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.

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

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].

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

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. 2024. 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. Curran Associates Inc.

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].

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. 2020. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems*, volume 33, pages 11525–11538.

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].

Alexander Neubeck and Hendrik van Antwerpen. 2024. So many tokens, so little time: Introducing a faster, more flexible byte-pair tokenizer.

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].

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

Mohamed Rashad. 2024. Arabic-nougat: Fine-tuning vision transformers for arabic ocr and markdown extraction. *arXiv preprint*. ArXiv:2411.17835 [cs].

Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic routing between capsules. *arXiv preprint*. ArXiv:1710.09829 [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].

Kalai Shakrapani. 2024. Gpt 4 vs gpt 4o (optimized): A comparison of large language models (llm).

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].