Tokenizasyonun Yeniden Ele Alınması: Çok Dilli NLP'de Verimlilik ve Dilbilimsel Bütünlüğün Dengesi

M. Ali Bayram[1], **Ali Arda Fincan**[2], **Ahmet Semih Gümüş**[2], **Sercan Karakaş**[3], **Banu Diri**[1]
[1]Yıldız Teknik Üniversitesi, [2]Yeditepe Üniversitesi, [3]OpenAI
malibayram20@gmail.com

Abstract

Tokenization is a critical preprocessing step in natural language processing (NLP), shaping how effectively large language models (LLMs) capture linguistic and semantic nuance. This paper presents a comprehensive framework for tokenization standards that prioritize morphological and semantic integrity, especially for morphologically complex and low-resource languages. Using Turkish as a case study, we evaluate eight LLM-derived tokenizers on a subset of the Massive Multitask Language Understanding (MMLU) benchmark. Our analysis goes beyond conventional efficiency metrics—such as vocabulary size, token count, and processing speed—by incorporating linguistic token percentages and semantic purity to assess how faithfully tokenizers preserve linguistic structure. The results highlight that language-specific tokenization strategies substantially improve downstream performance, even when training data is limited, and show that larger model parameters do not inherently yield better tokenization or enhanced results. These findings underscore the need to balance computational efficiency with linguistic alignment, tailoring tokenization methods to the morphological properties of each language. The proposed framework is adaptable to various linguistic contexts, guiding the development of tokenizers that improve model accuracy, robustness, and versatility. Future work will explore advanced morphological analysis, domain-specific customization, and cross-linguistic comparisons to further refine tokenization practices and advance the state of multilingual NLP.

Keywords: Tokenization Standards, Morphological Integrity, Semantic Fidelity, Low-Resource Languages, Multilingual NLP, Morphologically Complex Languages

Özet

Tokenizasyon, doğal dil işleme (NLP) alanında, büyük dil modellerinin (LLM'ler) dilbilimsel ve anlamsal detayları yakalama başarısını doğrudan şekillendiren kritik bir ön işleme adımıdır. Bu çalışma, özellikle morfolojik olarak zengin ve düşük kaynaklı dillerde, morfolojik ve anlamsal bütünlüğü önceliklendiren tokenizasyon standartları için kapsamlı bir çerçeve sunmaktadır. Örnek çalışma olarak Türkçeyi ele alarak, Massive Multitask Language Understanding (MMLU) benchmark'ının bir alt kümesinde sekiz farklı LLM tabanlı tokenizasyon yöntemi değerlendirilmiştir. Analizimiz, geleneksel verimlilik metriklerinin ötesine geçerek (örn. sözlük boyutu, token sayısı, işleme süresi), dilbilimsel token yüzdesi ve anlamsal saflık gibi metrikler üzerinden, tokenizasyonun dil yapısını ne derece doğru temsil ettiğini incelemiştir. Sonuçlar, dil spesifik tokenizasyon stratejilerinin, sınırlı eğitim verisi olduğunda dahi aşağı akış performansını önemli ölçüde iyileştirdiğini ve daha büyük model parametrelerinin her zaman daha iyi tokenizasyon veya sonuçlar sağlamadığını göstermektedir. Bu bulgular, hesaplama verimliliği ile dilbilimsel uyumu dengeleyerek tokenizasyon yöntemlerinin her dilin morfolojik özelliklerine göre uyarlanması gerektiğini vurgulamaktadır. Önerilen çerçeve, farklı dil bağlamlarına uyarlanabilir olup, model doğruluğunu, dayanıklılığını ve çok yönlülüğünü artıran tokenizasyon geliştirmeleri için bir rehber niteliğindedir.
Anahtar Kelimeler: Tokenizasyon Standartları, Morfolojik Bütünlük, Anlamsal Saflık, Düşük Kaynaklı Diller, Çok Dilli NLP, Morfolojik Olarak Zengin Diller

1 Giriş

Tokenizasyon, doğal dil işleme (NLP) alanında, dil modellerinin etkili ve verimli çalışmasını doğrudan etkileyen kritik bir ön işleme adımıdır. Bu süreç, metni kelime, alt kelime veya karakter gibi daha küçük birimlere ayırmayı içerir ve bu birimler modeller için temel giriş verilerini oluşturur. Tokenizasyon, tüm dillerde evrensel bir gereklilik olmasına rağmen, morfolojik olarak zengin ve eklemeli dillerde, örneğin Türkçede, bu süreç daha karmaşık hale gelir. Türkçe gibi dillerde, tek bir kelime genellikle bir kök ve birden fazla ek içerir; her biri ayrı bir dilbilgisel veya anlamsal anlam taşır. Bu gibi durumlarda, standart tokenizasyon yöntemleri ince dilbilimsel özellikleri yakalamakta başarısız olabilir ve aşağı akış görevlerindeki performansı düşürebilir.

Son yıllarda, Byte Pair Encoding (BPE) ve SentencePiece gibi alt kelime tokenizasyon tekniklerindeki gelişmeler, dillerin karmaşık yapılarının daha iyi temsil edilmesinde önemli iyileştirmeler sağlamıştır. Bu yöntemler, kelimeleri daha küçük alt kelime birimlerine bölerek, modellerin nadir ve görülmemiş kelimelerle daha etkili bir şekilde başa çıkmasına olanak tanır. Örneğin, Arapça için geliştirilen `Aranizer-PBE-86k` tokenizatörü, dilin morfolojik inceliklerini etkili bir şekilde yakalayarak Türkçe gibi diğer dillerdeki benzer zorlukların üstesinden gelme konusunda önemli ipuçları sunmaktadır [1]. Bu yöntemler, İngilizce gibi daha basit morfolojilere sahip dillerde dahi, özellikle veri kıtlığının olduğu senaryolarda, desen tanıma ve temsil verimliliğini artırmaktadır.

Tokenizasyon kalitesini değerlendirmek için iki kritik metrik vardır: *token saflığı* ve bir dil için *token yüzdesi*. Token saflığı, üretilen tokenların anlamlı dilbilimsel birimlere (ör. kökler, geçerli ekler veya anlamsal olarak tutarlı segmentler) ne derece uyduğunu ölçer. Yüksek bir token saflığı, kelimelerin anlamlı parçalarının tokenizasyon sırasında korunmasını sağlar, parçalanmayı en aza indirir ve modellerin dil kalıplarını daha etkili bir şekilde öğrenmesine olanak tanır. Belirli bir dilin token yüzdesi, örneğin Türkçe token yüzdesi (%TR), bir dilin yapısına uygun olarak geçerli kelimeler veya dilbilimsel birimler olan tokenların oranını belirtir. Bu metrik, tokenizasyonun hedef dilin yapısına uygunluğunu sağlar ve geçersiz veya dilbilimsel olmayan tokenlardan kaynaklanan gürültüyü azaltır.

Bu metrikler, yalnızca morfolojik olarak zengin diller için değil, aynı zamanda İngilizce gibi tüm diller için evrensel olarak önemlidir. Yüksek token saflığı ve dil spesifik token yüzdesi, dil modellerinin sınırlı eğitim verisiyle dahi anlamlı kalıpları daha etkili bir şekilde öğrenmesine olanak tanır. Dilbilimsel birimlerin anlamsal bütünlüğünün korunması, modellerin büyük ölçekli veri setlerine ihtiyaç duymadan genelleme yapmasına ve aşağı akış görevlerinde daha iyi performans göstermesine yardımcı olur. Bu durum, özellikle düşük kaynaklı diller veya eğitim verisinin kısıtlı olduğu uzmanlaşmış alanlar için büyük önem taşımaktadır.

Son yıllardaki gelişmelere rağmen, tokenizasyon hızını, sözlük boyutunu ve dilbilimsel bütünlüğü dengeleme konusu halen bir zorluktur. Aşırı parçalanma, anlamsal anlamı seyreltebilirken, aşırı kaba tokenizasyon kritik dilbilimsel detayları gözden kaçırabilir. Bu denge, yalnızca Türkçe gibi morfolojik olarak zengin diller için değil, aynı zamanda daha basit dillerdeki performans ve verimlilik için de hayati öneme sahiptir [2].

Bu makale, Türkçe için MMLU benchmark'ını kullanarak tokenizasyon yöntemlerini değerlendirmektedir. Token saflığı, token yüzdesi, sözlük boyutu ve işleme hızı gibi metriklere dayalı analiz yaparak, Türkçe NLP görevleri için en etkili yaklaşımları belirlemeyi hedeflemektedir. Bu çalışmadan elde edilen bulgular, tüm dillerde optimize edilmiş tokenizasyon stratejilerinin geliştirilmesine katkıda bulunmakta ve NLP modellerinin doğruluğunu ve verimliliğini artırmaktadır.

2 İlgili Çalışmalar

Tokenizasyon, doğal dil işleme (NLP) alanında dil modellerinin performansı, verimliliği ve doğruluğunu doğrudan etkileyen temel bir role sahiptir. Son çalışmalar, dilbilimsel bütünlüğü, hesaplama verimliliğini ve model ölçeklenebilirliğini dengelemek amacıyla çeşitli tokenizasyon stratejilerini ve bunların aşağı akış etkilerini araştırmıştır.

Arabic Tokenizers Leaderboard [3], Arapça'nın çeşitli lehçeleri ve yazım karmaşıklığının neden olduğu zorlukları vurgulayarak `rasaif-translations` ve `Moroccan Arabic Wikipedia` gibi datasetler kullanarak Arapça tokenizatörlerini karşılaştırmaktadır. Bu çalışmada kullanılan araçlardan biri olan *AraNizer* [1], Byte Pair Encoding (BPE) ve SentencePiece gibi alt kelime tabanlı tekniklerden yararlanarak, Arapça'nın morfolojik inceliklerini daha iyi yakalamayı ve aşağı akış görevlerindeki performansı artırmayı amaçlamaktadır.

Benzer şekilde, *NbAiLab Tokenizer Benchmark* [4], İskandinav dilleri için tokenizasyon stratejilerini değerlendirerek, çok dilli bağlamlarda dil spesifik adaptasyonların önemini vurgulamaktadır. Almanca için Diewald ve arkadaşları [5], `KorAP-Tokenizer` ve `SoMaJo` gibi araçları değerlendirerek, büyük ölçekli metinler için yüksek doğruluk ve hesaplama verimliliği sağlamayı başarmıştır.

EuroLLM ekibi, çok dilli yapıların desteklenmesi için geniş sözlüklere sahip tokenizatörlerin tasarımının önemini vurgulayan önemli bir katkı sunmaktadır [6]. Byte Pair Encoding (BPE) ile birlikte byte fallback kullanan ve 128,000 parçalı bir sözlük içeren EuroLLM tokenizatörü, düşük "fertility" (kelime başına token sayısı) ile parametre verimliliği arasında bir denge sağlamaktadır. EuroLLM'nin, Mistral, LLaMA-3 ve Gemma tokenizatörleriyle yaptığı fertility karşılaştırmaları, geniş sözlüklerin hesaplama maliyetleri ile dil işleme yetenekleri arasındaki dengeyi anlamaya yönelik önemli bulgular sunmaktadır.

Verimlilik konusundaki ilerlemeler, GitHub'ın hızlı BPE uygulamasında da gözlemlenmiştir [2]. Bu uygulama, milyarlarca token gerektiren görevlerde ölçeklenebilirliği artırarak, büyük ölçekli NLP görevleri için önemli bir performans artışı sağlamaktadır.

Rust ve arkadaşları [7], belirli dillere özel olarak uyarlanmış monolingual tokenizatörlerin, aşağı akış görev performansını önemli ölçüde artırdığını göstermektedir. Benzer şekilde, Lin ve arkadaşları [8], tokenlara fayda puanı atayan ve yüksek faydalı tokenlar üzerinde seçici eğitim uygulayan Selective Language Modeling (SLM) yöntemini önermektedir. Bu yöntem, Türkçe gibi morfolojik olarak zengin dillerde anlamlı tokenların korunmasının kritik olduğu durumlarda özellikle yararlıdır.

Bu çalışmalar, dilbilimsel bütünlük, hesaplama verimliliği ve aşağı akış performansı arasında bir denge sağlayan tokenizasyon stratejilerinin gerekliliğini vurgulamaktadır. Bu çalışmada, Türkçe için tokenizatörler değerlendirilerek token saflığı, sözlük boyutu ve işleme hızı gibi metrikler kullanılmıştır. Çok dilli projelerden, özellikle EuroLLM'den elde edilen içgörüler entegre edilerek, morfolojik olarak zengin diller için tekniklerin optimize edilmesine katkıda bulunulmuştur.

3 Yöntem

Bu çalışma, morfolojik olarak zengin ve eklemeli dillere yönelik tokenizasyon stratejilerini değerlendirmeyi hedeflemekte olup, Türkçe bu bağlamda temsilci bir örnek olarak kullanılmaktadır. Ana odak Türkçe olsa da, yöntem diğer dillerdeki benzer tokenizasyon zorluklarına uyarlanabilir bir esneklikle tasarlanmıştır.

Bu değerlendirmeyi gerçekleştirmek için, çeşitli konuları kapsayan 6.200 çoktan seçmeli sorudan oluşan Türkçe MMLU veri seti kullanılmıştır [9]. Bu veri seti, Hugging Face deposunda saklanan ön işlenmiş bir kaynaktan soruların ve cevapların çıkarılmasıyla hazırlanmıştır [9]. Ortaya çıkan metin korpusu, Türkçe'de karşılaşılan çeşitli dilbilimsel yapıları geniş bir kapsama alanıyla içeren birleşik bir veri setine dönüştürülmüştür. Türkçe verileri kullanılarak geliştirilen bu çerçeve, benzer dil araçları ve kaynaklarının uygulanmasıyla diğer dillere de uyarlanabilir.

Bu çalışma kapsamında, tokenizasyonun hem hesaplama hem de dilbilimsel yönlerini değerlendirmek için çeşitli metrikler kullanılmıştır:

Sözlük Boyutu: Bir tokenizatörün üretebileceği benzersiz tokenların (ör. kelimeler, alt kelimeler, karakterler) toplam sayısını ifade eder. Örneğin, 50.000 sözlük boyutuna sahip bir tokenizatör `"cat"` veya `"run"` gibi tokenları, ayrıca `"run"` ve `"ning"` gibi alt kelime birimlerini içerebilir. Daha büyük sözlükler daha karmaşık dilbilimsel yapıları yakalayabilir, ancak aşırı büyük sözlükler karmaşıklığı ve bellek kullanımını artırabilirken, daha küçük sözlükler nadir kelimeleri yeterince temsil edemeyebilir.

Toplam Token Sayısı: Tokenizatör uygulandıktan sonra veri setinde üretilen toplam token sayısını ifade eder. Örneğin, `"I love programming languages"` cümlesi, boşluk tabanlı bir tokenizatörle `["I", "love", "programming", "languages"]` olarak ayrıştırıldığında dört token oluşur. Alt kelime tabanlı bir tokenizatör, `["I", "love", "program", "ming", "languages"]` şeklinde beş token üretebilir. Daha düşük toplam token sayıları, daha kompakt temsiller anlamına gelebilir ve bu da verimliliği artırabilir.

İşleme Süresi: Veri setinin tamamını tokenize etmek için gereken süreyi (saniye cinsinden) ifade eder ve hesaplama verimliliğini yansıtır. Örneğin, bir milyon kelimelik bir korpus 3.2 saniyede işlenirse, işleme süresi 3.2 saniye olarak kaydedilir. Daha hızlı tokenizasyon, büyük ölçekli eğitimler ve gerçek zamanlı uygulamalar için avantajlıdır.

Token Yüzdesi (%TR): Hedef dilde geçerli kelimelere veya morfemlere karşılık gelen tokenların oranını ölçen bir dilbilimsel metriktir. Örneğin, `"Cats are playing"` cümlesi `["Ca", "ts", "are", "play", "ing"]` olarak tokenize edilirse, `"are"`, `"play"` ve `"ing"` geçerli dilbilimsel birimler olarak kabul edilir. Bu durumda:

$$\%TR = \frac{\text{Geçerli Tokenlar (3)}}{\text{Toplam Tokenlar (5)}} \times 100 = 60\%.$$

Bu metrik, tokenizasyonun dilin morfolojisine uygunluğunu sağlar ve geçersiz segmentleri en aza indirir.

Saf Token Yüzdesi (%Pure): Tokenların doğrudan anlamlı olup olmadığını ve daha küçük anlamlı parçalara ayrılıp ayrılamayacağını değerlendiren bir metriktir. Örneğin, `"The students are learning"` cümlesinde, `"The"` ve `"are"` saf tokenlardır, ancak `"students"` (`"student"` + `"s"`) ve `"learning"` (`"learn"` + `"ing"`) saf değildir, çünkü daha küçük birimlere ayrılabilirler. Eğer bir tokenizatör 100 benzersiz token üretmiş ve bunların 70'i saf token ise:

$$\%Pure = \frac{\text{Saf Tokenlar (70)}}{\text{Benzersiz Tokenlar (100)}} \times 100 = 70\%.$$

Doğru morfolojik analiz ve token doğrulaması sağlamak için ITU Turkish NLP Web Service [10] ve Kalbur kütüphanesi [11] gibi dil araçlarından yararlanılmıştır. Diğer diller için benzer dil analiz araçları ve kurallara dayalı sistemler entegre edilebilir. İşleme süresi ve token sayıları gibi hesaplama metrikleri, Python betikleri ve Hugging Face Tokenizers kütüphanesi [2] kullanılarak hesaplanmış ve bu, ölçeklenebilirliği ve uyarlanabilirliği sağlamıştır.

Tüm deneysel prosedürler, veri setleri ve yapılandırmalar çoğaltılabilirlik için belgelenmiştir. Bu çalışmada Türkçe birincil referans olarak kullanılmış olsa da, önerilen yöntem diğer dillere ve veri setlerine uygulanabilir bir yapı sunarak, farklı dilsel bağlamlarda tokenizasyon stratejilerini değerlendirmeye yönelik genel bir yaklaşım sunmaktadır.

4 Sonuçlar ve Analiz

Bu çalışmada, 1.605.376 karakter ve 198.193 kelimeden oluşan Türkçe MMLU veri seti üzerinde sekiz farklı tokenizatör değerlendirilmiştir [9]. Hem dilbilimsel uyumu hem de hesaplama etkinliğini

ölçmek amacıyla, Türkçe Token Yüzdesi (%TR) ve Saf Token Yüzdesi (%Pure) gibi dilbilimsel metrikler ile işleme süresi, model parametre boyutu ve MMLU skorları incelenmiştir. Analizlere model boyutunun dahil edilmesi, morfolojik olarak zengin diller için tokenizatör tasarımında yapılan ödünleşimleri daha ayrıntılı bir şekilde ortaya koymaktadır.

Table 1: Tokenizatör Karşılaştırma Sonuçları

Metrik	**gemma-2**	**llama-3.1**	**EuroLLM**	**Qwen2.5**	**aya-exp**	**Mistral**	**Phi3.5**	**gpt4o**
Model Parametre (Milyar)	27.2	70.6	9.2	7.6	32.3	12.2	3.8	Bilinmiyor
MMLU Skoru (%)	72.10	70.42	51.29	61.68	70.66	46.89	29.37	84.84
Sözlük Boyutu	256,000	128,256	128,000	151,665	255,029	131,072	32,011	200,019
Token Sayısı	497,015	488,535	497,173	561,866	434,526	534,930	803,971	491,137
İşleme Süresi (s)	2.95	3.12	3.20	3.31	2.77	3.14	4.55	0.51
Benzersiz Tokenlar	6,383	6,823	5,226	5,752	8,562	4,354	3,640	7,615
Türkçe Tokenlar	3,104	3,125	2,457	2,320	4,338	1,971	1,599	3,209
%TR	48.63	45.80	47.01	40.33	50.67	45.27	43.93	42.14
Saf Tokenlar	2,365	2,109	1,838	1,734	2,822	1,571	1,253	2,184
%Pure	37.05	30.91	35.17	30.15	32.96	36.08	34.42	28.68

Tablo 1, sözlük boyutu, token sayıları, çalışma süresi ve dil spesifik uyum gibi metriklerin, modellerin performansıyla nasıl etkileşime girdiğini özetlemektedir. Özellikle, daha büyük parametre boyutlarının üstün sonuçlar garanti etmediği gözlemlenmiştir. Örneğin, `gemma-2` (27.2 milyar parametre), `llama-3.1` (70.6 milyar parametre) gibi oldukça büyük bir modelden daha yüksek Türkçe MMLU doğruluğu sağlamıştır. Bu durum, `gemma-2`'nin tokenizatörünün Türkçe morfolojisine daha iyi uyum sağladığını göstermektedir. Bunun aksine, İngilizce MMLU sıralamalarında `llama-3.1` üstün gelmektedir [12], bu da tokenizatör etkinliğinin dile bağımlılığını ortaya koymaktadır.

`gemma-2`, %TR (48.63%) ve %Pure (37.05%) açısından, `llama-3.1`'den (%TR: 45.80%, %Pure: 30.91%) daha yüksek değerler elde etmiştir. Bu durum, dil spesifik tokenizasyonun morfolojik yapıyı yakalamada ne kadar önemli olduğunu vurgulamaktadır. `aya-expanse` (32.3 milyar parametre) en yüksek %TR (50.67%) değerine ulaşsa da, MMLU skoru (70.66%) dilsel uyumun hesaplama verimliliği ve görev optimizasyonu ile dengelenmesi gerektiğini göstermektedir.

Diğer yandan, `o200k-gpt40`, en yüksek MMLU skoruna (84.84%) ve en hızlı işleme süresine (0.51s) sahip olmasına rağmen, göreceli olarak düşük bir %TR (42.14%) sergilemiştir. Bu durum, damıtma ve budama gibi optimizasyon tekniklerinin [13, 14], dilsel bütünlük pahasına performans ve hızı artırabileceğini göstermektedir. Benzer şekilde, `Mistral` (12.2 milyar) ve `Phi3.5` (3.8 milyar) gibi daha küçük modeller, hem dilsel hem de hesaplama açısından düşük performans göstermiştir. Bu bulgular, sınırlı parametre boyutlarının, karmaşık morfolojileri etkili bir şekilde işlemek için daha gelişmiş tokenizasyon stratejilerine ihtiyaç duyduğunu göstermektedir.

Şekil 1, bu ödünleşimlerin çok boyutlu bir görselleştirmesini sunmaktadır. MMLU skorlarını %TR'ye karşı çizerek, marker büyüklüğüyle parametre sayısını ve renk kodlamasıyla %Pure'u temsil etmektedir. Bu görselleştirme, dilin morfolojik zenginliğini daha iyi yakalayan modeller ile daha büyük parametre sayısına veya belirli optimizasyonlara sahip modellerin genel performansta nasıl bir fark yarattığını holistik bir şekilde karşılaştırma olanağı sağlar.

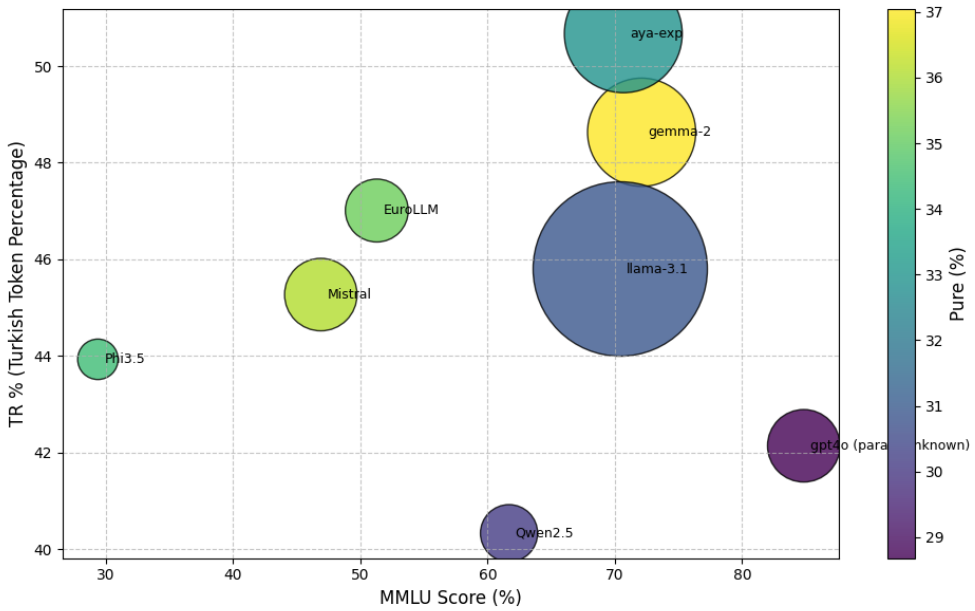

Figure 1: Model Karşılaştırması: MMLU vs TR%, Parametre Boyutu (gpt4o için bilinmiyor) ve Pure%

Sonuç olarak, bu bulgular, dilsel uyumun, morfolojik olarak karmaşık ortamlarda aşağı akış görevlerindeki başarıyı büyük ölçüde şekillendirdiğini doğrulamaktadır. Özel olarak uyarlanmış tokenizasyon, daha küçük modellerin etkili bir şekilde rekabet etmesine yardımcı olabilirken, daha büyük veya optimize edilmiş modeller bile tokenizatörleri dilsel olarak uyumlu olmadığında yetersiz kalabilir.

5 Gelecek Çalışmalar

Bu çalışma, morfolojik olarak zengin ve düşük kaynaklı diller için optimize edilmiş tokenizatörlerin geliştirilmesine rehberlik eden bir çerçeve sunarak, dilsel bütünlük ve hesaplama verimliliğinin önemini vurgulamaktadır. Araştırma kapsamında geliştirilen çeşitli tokenizatörler, Türkçe gibi morfolojik açıdan karmaşık dillerde promising performans sergilemektedir. Ancak, bu tokenizatörler, bu tür dillerin tokenize edilmesiyle ilgili zorlukların yalnızca başlangıç aşamasını ele almaktadır. Tablo 2, `AhmetSemih/tr_tokenizer` ve `aliarda/turkish_tokenizer` gibi tokenizatörlerin, yüksek Türkçe Token Yüzdeleri (%TR) ve Saf Token Yüzdeleri (%Pure) ile dikkat çektiğini göstermektedir.

Table 2: Tokenizatörlerin İlk Geliştirme Aşamasındaki Performans Metrikleri

Tokenizatör	Sözlük Boyutu	Token Sayısı	Zaman (s)	Benzersiz Tokenlar	Türkçe Tokenlar	%TR	Saf Tokenlar	%Pure
alibayram/tr_tokenizer	30,158	476,556	2.42	11,531	11,342	98.36	11,055	95.87
AhmetSemih/tr_tokenizer	59,572	451,883	2.48	13,370	13,253	99.12	13,357	99.90
aliarda/turkish_tokenizer_256k	256,000	488,267	2.51	13,631	13,351	97.95	12,981	95.23
aliarda/turkish_tokenizer	58,526	451,936	2.34	13,268	13,170	99.26	13,256	99.91

Bu umut verici sonuçlara rağmen, bu tokenizatörlerin tam potansiyelini ortaya çıkarmak için daha fazla çalışma gerekmektedir. Gelecekteki iyileştirmeler, Türkçe'nin zengin dilbilgisel ve anlamsal yapısını daha iyi yakalamak için gelişmiş morfolojik analiz adımlarının entegrasyonuna odaklanacaktır. Bu adımlar arasında daha sofistike dilbilimsel kuralların entegrasyonu, nadir görülen morfemlerin ele alınması ve bağlama dayalı varyasyonların hesaba katılması yer alabilir. Bu tür geliştirmeler yalnızca dilsel bütünlüğü artırmakla kalmayacak, aynı zamanda tokenizatörlerin çeşitli NLP uygulamaları için kapsamını da genişletecektir.

Ek olarak, tokenizasyon süreçlerini dinamik hale getiren yinelemeli iyileştirme yöntemleri araştırılacaktır. Bu yöntemler arasında, aşağı akış görevleri ve alan spesifik gereksinimlere göre dinamik token üretimi yer alabilir. Örneğin, tokenizatörler, tıbbi, hukuki veya teknik metinler gibi belirli

alanlar için ince ayar yapılarak, özel uygulamalarda yüksek performans sağlamaları hedeflenebilir. Ayrıca, denetimsiz ve yarı-denetimli öğrenme yaklaşımlarının geliştirme süreçlerine dahil edilmesi, morfolojik ve anlamsal kapsama alanındaki eksikliklerin giderilmesine yardımcı olacaktır.

Henüz gelişimin erken aşamalarında olmasına rağmen, bu tokenizatörler daha fazla yenilik için sağlam bir temel sunmaktadır. İlk performansları, hedeflenen iyileştirmelerle birlikte, morfolojik olarak zengin dillerin tokenize edilmesinde güçlü ve çok yönlü araçlara dönüşebilecekleri umudunu vermektedir. Bu ek adımların uygulanması ve farklı diller ve görevler üzerinde daha fazla değerlendirme yapılmasıyla, bu araştırma, dilbilimsel açıdan bilgilendirilmiş tokenizasyon için yeni bir standart oluşturmayı ve dil modellerinin kalite ve verimliliğini çeşitli uygulamalarda ileri taşımayı amaçlamaktadır.

6 Sonuç

Bu çalışma, tokenizasyon stratejilerini değerlendirmek için kapsamlı bir çerçeve sunarak, dilsel bütünlüğün korunmasının yanı sıra hesaplama verimliliğinin önemini vurgulamaktadır. Token saflığı (*token purity*), Türkçe Token Yüzdesi (TR %) ve işlem verimliliği gibi metriklere odaklanarak, tokenizasyon stratejilerinin özellikle morfolojik olarak zengin dillerde, Türkçe gibi, model performansı üzerinde önemli bir etkisi olduğunu göstermiştir. Analizlerimiz, parametre boyutunun tek başına performansın kesin bir göstergesi olmadığını ortaya koymuştur. Örneğin, `gemma-2` (27.2 milyar parametre), Türkçe MMLU değerlendirmelerinde daha büyük bir model olan `llama-3.1`'i (70.6 milyar parametre) geride bırakarak, tokenizasyonun dil yapısıyla uyumunun kritik rolünü vurgulamıştır. Buna karşılık, genel amaçlı modeller olan `o200k-gpt4` gibi, kapsamlı optimizasyonlar sayesinde aşağı akış görevlerinde mükemmel sonuçlar elde ederken, daha düşük dilsel bütünlük sergilemiş ve görev odaklı model optimizasyonunun doğal getirdiği ödünleşimlere dikkat çekmiştir.

Bu bulgular, dilsel korunumu ve hesaplama gereksinimlerini dengeleyen özelleştirilmiş tokenizasyon stratejilerinin, çeşitli dillerde sağlam NLP performansı elde etmek için gerekli olduğunu vurgulamaktadır. Önerilen çerçeve yalnızca Türkçe'ye değil, diğer dillere ve alanlara da uygulanabilir olup, tokenizasyon yöntemlerini optimize etmek ve çok dilli NLP uygulamalarını iyileştirmek için bir temel sunmaktadır. Gelecek araştırmalar, bu çerçeveyi görev özelinde değerlendirmeleri ve diller arası karşılaştırmaları kapsayacak şekilde genişleterek, çeşitli dil bağlamları için tokenizasyon stratejilerini daha da geliştirmeyi hedefleyecektir.

References

[1] Anis Koubaa, Lahouari Ghouti, Omar Najar, and Serry Sebai. github.com/riotu-lab/aranizer, December 2024. original-date: 2023-12-19T07:57:47Z.

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

[3] Mohamed Rashad. Arabic Tokenizers Leaderboard - a Hugging Face Space by MohamedRashad.

[4] Javier de la Rosa and Rolv Arild. NbAiLab/tokenizer-benchmark, November 2024. original-date: 2024-03-23T09:22:14Z.

[5] Nils Diewald, Marc Kupietz, and Harald Lüngen. Tokenizing on scale.Preprocessing large text corpora on the lexical and sentence level. 2022.

[6] 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].

[7] Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. How Good is Your Tokenizer? On the Monolingual Performance of Multilingual Language Models, June 2021. arXiv:2012.15613 [cs].

[8] 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.

[9] M. Ali Bayram, Ali Arda Fincan, and Ahmet Semih Gümüş. Turkish Mmlu Leaderboard - a Hugging Face Space by alibayram.

[10] Gülşen Eryiğit. ITU Turkish NLP Web Service. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–4, Gothenburg, Sweden, 2014. Association for Computational Linguistics.

[11] Ahmet Aksoy. ahmetax/kalbur, October 2024. original-date: 2016-10-26T10:25:48Z.

[12] DocsBot AI. Llama 3.1 70B Instruct vs Gemma 2 27B - Detailed Performance & Feature Comparison.

[13] Lisa Lacy. GPT-4o and Gemini 1.5 Pro: How the New AI Models Compare, May 2024.

[14] Kalai Shakrapani. GPT 4 vs GPT 4o (optimized): A Comparison of Large Language Models (LLM) | LinkedIn.