

TÜRKÇENİN TOKEN DURUMU¹

The Tokenization Status of Turkish

Dila CAN²

Öz

Bu araştırma, Türkçenin büyük dil modelleri karşısındaki token durumunu incelemektedir. Günümüz büyük dil modellerinin temelini oluşturan transformer mimarisi ve buna dayalı BPE tabanlı tokenleştirme sistemleri, veri setlerinin İngilizce ağırlıklı olması nedeniyle doğrudan İngilizce dil yapısına göre optimize edilmiştir. Bu durum, Türkçe gibi eklemeli ve zengin morfolojiye sahip dillerin aynı içerik için önemli ölçüde daha fazla token harcamasına, dolayısıyla daha yüksek işlem maliyetine katlanmasına yol açmaktadır. Araştırmada, dünya dillerinin yapısal sınıflandırması (yalınlayan, bağlantılı, bükümlü) temel alınarak Türkçenin bu sınıflandırmadaki konumu ile tokenleştirme yöntemleri arasındaki ilişki ortaya konmuştur. Çalışma kapsamında karakter temelli, BPE temelli, WordPiece temelli, morfoloji temelli ve kelime temelli olmak üzere beş temel tokenleştirme yöntemi karşılaştırılmıştır. Her yöntemin hangi dil yapılarıyla uyumlu olduğu örnek kelime ve cümleler üzerinden gösterilmiştir. OpenAI'nın tiktoken kütüphanesi (Sürüm 0.13.0) kullanılarak GPT-4o'nun o200k_base ve GPT-4/GPT-3.5 turbo'nun cl100k_base kodlayıcılarıyla yapılan ölçümler, Türkçe cümlelerin İngilizce eşdeğerlerine kıyasla belirgin biçimde daha fazla token gerektirdiğini göstermiştir. Ayrıca Latin, Arap ve Kiril alfabeleriyle yazılan aynı Türkçe kelimelerin bayt büyüklüğü ve token sayısı bakımından farklılaştığı, özellikle Türkçeye özgü seslerin (ö, ü, ç, ğ) bu farkı derinleştirdiği tespit edilmiştir. Alanyazınında BERTurk, ITU TurkBERT, Zemberek, füzyon mimarisi ve TurkishTokenizer çalışmaları incelenmiştir. Araştırmanın sonucunda Türkçe için en uygun yöntemin, doğal dil işleme sözlüğü mevcut olduğunda morfoloji temelli, mevcut olmadığında ise WordPiece temelli tokenleştirmenin veya kök-ek yapısına duyarlı model mimarilerinin yaygınlaştırılması olduğu ortaya koyulmuştur.

Anahtar Sözcükler: Token, tokenleştirme yöntemleri, Türkçe.

Abstract

This research examines the token status of Turkish in the face of large language models. The transformer architecture underlying today's large language models, along with the BPE-based tokenization systems built upon it, has been optimized directly for the structure of English, as training datasets are predominantly English-based. This causes agglutinative, morphologically rich languages such as Turkish to consume significantly more tokens for the same content, thereby incurring higher processing costs. Building on the structural classification of world languages (isolating, agglutinative, polysynthetic, and inflectional), the study establishes the relationship between Turkish's position

¹ Bu araştırmanın gerçekleştirilmesinde önemli katkıları olan Muzaffer MAHI'ye teşekkür ederim.

²Arş. Gör., İstanbul Sabahattin Zaim Üniversitesi, Türkçe ve Sosyal Bilimler Eğitimi Bölümü, dila.can@izu.edu.tr, <https://orcid.org/0000-0001-7121-1962>

within this classification and various tokenization methods. Five fundamental tokenization methods—character-based, BPE-based, WordPiece-based, morphology-based, and word-based—are compared within the scope of the study, with the compatibility of each method to specific language structures demonstrated through sample words and sentences. Measurements conducted using OpenAI's tiktoken library (Version 0.13.0), employing GPT-4o's o200k_base encoder and GPT-4/GPT-3.5 turbo's cl100k_base encoder, show that Turkish sentences require markedly more tokens than their English equivalents. Furthermore, identical Turkish words written in Latin, Arabic, and Cyrillic alphabets are found to differ in byte size and token count, with sounds specific to Turkish (ö, ü, ç, ğ) deepening this discrepancy. Studies on BERTurk, ITU TurkBERT, Zemberek, fusion architecture, and TurkishTokenizer are reviewed in the literature. The findings indicate that the most suitable approach for Turkish is morphology-based tokenization where a natural language processing dictionary is available, or, in its absence, the wider adoption of WordPiece-based tokenization or model architectures sensitive to root-affix structure.

Keywords: Token, tokenization methods, Turkish.

GİRİŞ

Dil yapıları üzerine yapılan çözümlemeler Panini'den (MÖ. 5) başladığı varsayılararak 2026'ya kadar incelenmiştir. Dillerin akrabalıkları araştırılmış ve yapı benzerlikleri kelimelerin kök ve coğrafyalarında derinleşerek bağlantıları oluşturulmuştur. 21. yüzyılda bilgi temelli çağa girişin ardından büyük dil modelleri, makineye entegre edilerek bilgisayar dili oluşturulmuştur. 2023 yılında ortaya çıkan yapay zekâ algoritmasının altındaki büyük dil işleme modeli yıllarca üzerinde veri setleri ile çalışılmıştır. Bu veri setleri İngilizce olduğu için makine dil ve algoritmasının ana vatani, üzerine transformer de facto modeli kurgulanan İngilizcedir. İngilizce kurgulanan bu tasarı ve makinede tüm işlem hızları ve işlenen kelime parçacıkları da İngilizcenin en kârlı olacağı formatta tasarlanmıştır. Diğer dünya dillerini konuşan bireyler, bu erişilebilir teknoloji ve yapay zekâyı kendi dil yapıları doğrultusunda kullanmaktadır. Ancak bu durum çeşitli maliyet zorluklarını beraberinde getirmiştir. Her bir söz birimi olan token durumu, kullandığımız yapay zekaların arkasında bulunan işlemcinin dil modeline göre harcadığı maliyet hesabını göstermektedir.

Dilbilim, en kısa tanımı dili inceleyen bilim, dilin bilimidir. Dil nasıl insanın fizik, düşünce ve ruh yapısı ve çeşitli eylemleriyle ilişkiliyse, dilbilim de bütün bu eylemleri kendisine konu olarak alan öteki bilim dallarıyla sıkı sıkıya ilişkilidir. Fizik, fizyoloji, felsefe, ruhbilim, toplumbilim, insanbilim ve dahi etnoloji, coğrafya ve matematik gibi çok çeşitli alanlara giren sorunlara da eğilen bir bilim dalıdır (Aksan, 2000). Dil yapılarının temel inceleme konusu olan dilbilim, matematik ve ekonominin inceleme alanları ile birleşerek NLP (Doğal Dil İşleme) modelleri kapsamına girmektedir. NLP, her dilde farklı işlenmektedir. Bunun sebebi dünyadaki dillerin farklı morfolojik özelliklerinden kaynaklıdır.

Dünya dillerinin yapı bakımından sınıflandırılması üçe ayrılmaktadır. Bunlar yalınlayan (tek heceli) diller, bağlantılı (eklemeli) diller ve bükümlü dillerdir. Araştırmada, token ve tokenleştirme kavramlarını uygulamaya geçişinde anlaşılır kılmak için temel dil yapıları örneklerle açıklanmıştır.

Biçimsel olarak diller yalınlayan (tek heceli), bağlantılı (eklemeli) ve bükümlü diller olarak üç yapısal türde toplanır. Dil yapılarındaki bu üçlü ayırım eklerin kökle ve birbiriyle kaynaşma derecesine dayanan bir sınıflandırmadır. Sapir (1921), biçimsel yapının bundan bağımsız ikinci bir boyutunun daha bulunduğunu göstermiştir. Sözcük başına düşen biçimbirim sayısı ile ölçülen bireşim derecesini analitik, sentetik ve çok bireşimli olarak ayırmıştır. Greenberg (1960), bu ölçütü niceliğe dökmüş, iki eksenli çerçeveyi standart biçimde yerleştirmiştir. Dil yapılarındaki sınıflandırmalar dilleri yalınlayan, bağlantılı ve bükümlü diller olarak sınıflandırır, ancak bazı dillerin yeri analitik-sentetik özellikleri dikkate alınmadan tam olarak belirlenemez ki bu durum da gelecek araştırmalarda makine öğrenme sistemine etki eder. Aşağıdaki tabloda dillerin temel sınıfları yer almaktadır.

Tablo 1. Dillerin Yapısal Sınıflandırılması

Dil yapısı	Dil	Birleşim durumu	Açıklama
Yalınlayan	Çince	Analitik	Sözcükler büyük ölçüde tek biçimbirimlerden oluşur.
Yalınlayan	Vietnamca	Analitik	Sözcükler değişmez, zaman ve olumsuzluk gibi işlevler ayrı sözcüklerle gösterilir.
Bağlantılı	Türkçe	Sentetik	Değişmeyen bir köke ardışık eklerle tek sözcükte birçok dil bilgisi işlevi yüklenir.
Bağlantılı	Fince	Sentetik	Ad ve eylemler çok sayıda ek alır.
Bağlantılı	Grönlandca	Çok bireşimli	Kök ve çok sayıda ek bütün cümleyi tek sözcükte ifade ettirir.
Bükümlü	Arapça	Sentetik	Kök ünsüzler ünlü kalıbı ve çekim ekleri tek sözcükte kaynaşır.
Bükümlü	Almanca	Sentetik	Cinsiyet ve kişi çekimleri eklerle yapılır.
Bükümlü	İngilizce	Analitik	Köken olarak sentetik-bükümlü iken çekimini büyük ölçüde yitirmiştir.

(Sapir, 1921)

Sapir (1921), morfolojik yapının kaynaşma türünden bağımsız ikinci bir eksen bulunduğunu belirtmektedir. Bir dilin sözcük başına düşen biçimbirim sayısına göre analitik, sentetik ya da çok bireşimli olarak sınıflandırılabilceğini göstermiştir. Bu iki eksenin birbirinden bağımsız olması aynı kaynaşma türündeki dillerin farklı bireşim derecelerinde yer alabilmesi anlamına gelir. Örneğin bükümlü Arapça sentetik iken bükümlü İngilizce analitiktir, eklemeli Türkçe sentetik iken eklemeli Grönlandca çok bireşimlidir. Tokenleştirme açısından bu ayrım önemlidir, çünkü bir sözcüğün kaç parçaya bölüneceğini belirleyen etken dilin kaynaşma türüyle beraber bireşim derecesidir.

Yalınlayan dilleri temsil eden başlıca örnek Çince'dir. Çince'de çekim ve ek yok denecek kadar azdır; sözcükler büyük ölçüde değişmeden, tümce içinde sözcük sırasıyla birbirine yaklaşarak yeni anlam bütünleri oluşturur (Aksan, 2000). Çincenin yanı sıra Vietnamca, Tayca ve bazı Batı Afrika dilleri de yalınlayan (analitik) kategoride değerlendirilir.

Bağlantılı (eklemeli) diller, Türkçenin de içinde yer aldığı, ekleme yoluyla genişleyen dillerdir. Buradaki bağlantı, değişmeyen bir köke başka biçimbirimlerin açık sınırlarla art arda eklenmesidir (Aksan, 2000). Bu türün en bilinen örnekleri arasında Türkçe, Moğolca, Japonca, Fince ve Macarca yer alır. Kaynaştıran (çok bireşimli) diller ise kökü çeşitli eklerle zenginleştirerek bir sözcükte birçok kavramı, çoğu zaman bütün bir tümceyi ifade eden dillerdir; biçimbirim sınırları belirgin ve düzenli olduğundan bu diller, bağlantılı dillerin bir yan kategorisi olarak ele alınır (Aksan, 2000). Bazı görüşlere göre Grönlandca gibi çok bireşimli diller de eklemeli dillerin bir uç derecesi olarak görülmüştür (Shopen, 2007). Başka bir deyişle Grönlandca, birinci eksenle Türkçeyle aynı türdendir (eklemeli), aradaki fark dil yapısı değil bireşim derecesidir. Türkçe eklemeli-sentetik, Grönlandca ise eklemeli-çok bireşimli kategoride değerlendirilmektedir.

Bükümlü dillerde çekim, kökle kaynaşan eklerle gerçekleşmektedir. Bükümlü dillerde tek bir ek aynı anda birden çok dil bilgisi işlevini taşıyabildiğinden kök güçlülükle ayrılır ve kimi zaman kökün kendisi, özellikle kökteki ünlü değişir (Aksan, 2000). Bu türün belirgin örneği, kök-kalıp örgüye dayanan Arapçadır; ayrıca Yunanca, Roman dilleri ve Germen

dilleri de bükümlü özellikler taşır. İngilizce de köken olarak bu Germen (bükümlü) tabana dayanır ve Onan (2014) ile Eker (2005) tarafından bükümlü diller arasında değerlendirilir. İngilizce, bükümlü türde olmakla birlikte bireşim ekseninde büyük ölçüde analitiktir. Çekimini neredeyse tümüyle yitirmiş, geriye sing-sang-sung türü kök değişimi ile hem çokluğu hem üçüncü tekil kişiyi işaretleyen -s eki gibi bir kalıntı bırakmıştır. Dil bilgisi ilişkilerini ise yardımcı fiiller, edatlar ve sözcük sırasıyla, yani analitik yolla kurmuştur (Sapir, 1921). Dolayısıyla İngilizce "analitik-bükümlü" olarak nitelendirilmektedir. Latince ya da Rusça gibi tam bükümlü dillerden farkı tür değil derecedir. Bu diller sentetik-bükümlü iken İngilizce analitik-bükümlüdür.

Aşağıdaki tabloda bu dillerin yapı ve çekimleri gösterilmiştir. Çalışmanın devamında büyük dil modellerinin yöntem algoritmasında bu dil yapılarının sebep ve sonuçları ortaya koyulurken tabloya tekrar başvurulmuştur.

Tablo 2. Temel Dil Yapıları ve Örnekler

DİL YAPISI	BİREŞİM	DİL	ÖRNEK SÖZCÜK	ÖRNEK YAPI
Yalınla yan	Analitik	Çince	我写 wo şıye / yazıyorum, yazarım 我不写 wo bu şıye / yazmıyorum,	我 (ben) + 写 (yaz-) 我 (ben) + 不 (olumsuzluk) + 写 (yaz-)
	Analitik	Vietna mca	Tôi ăn cơm / yemek yerim Tôi đã ăn cơm / yemek yedim	Tôi ben · ăn ye- · cơm pirinç · đã (geçmiş) (ayrı sözcük)
Bağlan tılı	Sentetik	Türkçe	gezdirilebilmesi	gez-dir-il-e-bil-me-si
	Sentetik	Fince	taloissammekin / evlerimizde de	talo-i-ssa-mme-kim
Bükü mlü	Çok bireşimli	Grönla ndca	qajaliorpoq / o kayak yapay	qajaq-lior-poq (kayak +yapmak+o)
	Sentetik	Arapça	قَالَ qa:le / dedi يَقُولُ yequ:lū / der, söyler قُل qul / söyle, de	kök ünsüzler sabit (q, l)
	Sentetik	Alman ca	singen sang gesungen	kök ünlüsü bükülür (i-a- u)
	Analitik	İngilizce	sing / sang / sung foot / feet	kök ünlüsü değişir (büküm kalıntısı), dil bilgisi çoğunlukla analitik (will sing, of the book)

(Göksel ve Kerslake, 2005; Aksan, 2000; Sapir, 1921)

Dil yapılarından verilen örnekler, makine öğrenme sisteminin altında yatan dilbilimsel durumu ortaya koymaktadır. Makine öğrenmesi ile temel dilbilim arasındaki ilişki, teknolojik araçlarla dilin doğasını anlama çabasının karşılıklı etkileşimini içeren disiplinlerarası bir çalışmadır.

Makine Öğrenmesi

Makine öğrenmesi, matematiksel ve istatistiksel yöntemleri kullanarak mevcut verilerden çıkarımlar yapan, bu çıkarımlarla bilinmeyece dair tahminlerde bulunan bir veri madenciliği yöntemi olarak tanımlanmaktadır (Bergmann, 2026). Bu disiplin, bir bilgisayarın verileri analiz etmek, modeller oluşturmak ve kararlar vermek için öğrenmesini

sağlayan bir yapay zekâ alanı olarak da ifade edilmektedir (Bergmann, 2023). Makine öğrenimi, bilgisayarların, açıkça programlamadan karmaşık görevleri otomatikleştirmeyi öğrenmelerini sağlar. Belirli görevleri yerine getirmeleri için sağlanan verilerdeki örüntüleri ve ilişkileri keşfederek bu örüntüleri yeni durumlara uygulama yeteneği kazandırır. (Bergmann, 2023). Dilbilimsel açıdan bakıldığında ise makine öğrenmesinin, doğal dil işleme teknikleriyle iç içe geçmiş bir yapısı bulunmaktadır. Sebastiani'nin (2002), yazdığı makalede yapay zekânın temellerini dilbilim ile bağdaştırmıştır. Buna göre dil, karmaşık ve kuralları tam olarak tanımlanamayan bir veri türüdür. Nitekim bu bağlamda makine öğrenmesi de geleneksel kural tabanlı yaklaşımların aksine, makine öğrenmesi büyük metin yapılarından dil örüntülerini veri madenciliği ile öğrenerek metin sınıflandırma, analiz ve konuşma tanıma gibi görevlerde istatistiksel modeller oluşturur (Sebastiani, 2002). Buradan hareketle söylenebilir ki dilbilim, makine öğrenmesinin temel varsayımını kurduğu yerdir.

Makine öğrenmeleri sinir ağı mimarileri yoluyla oluşturulmaktadır. Standart kullanılan ChatGPT, Claude, Gemini yani GPT ve BERT modelleri transformer sinir ağı mimarisinin ürünüdür (Bergmann, 2026). Transformer, 2017'de "Attention is All You Need" makalesi ile yayınlanan (Vaswani vd., 2017) öz-dikkat mekanizmasıdır. Yani bugünkü büyük dil modellerinin standart temelini, transformer tabanlı dil modeli oluşturmaktadır. Model, bir cümledeki her bir kelimeyi işlerken o kelimenin diğer bütün kelimelerle ilişkisine (sıklığına) aynı anda bakmaktadır. Ancak transformer model, metni doğrudan işleme kapasitesine sahip değildir. Önce metin sayısal birimlere yani tokenlere çevrilmesi gerekir. Bu sayısal birimlerin temsili tokenler modelle bağlı bileşenlerdir.

Harris (1954) tarafından yapılan araştırmada, modern dil modelleri sözcükleri anlamlandırma biçimi, bir sözcüğün anlamının onu çevreleyen bağlamdan türediğini öne süren dağılımsal varsayıma dayanmaktadır. Bugün ise embedding denilen gömü, temsil öğrenme mimarisidir ve dilbilimsel ilkelerin matematikselleştirilmiş halidir. Buradan hareketle dilin makineyle en somut biçimde karşılaştığı yer *tokenleştirme* aşamasıdır. Tokenleştirmede metin, model henüz devreye girmeden alt-sözcük birimlerine bölünür ve bölümlemenin nasıl gerçekleşeceğini belirleyen şey doğrudan dilin biçimbirimsel durumudur (Sennrich vd., 2016). Bu durumda da tokenleştirme maliyetleri dil yapısının alt-sözcük durumuna göre farklılık göstermektedir.

Dilin doğal sınırlarında bulunan kelime, hece ve harf gibi birimler, tokenlerde farklı işlenmektedir. Token, doğal dil işleme mekanizmasının kurgulandığı veri setindeki istatistik doğrultusunda kelime ile harfin arasındaki birimlerdir (Haslett, 2025). Yapay zekâ sistemlerinin altında yatan verilerin, işleme mekanizması tokeni yani anlam birimini oluşturur. Bir benzetme ile örnek verilecek olursa token, yapı mimarisinde kullanılan her bir tuğladır. Tuğlayı oluşturan taneciklerin miktarı tuğlada olduğu gibi görünmez. Var olan somut gerçek yalnızca tuğla parçasıdır. Token her bir tuğlanın birleşimiyle ortaya çıkan veri setindeki bölümlenmelerdir. Dolayısıyla, eldeki maliyete ve ihtiyaca göre o işe token harcanması gerektiği analiz edilir. Nitekim bu parçaların üretim menşei de İngilizce olduğu için, İngilizce ve benzeri yapıdaki dillerde çok daha az token harcanmaktadır. Bu farklılık aşağıdaki tabloda örneklerle açıklanmıştır.

Tablo 3. Transformer Tabanlı BPE Modeline Göre Dilin Tokenleri

DİL	SÖZCÜK	YAPI GÖRÜNÜMÜ	TOKEN AYRIMI	TOKEN SAYISI
Türkçe (eklemeli)	kitaplarımızdan	kitap-lar-ımız-dan	kit+ap+larımız+dan	4
Türkçe (eklemeli)	bağımsızlık	ba-ğım-sız-lık	ba+ğı+ms+ız+lık	5
İngilizce	detokenization	de-token-ization	det+oken+ization	3

(bükümlü-analitik)				
İngilizce (bükümlü-analitik)	independence	in-de-pen-dence	ind+epend+ence	3
Almanca (bükümlü-sentetik)	Unabh�ngigkeit	un-abh�ng-ig-keit	Un+abh+�ng+igkeit	4

Tiktoken (S r m 0.13.0), o200k_base ile hesaplanmıřtır (OpenAI, 2026).

Tablo 3'te g r ld ė   zere BPE token ayrımı, s zc ė n bi imbirim sınırlarıyla  rt řmez; model s zc ė  dil bilgisel yapısına g re deėil, derlemdeki sıklıėına g re b ler.  rneėin *baėımsızlık* s zc ė  bi imbirimsel olarak *baė-im-sız-lık* bi iminde c z mlenirken, BPE bunu *ba+ėı+ms+ız+lık* olarak beř par aya ayırır ve k k *baė* korunmaz. Aynı durum İngilizce ve Almanca  rneklerinde de g r l r. BPE modelinde tokenlerin dilin bi imbirimsel  zelliklerine g re deėil, veri setini iřleyen modelin ortaya koyduėu d ř nmeye/frekansa g re kelimelere tokenlere ayırdıėı bu  rneklerle g sterilmiřtir. Ayrımlar yalnızca tablodaki  rneklerle ait deėildir. BPE (GPT-4o) modelinin tokenleřtirme sonu ları eklemeli dillerde bi imbirimsel yapıdan baėımsız sonu lar doėurmaktadır. Bunun sebebi, modelin d ř nme algoritmasındaki veri setinde ge en s z birimlerinin sıklıėıdır. Ancak token ayrımı dil yapısından baėımsız deėildir. Aynı i erik bazı dillerde 2 ile 15 kat daha fazla token gerektirmektedir (Petrov vd., 2023). Bu durum da iřlem y k  ve maliyet artıřına sebep olmaktadır. Ařaėıdaki c mle  rneklerinde durum a ıklanmıřtır.

Tablo 4. Transformer Tabanlı BPE Modeline G re C mlelerin Tokenleri

DİL	C�MLE	TOKEN AYRIMI	KELİME SAYISI	TOKEN SAYISI
T�rk�e	Ben İstanbul'da bir �niversitede �alıřıyorum	Ben + İstanbul + 'da + bir + �n + ivers + ited + e + �alıř + ıyorum	5	11
İngilizce	I work at a university in Istanbul	I + work + at + a + university + in + İstanbul	7	7
T�rk�e	Kitaplarımızdan bahsetti	Kit+ap+larımız+dan +bah+set+ti	2	7
İngilizce	She talked about our books	she + talked +about+our+books	5	5
T�rk�e	T�rk�e Eėitimi, 1989 yılında Gazi �niversitesinde kuruldu	T�r + k + �e + Eė + it + imi + 198 + 9 + yılında + G + azi + �nivers + ites + inde + kur + uldu	7	16
İngilizce	Turkish Language Education was established at Gazi University in 1989.	Tur + kish + language + education + was + established + at + G + azi + University + in + 198 + 9	10	13

Tiktoken (S r m 0.13.0), o200k_base ile hesaplanmıřtır (OpenAI, 2026).

Tablo 4'te bulunan  rnekler kelime sayısına oranla T rk e token miktarının olduk a y ksek seviyede olduėunu g stermektedir. Bu  alıřmanın amacı, tokenlerin T rk e  zerindeki iřleme sistemini belirlemek ve T rk enin dil yapısına uygun modeli arařtırmaktadır. Arařtırma, T rk e merkezli olup T rk enin bilgisayar sistemindeki konumunu belirlemek ve dilbilimsel bir bakıř a ısıyla avantaj ve dezavantajlarını g stermektedir. Yapay zek  sistemleri i erisindeki s re lerde T rk enin konumunu

belirlemek, çalışmanın çerçevesini oluşturmaktadır. Bu kapsamda aşağıdaki problem sorularına yer verilmiştir.

1. Yapay zekâ algoritması neden Türkçede daha fazla token harcamaktadır?
2. Yapay zekâ algoritmalarında Türkçenin daha az token harcaması için ne gereklidir?

YÖNTEM

Araştırmanın Modeli

Bu çalışmada Türkçenin dil yapısı bakımından yapay zekâ araçlarındaki temel durumu incelenmiştir. Merriam'a (2013) göre durum (vaka) araştırmaları sınırlı bir sistemdir. Bu bakımdan çalışmanın karakteristik özelliği, çalışmanın sınırlandırılmasıdır. Durum çalışmasının doğası gereği araştırmanın sınırı tokenleştirme yöntemlerinin incelenmesi ve açıklamasını içermektedir. Tokenleştirme yöntemlerinin dışındaki teori ve sistemlere değinilse de araştırmanın temelini oluşturmamaktadır. Bu kapsamda 2026 yılında kullanılan temel tokenleştirme yöntemlerinden *karakter temelli*, *BPE temelli*, *WordPiece temelli*, *morfoloji temelli* ve *kelime temelli* tokenler işlenmiştir. Tokenleştirme dışındaki dil modeli teorileri çalışmada arka plan bilgisi olarak yer almış olup çalışmanın odağını oluşturmamıştır.

Verilerin Toplanması ve Analizi

Veri toplama sürecinde iki kaynak türü kullanılmıştır: alanyazın taraması ve doğrudan tokenizer ölçümleri. Alanyazın taraması kapsamında Toraman vd. (2022) tarafından geliştirilen BERTurk modelinin beş tokenizasyon yöntemiyle karşılaştırıldığı çalışma, Kaya ve Tantıuş'un (2024) ITU TurkBERT modelini 32K, 64K, 128K ve 256K sözlük boyutlarıyla eğittiği ve Zemberek çözümleyicisini entegre ettiği çalışma, Turan'ın (2025) Zemberek sözlüğünü WordPiece alt yapısıyla birleştiren füzyon mimarisi, Bayram vd.'nin (2025) gemma-2, llama-3.1, Qwen2.5 ve aya-expanse gibi mevcut dil modellerini %TR ve %Pure ölçütleriyle değerlendirdiği çalışma ile Bayram vd.'nin (2026) geliştirdiği ve %99,48 başarı oranına ulaşan melez morfoloji-BPE tabanlı TurkishTokenizer modeli incelenmiştir.

Doğrudan ölçüm boyutunda ise OpenAI'nın açık kaynaklı tiktoken kütüphanesi (Sürüm 0.13.0) kullanılmıştır. Ölçümler, GPT-4o modelinin kullandığı o200k_base kodlayıcısı (sözlük boyutu: 200.019) ile GPT-4 ve GPT-3.5 turbo modellerinin kullandığı cl100k_base kodlayıcısı (sözlük boyutu: 100.277) üzerinden 6 Temmuz 2026 tarihinde gerçekleştirilmiştir. Örneklem, Türkçe ve İngilizce eşdeğer kelime ve cümlelerden; ayrıca Türkçeye özgü seslerin (ö, ü, ç, ğ) yer aldığı kelimelerin Latin, Arap ve Kiril alfabeleriyle yazılmış biçimlerinden oluşturulmuştur. Karşılaştırma amacıyla, GPT-2/3/4, LLaMA, Mistral, RoBERTa ve BLOOM gibi modellerin temelini oluşturan BPE yöntemi ile mBERT'in temelini oluşturan WordPiece yöntemi de örnekler üzerinden analiz edilmiştir. Kelime sayıları boşlukla ayrılan birimler esas alınarak belirlenmiş, elde edilen veriler yalnızca OpenAI tokenizer'larına özgü kabul edilmiştir. LLaMA ve Qwen gibi yerel modeller farklı tokenizer yapıları nedeniyle bu ölçümlerin dışında tutulmuştur. Verilerin analizinde betimsel karşılaştırma tekniği uygulanmıştır. Tokenizasyon yöntemlerinin dil yapıları (yalınlayan, bağlantılı, bükümlü) ile uyumu, örnek kelime, cümle ve yazı sistemi (alfabe) karşılaştırmaları üzerinden değerlendirilmiştir.

Veri Kaynağı

Tokenleştirme yöntemleri Toraman vd., (2022) ve Mielke, vd., (2021) tarafından yapılan çalışmadaki temel modeller kullanılmıştır. Bu çalışmalar *tokenizer* olarak tanımlanan modellerdeki kelime, alt kelime ve hibrit yaklaşımların gelişimini barındıran standart veri niteliği taşımaktadır. Yapay zekâ sistemlerinin kullanıma açılmasıyla birlikte dil yapılarına

uygun tokenleştirme modelleri ve hesaplamaları ulusal ve uluslararası alanda çalışılmaya başlamıştır. Bunlar için dil yapısına uygun doğal dil işleme sözlük kütüphanelerinden yararlanılmış, frekansa dayalı sıklık modelleri geliştirilmiş/egitilmiş veya yeni mimariler oluşturulmuştur. Bu çalışmaların tamamı, alan yazınında yapılan denemelerin ürünleri olup günlük hayatta kullanılan yapay zekâ uygulamalarından yola çıkılarak tasarlanmıştır. Bu çalışmanın veri kaynağında tokenleştirme modellerinde alanda yapılan çalışmalar esas alınmış olup yapılan ölçümler açık kaynaklı ve standart tiktokenizer ölçümleridir.

Geçerlik ve Güvenirlik

Türkçe ve diğer dillerde yapılan *tiktokenizer* ölçümleri GPT modelinin temelini oluşturan açık kaynaklı ölçümlerdir. Dolayısıyla bu ölçümler yeni model ve yöntemler geliştirilip kullanıma sürülene kadar geçerliliğini korumaktadır. Nitekim yeni modellerin ortaya çıkmasından sonra dahi tokenizer ölçümleri eski sistem üzerinden ölçme seçeneğini sunmaktadır. Örneğin bu çalışmada “*o200k_base*” ve “*cl100k_base*” olarak biri diğerinden daha güncel olan 2 model ölçümü kullanılmıştır. Bu durum çalışmanın geçerliliğini korumasına ve sürdürülebilirliğine imkân tanımıştır. Geçerlilik, araştırma bulgularının dış dünyadaki gerçekliğe uyup uymadığı sorunsalıyla ilgilenir. Geçerlik, bir üründen çok bir amaçtır ve kanıtlanabilecek ya da olmuş/gerçekleşmiş gözülle bakılabilecek bir şey değildir (Mariam, 2013; Maxwell, 2005). Nitel geçerlik araştırmacının belirli süreçleri uygulayarak bulguların doğruluğunun test edilmesi işlemini yapma yöntemiyle nitel güvenilirlik araştırmacının yaklaşımının farklı araştırmacılar ve farklı projeler arasında tutarlı olmasıdır (Creswell ve Creswell, 2021; Gibbs, 2007).

Bu çalışmada, modellerin temel kuramına ve frekans yapılarına müdahale edilmemiştir ve yeni bir model/mimari geliştirilmediği için güvenirliliği tehdit edecek herhangi bir unsur bulunmamaktadır. Araştırma temel sistemlerin Türkçeye uygunluğunu dilbilimsel ve makine öğrenmesi alanlarıyla karşılaştırmakta ve buna göre konumlandırmaktadır. Ölçümler internet ortamında herkese açık kaynaklı olarak tiktokenizer tarafından teyit edilebilmektedir. Aynı zamanda ölçümlerin ulaştığı sonuçlar ve Türkçenin token uyumluluğu alan yazınının tartışma bölümüyle uyumluluk göstermektedir. Bu durum (ve tutarlılık) araştırmacının güvenirliliğini sağladığını göstermektedir.

BULGULAR

Bu araştırmada yapay zekanın Türkçede daha fazla token harcadığı ortaya koyulmuştur.

Karakter bayt düzeyi (eğitim gerektirmez)	BPE alt-kelime (frekans tabanlı)	WordPiece alt-kelime (frekans tabanlı)	Morfoloji kök + ek (dilbilgisi tabanlı)	Kelime tam kelime (eğitim gerektirmez)
---	--	--	---	--

Bu problemten yola çıkarak yapay zekâ algoritmasının neden Türkçede daha fazla token harcadığını bulmak için tokenleştirme modelleri araştırılmıştır. Araştırmada tokenleştirmenin temelde 5 yönteme dayandığı Toraman vd. (2022) tarafından ortaya koyulmuştur.

en küçük birim

en büyük birim

Şekil 1. Tokenleştirme Yöntemleri

(Toraman vd., 2022)

Şekil 1’de yer alan yöntemlerden *karakter* harf düzeyini temsil etmektedir. *BPE* ve *WordPiece* frekans temelli tokenleştirme yöntemleridir. Toraman vd. (2022)’ye göre BPE ve WordPiece tokenleştiricinin eğitimi sırasında karşılaşılmayan kelimeleri tokenize edebilir ve tek tek tokenlerin modelleme gücünü artırmaya çalışan sıklığa dayalı algoritmalarlardır. BPE alt kelimeleri (güncel kullanılan token mekanizmasını) temsil eder, bayt temellidir ancak dilbilgisel olarak yorumlamaya kapalıdır (Bayram, vd. 2025a). Sistem, kelimeleri anlamdan değil veri setinde geçen kelimelerin sıklığından alır. Ancak BPE en sık çifti birleştirirken WordPiece eğitim verisi doğrultusunda olabilecek en doğru çifti seçer (Schuster ve Nakaima, 2012). Bu bakımdan WordPiece kodlayıcı temellidir.

Morfoloji yöntemi kök ve ek ayrımlarını tokenler olarak kullanır. Dilbilgisel olarak anlamlı ve yorumlanabilir parçalar elde eder ancak kök kelimeler bölünmediği için geniş bir dil modeli sözlüğüne bağlı kalması gerekir. *Kelime* yöntemi ise her bir kelimenin sayıldığı token mekanizmasıdır. Her kelime bir tokeni temsil eder ancak morfolojik yöntemle kıyasla çok daha büyük çaplı bir sözlük seti gerektirir.

Bu bakımdan karşılaştırıldığında her bir tokenleştirme yönteminin dilbilimsel bakımdan uyumlu olduğu dil yapıları bulunmaktadır. Öncelikle bu modellerin her birinde Türkçenin entegre edilmesi durumunda ortaya çıkacak token yapısı Tablo 5’te ortaya koyulmuştur.

Tablo 5. Tokenleştirme Yöntemine Göre Türkçe Kelime Yapısı

Model	Kelime	Token Yapısı	Token Sayısı
Karakter	kalemlik	k-a-l-e-m-l-i-k	8
BPE (GPT-4o)	kalemlik	kal-em-lik	3
WordPiece (mBERT)	kalemlik	kale-mli-k	3
Morfoloji	kalemlik	kalem-lik	2
Kelime	kalemlik	kalemlik	1

(BPE = GPT-4o/o200k, WordPiece = mBERT; morfoloji ve kelime satırları yazarın dilbilgisel çözümlemesi)

Yukarıda verilen tabloda kalemlik kelimesinin tokenleştirme yöntemi bakımından farkları gösterilmiştir. BPE ve WordPiece, frekans tabanlı tokenleştirme modelleri olduğu için frekansın eğitilmesi durumunda tablodakinden farklı sonuçlar verebilir. Bu tokenleştirmeler 2026 yılında yapay zekâ araçları tarafından standart kullanılan GPT-4o ve mBERT örneği verilmiştir.

Aşağıdaki örneklerde her bir tokenleştirme yönteminin avantajlı ve dezavantajlı olduğu diller verilmiştir.

Tablo 6. Tokenleştirme Yöntemlerine Dil Örnekleri

Yöntem	Avantajlı Dil (Örnek)	Sebebi	Dezavantajlı Dil (Örnek)	Sebebi
Karakter	Çince	Çince’de her bir karakter anlamlı bir sözcüğü temsil eder	İngilizce-Türkçe	Karakter temelli birimler anlam ifade etmez.
BPE (GPT-4o)	İngilizce	İngilizce’de her bir kelime neredeyse 1 veya 1.2 tokene tekabül eder	Türkçe,	Eklemelilerde morfeyle uyumsuz fazla token üretir
WordPiece (mBERT)	İngilizce	Kök ve ekler mBERT modelinde İngilizce tabanlı frekanslarla kurulmuştur.	Çince, Japonca	Kelime sınırı yoktur, karakter merkezli fazla token üretir
Morfoloji	Türkçe,	Kök kelime korunarak	Çince	Ayrıştıracak bir

	Korece,	ekler dilbilgisel temelli token alır			morfem olmadığı için avantaj sağlamaz.
Kelime	İngilizce	Kelime ekleri azdır, Türkçe dolayısıyla kurulabilir.			Eklmeli yapıdan dolayı hacmi genişler.

(Toraman vd., 2022'den yararlanılarak yazar tarafından oluşturulmuştur).

Yukarıdaki tabloda dil modellerinin temelli örnek dil yapılarını temsil eden dillerle gösterilmiştir. Bu bakımdan karakter temelli tokenleştirme, tek heceli yani karakterin anlam ifade ettiği dillerden olan Çince, Tibetçe, Vietnam, Himalaya, Endonezya ve Afrika dillerine avantaj sağlar.

BPE modeli ise *GPT-2/3/4*, *LLaMA*, *Mistral*, *RoBERTa*, *BLOOM* gibi byte temelli ChatGPT, Gemini ve Claude gibi uygulamalar tarafından kullanılır. Model doğrudan İngilizce için kurgulanmıştır ve düşük token miktarları ile avantaj sağlar. Ancak Türkçe ve Arapça gibi bükümlü ve eklemeli dillerde kelime başı token harcama potansiyeli fazla olduğu için dezavantajlıdır.

WordPiece, kodlama temelli uygulamadır. Karakter ve kelimeler arasındaki sıklığa bakar ancak içgüdüsel bir tasarım olduğu için doğru eğitimlerle token sayılarını değiştirme potansiyeli vardır. WordPiece tokenleştirme yönteminde kök ve ekler kodlamalar ile değiştirildiği durumda avantajlı ve dezavantajlı olduğu diller değişim gösterebilir. Ancak kodlayıcı, kelime kök ve ekleri seçmede veri setindeki boşluklardan yararlanır. Bu bakımdan da boşluk olmayan dillerde WordPiece yine fazla token üretir. Dolayısıyla Japonca ve Çince karakterlerde dezavantajlı durumdadır.

Morfoloji temelli tokenleştirmede kök kelime korunur, kelimeye gelen ekler dilbilgisel olarak ayrılır. Dolayısıyla alacağı tokeni önceden hesaplamak mümkündür ve bu da modeli diğerlerine kıyasla daha kontrol edilebilir bir noktada tutar, ancak karakter temelli dillerde dezavantajlı bulunmaktadır. Ayrıca model için bir doğal dil işleme kütüphanesi kurulması gereklidir.

Kelime temelli tokenleştirme yönteminde ise token sayısı bakımından avantaj-dezavantaj doğrudan izlenmemektedir. Kelime yöntemi, doğal dil işleme sözlüğü bakımından geliştiricileri zorlamaktadır. Bu bakımdan mevcudiyetinde az kelimesi bulunan diller avantajlıyken çok kelimesi bulunan zengin diller dezavantajlı durumdadır. Dolayısıyla sözlük kütüphanesinin bulunmadığı durumlarda modelin içeriği hakkında yoruma kapalıdır.

Dil yapılarının avantajlı ve dezavantajlı durumları farklıdır. Aynı dil yapısındaki dillerin token avantajları model türüne göre farklılık gösterir. Burada da bakılması gereken makinenin odaklandığı dilin yazı sistemi ve dil yapısıdır. Dil yapısı aynı olsa bile yazı sistemi farklı olan ekleme dillerde token avantajı değişiklik göstermektedir. Bunu örneklemek için Türkçe “evlerimizden” kelimesinin günümüzde kullanılan olan 3 alfabeden Arap, Kiril ve Latin harfleri ile yazımının örneği verilmiştir.

Tablo 7. “Evlerimizden” Kelimesinin Farklı Yazımlardaki BPE Model Token Görünümü

Yazı	Kelime	Karakter sayısı	Bayt	o200k	cl100k	Parçalanma (o200k)
Latin	evlerimizden	12	12	4	5	ev-ler-imiz-den
Arap	اولریمزین	9	18	4	8	اول · ریم · زد · ن
Kiril	евлеримизден	12	24	4	7	ев-лер-имиз-ден

Tiktoken (Sürüm 0.13.0); o200k_base ve cl100k_base ile 6 Temmuz 2026'da ölçülmüştür (OpenAI, 2026).

Karşılaştırma için verilen eski değerler GPT-4 ve GPT-3.5 turbo modellerinin kullandığı "cl100k_base" kodlayıcısına (sözlük boyutu: 100.277) aittir. Kelime sayıları, boşlukla ayrılan birimler esas alınarak belirlenmiştir. Verilen değerler yalnızca OpenAI tokenleştirmesine özgüdür. LLaMA veya Qwen gibi yerel modellerin tokenleştirme yöntemleri Türkçeye farklı bir sonuç verebilir veya modeller güncellendikçe dil yapıları arasındaki token miktarları değişiklik gösterebilir.

Tablo 7'de görüldüğü üzere aynı kelime ve ekleme yapısının cl100k modelinde farklı token miktarı bulunmaktadır. Karakter sayısı, kelimenin bilgisayarda kapladığı (bayt) alanı değiştirmektedir. o200k (güncel tokenizer) kelimeleri eşit olarak 4 token görmüştür ancak bu görünüm yüzeysel bir eşitliktir. Kelimelerin karakter sayısını ve bayt büyüklüğü eski tokenleştirme olan cl100k üzerinde görülmektedir. Bu durumda dezavantajlı fazla yer kaplayan ve dolayısıyla fazla maliyete yol açan yazı sistemi Kiril alfabesidir. Çünkü güncel BPE modelinin çalışma sisteminde bayt temelli büyüklük kelimeler arası maliyet farkını oluşturmaktadır.

Türkçeye özel ünlüler bakımından durum incelendiğinde ise sonuç alfabe bakımından daha farklı sonuç vermiştir. Aşağıdaki tabloda "çöplüğümüzü" kelimesine örnek verilmiştir. Bu örneğin seçilme sebebi, "ö", "ü", "ç", "ğ" gibi Türkçeye özgü sesleri bir arada taşımasıdır.

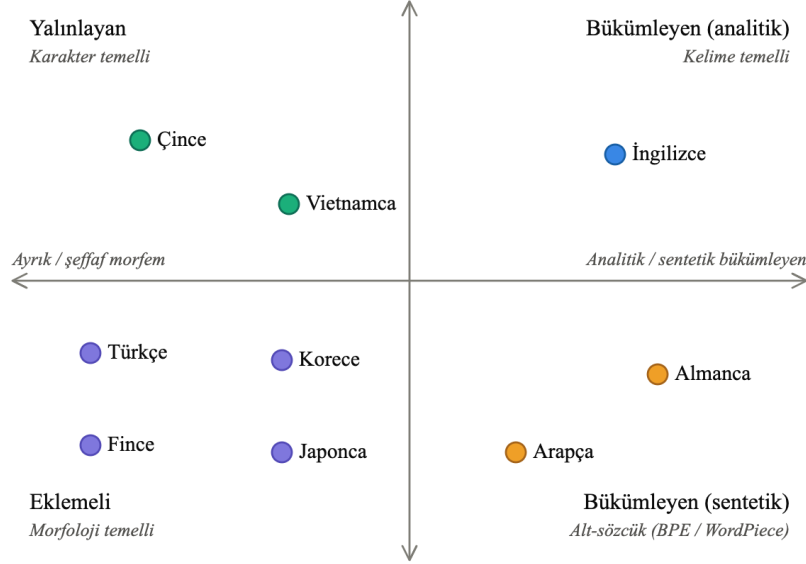
Tablo 8. Çöplüğümüzü Kelimesinin Farklı Yazımlardaki BPE Model Token Görünümü

Yazı	Kelime	Karakter sayısı	Bayt	o200k	cl100k	Parçalanma (o200k)
Latin	çöplüğümüzü	11	18	6	8	ç-ö-pl-üğ-ümüz-ü
Arap	چوپلگمیزی	8	16	6	9	چ·و·پل·گ·م·زى
Kiril	чоплүгүмүзү	11	22	8	17	ч-өп-л-ү-г-үм-үз-ү

Tiktoken (Sürüm 0.13.0); o200k_base ve cl100k_base ile 6 Temmuz 2026'da ölçülmüştür (OpenAI, 2026).

Bu örnekte güncel token sayıları Latin ve Arap alfabesinde 6, Kiril alfabesinde 8 olarak görülmektedir. Karakter sayıları ve bayt büyüklükleri evlerimizden örneğine göre farklı işlemiştir. Çünkü çöplüğümüzü kelimesinin Arap alfabesindeki yazımında kısa ünlüler düşmektedir. Ancak Arap alfabesinde az token ve düşük bayt miktarı maliyet açısından uygun görülse bile dilbilgisel açıdan tamamen belirsizdir. Nitekim bu örnekte Latin ve Kiril alfabesindeki bölünmeler de morfolojik açıdan anlamsız birimlere ve fazla bölünmelere yol açmıştır. BPE temelli tokenleştirme yönteminin "çöplüğümüzü" veya "evlerimizden" gibi tek kelimelik birimlerde 4-6 token harcaması, yapının Türkçe ile tamamen uyumsuz olduğunu gösterir. Aynı yapıdaki diller olmasına rağmen makine öğrenme sisteminin yazı ve karakter farklılıkları bu örneklerle ortaya koyulmuştur.

Buraya kadar çalışmada dil yapılarında Türkçe ile uyumlu sistemlerin, maliyet açısından en avantajlı olan morfoloji ve kelime temelli tokenleştirme ile yapılabileceği görülmüştür. Morfoloji ve kelime temelli tokenleştirmenin uygulanmamasındaki sebebin doğal dil işleme sözlük kütüphanesinin bulunmaması ve bu sözlükleri oluşturmanın aldığı zamandan kaynaklandığı ortaya koyulmuştur. Dolayısıyla, doğal dil işleme sözlüklerinin mevcut olduğu varsayılırsa, yapılara göre dillerin uyumlu olduğu tokenleştirme modeli aşağıdaki şema örneğiyle gösterilmiştir.



Şekil 2. Dil Yapılarının Tokenleştirme Uygunluğu

Yazar tarafından oluşturulmuştur; Aksan, (2000) ve Toraman vd., (2022) çalışmalarından yararlanılmıştır.

Şekil 2'ye göre alınlayan dillerin karakter temelliye, eklemeli dillerin morfoloji temelliye, bükümleyen dillerin alt sözcük temelliye uygun olduğu görülmektedir. Çince ve Vietnamca karakter köşesinde olmasına rağmen Vietnamca Latin alfabeli olduğu için kelime temelliye uygunluğu Çinceye göre daha yakındır.

Arapça, yapı olarak bükümleyen bir dildir ancak çekimlenmesi bakımından morfoloji temelli eğilim göstermektedir (Atlı, 2015), bu sebeple morfoloji temelli yapıya da yakın konumlandırılmıştır. Korece ve Japonca ise eklemeli yapıda olmasına rağmen yazı sistemlerindeki farktan dolayı BPE modellerinde ek alan/bayt gerektirmektedir. Bu yazı sistemi farkları Korece ve Japonca'yı diğer modellere de yakınlaştırmaktadır.

Doğal dil işleme sözlüklerinin bulunması durumunda en uygun tokenleştirme yöntemi dil yapılarına göre bu şekilde belirlenmiştir. Sözlüğün bulunmaması durumunda ise var olan tokenleştirme yöntemlerinin (BPE, WordPiece) modellerin mBERT, TurkBERT gibi alt sözcük mekaniğinin eğitilmesi, frekans tabanlı modellerde mümkündür. Bu bulgular alanda Türkçe için yapılan tokenleştirme çalışmalarından elde edilen çıkarımlardır.

Toraman vd. (2022) tarafından yapılan çalışmada beş tokenleştirme yöntemi (karakter, BPE, WordPiece, morfoloji ve sözcük) ayrı ayrı eğitilerek Türkçe için orta ölçekli dil modelleri (RoBERTa-TR-medium) OSCAR derlemi üzerinde önceden eğitilmiş ve bu modeller altı görevde karşılaştırılmıştır; karşılaştırmada dbmdz ekibince geliştirilen ve WordPiece tabanlı BERTurk temel model (baseline) olarak kullanılmıştır. Söz konusu görevler haber sınıflandırma, nefret söylemi tespiti, duygu analizi, varlık tanıma, metin benzerliği ve doğal dil çıkarımıdır. Çalışmanın bulgularına göre fiili standart alt-sözcük yöntemleri olan WordPiece ve BPE genel olarak en başarılı sonuçları vermiş ve birbirine çok yakın performans göstermiştir. Morfoloji temelli yöntem bu iki yöntemle rekabet edebilecek düzeyde olmakla birlikte bir miktar geride kalmış (aralarındaki fark yaklaşık 0.01-0.02

düzeyindedir); sözcük temelli yöntem sözlük dışı (bilinmeyen) birimler nedeniyle, karakter temelli yöntem ise bu ölçekte daha düşük başarı göstermiştir. Morfoloji temelli tokenleştirilmenin Türkçe için uygulanabilir olması, bu yöntemin dayandığı biçimbirimsel çözümlemeyi sağlayan Zemberek (Akın ve Akın, 2007) gibi açık kaynaklı doğal dil işleme araçları sayesinde mümkün olmaktadır.

Kaya ve Tantuğ (2024), tarafından yapılan çalışmada WordPiece'e odaklanılmıştır. WordPiece'in sözlük boyutu üzerinden değişken belirlenmiştir. ITUTurkBERT adındaki BERT modellerini 32K, 64K, 128K, 256K sözlüklerle eğitirler. Ek olarak morfolojik etiketleme olarak da *Zemberek* çözümleyicisi eklemeyi denerler. Bu çalışmadaki bulguda sözlük büyüdükçe ayrıntılar düşmektedir. Yani sözlük ne kadar büyürse kelimeler o kadar az parçalara ayrılır. Bu durum da Türkçeye dezavantaj getirmektedir.

Turan (2025), yüksek lisans tezinde yeni bir tokenleştirme yöntemi önermekten öte Türkçe için bir füzyon mimarisi kurar. Türkçenin eklemeli dil yapısının gerektirdiği geniş sözlük havuzunu kurmak yerine *Zemberek* sözlüğündeki kelimeleri alt parçalara bölerek WordPiece alt yapısını kullanmaktadır.

Sezer (2025), Türkçe Eğitimi alanında bilişimsel dilbilim çalışmıştır. Araştırma kapsamında Türkçe bir metnin Türkçenin yapısına uygun bir şekilde birimlendirilmesi için kullanılabilecek verimli bir işlemler dizisi tasarımı oluşturmuştur. Farklı sözlükler ve düzenli ifadelerden yararlanılarak, hibrit bir yaklaşımla tasarlanacak bir birimlendiricinin (tokenleştiricinin) daha verimli olduğu sonuçlara ulaşacağını tespit etmiştir. Nitekim Bayram, vd. (2025b) çalışması da ileride bu hibrit yaklaşım ile tasarlanacak TurkishTokenizer'dir.

Bayram vd. (2025a), çalışmasında, var olan dil modellerini (gemma-2, llama-3.1, Qwen2.5, aya-expanse) değerlendirmektedir. Bu çalışma yeni bir model önermez ancak bir ölçüt önermesi olarak değerlendirilmektedir. Bu bakımdan yöntemlerin çıktısının Türkçeye hizalanması bakımından dengeyi %TR (üretilen token'ların geçerli Türkçe kelime/morfem olma oranı) ve %Pure (token'ların bölünemez kök/ek birimleriyle hizalanma oranı) üzerinden değerlendirmektedir. Bu çalışmada bir tokenizer Türkçeye ne kadar uygun bölerse, model o kadar başarılı olmaktadır.

Bayram vd. (2025b), tarafından yapılan çalışmada TurkishTokenizer adında Türkçe tokenleştirme modeli kurmuştur. Bu model hazır yöntemlerden daha iyi çalışmakta ve kelimeyi tam kök ve eklerinde ayırmaktadır. 2025 yılında yapılan çalışmanın devamı niteliğindedir. Yöntem morfoloji öncelikle melez bir tokenizerdir. Morfoloji tabanlı sistem ile sözlük dışı BPE yedeği oluşturmaktadır. Yani morfoloji ile BPE karmasıdır. Ardından bu modeli ölçmek için 2025 yılında geliştirilen ölçüt ile ölçer ve %99.48 oranı elde eder.

TARTIŞMA VE SONUÇ

Bu çalışmada Türkiye'de yapay zekânın token sistemi üzerine Türkçe için yürütülen çalışmalar açıklanmış ve temel tokenleştirme yöntemlerinin dil yapılarına uygunluğu incelenmiştir. ITU TurkBERT, TurkishTokenizer ve füzyon mimarisi, Türkçeye üstünlük sağlayan gelişmeler olarak öne çıkmaktadır. 2026 yılı itibarıyla yürütülen bu çalışmalar, bir bakıma Türkçenin makineye kaç mâl olduğu üzerine yapılan çözümlemelerdir. Araştırmanın başında sorulan iki soruya bulgular ışığında doğrudan yanıt verilebilir. Yapay zekânın Türkçede neden daha fazla token harcadığı üç nedene dayanmaktadır: BPE ve benzeri yöntemlerin İngilizce ağırlıklı derlemlerdeki sıklıklara göre optimize edilmiş olması, Türkçenin eklemeli yapısının ürettiği sözcüklerin biçimbirim sınırlarıyla örtüşmeyen parçalara bölünmesi ve yazı sistemi ile karakter/bayt farklarının aynı sözcüğü farklı maliyetlere taşımasıdır. Daha az token harcanması için gereken ise Türkçeye özgü bir doğal dil işleme sözlüğü bulunduğunda morfoloji temelli tokenleştirme, bu mümkün olmadığında

ise WordPiece'in Türkçenin kök-ek yapısına göre eğitilmesi ya da kök-ek duyarlı model mimarilerinin geliştirilmesidir.

BPE (GPT-4o) gibi standart olarak İngilizce üzerine kurgulanmış modellerin Türkçede dil yapısı gereği dezavantajlı konumda kalması, tokenleştirmenin İngilizce merkezli bir döviz kuru sistemine benzediğini göstermektedir. Ancak bu maliyet yalnızca işlem ücretiyle sınırlı değildir. Aynı içeriğin daha fazla token gerektirmesi, sabit bir bağlam penceresine daha az Türkçe metnin sığmasına neden olmaktadır. Üstelik sıklığa dayalı bir model olduğu için metnin çıkarım süresinin ve gecikmenin artması da diller arasında bir eşitsizlik anlamına gelmektedir. Nitekim Petrov vd. (2023) bu durumu diller arasında adaletsizlik olarak nitelendirmektedir. Bununla birlikte farklı bir tokenleştirme yöntemi uygulandığında diller farklı sonuçlar vermektedir. Örneğin kelime temelli tokenleştirmenin Türkçe için tercih edilmemesinin nedeni, her kelimeyi kapsayacak bir sözlük kütüphanesinin kurulmasındaki güçlülüdür. Bu nedenle frekans tabanlı tokenleştirmelerde WordPiece, kök-ek temelli tokenleştirmelerde ise morfoloji yöntemi Türkçede öne çıkmaktadır. Var olan dil modeli sözlüğü ile alt-sözcük frekansına dayalı modelin ortak paydada birleştiği mekanizmalar tercih edilmektedir. Türkçe morfoloji temelli çalışmalarda sıklıkla başvurulan Zemberek (Akın ve Akın, 2007), Türk dilleri için geliştirilen ilk ve en yaygın referans gösterilen açık kaynaklı doğal dil işleme kütüphanelerinden biridir. Türkçe için açık kaynaklı sözlüklerin sınırlı kalması, bilgi temelli ekonomiye geç giriş ve yerel girişimcilik desteğinin zayıflığı gibi nedenlere bağlanabilir. Ancak geline nokta küresel sahaya uyum sağlamak ve makine diline göre çözüm üretmek de bu gecikmenin önüne geçebilir.

Türkçe ve Altay dil ailesine mensup diller, eklemeli morfolojileri nedeniyle tokenizasyon süreçlerinde benzer zorluklar yaşamaktadır. Eşref (2019) de çalışmasında eklemeli yapının gereksinimlerini ortaya koymuştur. Eklemeli dillerin tokenleştirme zorluğu dilin karmaşıklığından değil, zengin ek gücüne sahip olmasından ileri gelmektedir. Dil yapıları ile tokenleştirme yöntemleri karşılaştırıldığında, benzer yapıdaki dillerin aynı token kategorisinde uyumlu olacağı düşünülebilir. Ancak makine, dilin yapısından burada ayrılır. Makine yalnızca yazılı dile odaklandığı için dildeki karakter, boşluk, noktalama işareti, büyük-küçük harf farkları aynı ailedeki dilleri çok uzak noktalara taşıyabilir. Örneğin Japonca gibi eklemeli yapıya sahip olan bir dil olmasına rağmen Türkçe ile benzer avantaj durumunda değildir. Japoncanın karakteri ve boşluksuz dili mBERT tabanlı WordPiece yönteminde Türkçeden daha dezavantajlı çalışır. Ancak Türkiye'de olduğu gibi diğer ülkelerde de kendi dillerine göre doğal dil işleme kütüphanesi (örneğin Japonca için MeCab) kurmuştur. Doğal dil işleme kütüphaneleri ile dil yapıları var olan tokenleştirme yöntemlerine göre uyumluluğu hesaplanarak geliştirilebilmektedir. Token üzerine yapılan çalışmaların temel amacı, tokenin davranışını belirlemek ve maliyeti düşürmektir. Bu araştırmanın sonucunda görülmektedir ki Türkçenin daha az token harcaması daha az maliyet demektir. Araştırmada alfabe farklılığına göre durum incelenmiştir ancak bunun için gerekenin bir alfabe değişikliği olmadığı sonucuna ulaşılmıştır. Doğrudan sözcük ya da morfoloji tabanlı bir sözlük oluşturma mümkün olmadığı durumlarda, WordPiece modelinin Türkçenin kök-ek yapısına göre eğitilmesi ve yeni türetilen sözcüklerin de sisteme doğrudan katılabilmesi için modelin frekans gücünden yararlanılması gerekmektedir. Türkçe üretken bir eklemeli dile sahip olduğundan dolayı sabit bir kelime sözlüğü zamanla eskiyecektir. Bu sebeple alt-sözcük ve morfoloji temelli yöntemler yeni kelime biçimlerine dayanıklı kalır. Bu bakımdan WordPiece üzerinden yürütülen modelleme ve mimari çalışmalarının sürdürülebilirliği, diğer yöntemlere kıyasla çok daha güçlüdür. Bu melez tokenleştirme yapısı, morfoloji temelli çözümlemeyi frekans tabanlı bir yedekle birleştiren TurkishTokenizer (Bayram vd., 2025b) gibi güncel çalışmalarda da izlenmektedir. Bununla birlikte araştırmanın bazı sınırlılıkları bulunmaktadır. İlk olarak, doğrudan ölçümler yalnızca OpenAI'nin tokenleştiricileriyle (o200k_base ve cl100k_base)

yapılmıştır. LLaMA, Gemma veya Qwen gibi farklı model ailelerinin tokenleştiricileri kapsam dışı tutulduğundan bulgular bu modellere doğrudan genellenemez. İkincisi, karşılaştırmalar tekil sözcük ve tümce örnekleri üzerinden yürütülmüş olup geniş ve temsil gücü yüksek bir derlem üzerinde sözcük başına ortalama token hesabına dayanmamaktadır. Bu nedenle sonuçlar eğilim düzeyinde yorumlanmalıdır. Üçüncüsü, ölçümler Temmuz 2026 tarihli sürümlere ait olduğundan ve tokenleştiriciler güncellendikçe verilen değerler tarihsel nitelik taşıdığından, bulgular ilgili tarihle birlikte değerlendirilmelidir.

Tüm model ve yeniliklerin ötesinde bu çalışmalar, yaklaşık beş yıllık büyük dil modeli döneminin temellerini oluşturmaktadır. Bu temellerin atılmasında kullanılan veri setleri ve doğal dil işleme sözlükleri, modelin altyapısını belirlemektedir. Neredeyse üç-dört ayda bir güncellenen modeller karşısında Türkçe üzerine kurgulanacak yapay zekâ algoritmalarının zaman tasarrufunu da gözetmesi gerekmektedir. Sonuç olarak bu araştırma, Türkçenin token maliyetinin bir alfabe değil, dil yapısına uygun bir yöntem ve altyapı sorunu olduğunu ortaya koymaktadır. Morfoloji temelli tokenleştirme ile Türkçeye özgü doğal dil işleme kaynaklarının geliştirilmesini, bunun mümkün olmadığı durumlarda ise kök-ek duyarlı WordPiece eğitimini hem maliyet hem de sürdürülebilirlik açısından en gerçekçi yol olarak önermektedir.

KAYNAKÇA

- Akın, A. A., & Akın, M. D. (2007). Türk dilleri için açık kaynaklı doğal dil işleme kütüphanesi: Zemberek. *Elektrik Mühendisliği*, 431, 38-44.
- Aksan, D. (2000). *Her yönüyle dil (Ana çizgileriyle dilbilim)*. Türk Dil Kurumu Yayınları.
- Atlı, M. H. (2015). Almanca çoğul eklerin edinimi semaî midir? *Uluslararası Sosyal Araştırmalar Dergisi*, 8(37), 38-48.
- Bayram, M. A., Fincan, A. A., Gümüş, A. S., Karakaş, S., Diri, B., & Yıldırım, S. (2025a). *Tokenization standards for linguistic integrity: Turkish as a benchmark*. arXiv. <https://doi.org/10.48550/arXiv.2502.07057>
- Bayram, M. A., Fincan, A. A., Gümüş, A. S., Karakaş, S., Diri, B., Yıldırım, S., & Çelik, D. (2025b). *Tokens with meaning: A hybrid tokenization approach for Turkish*. arXiv. <https://doi.org/10.48550/arXiv.2508.14292>
- Bergmann, D. (2025, 18 Ağustos). *What is machine learning?* IBM. <https://www.ibm.com/think/topics/machine-learning>
- Bergmann, D. (t.y.). *What is machine learning?* IBM. 9 Temmuz 2026 tarihinde erişildi. <https://www.ibm.com/think/topics/machine-learning>
- Creswell, J. W., & Creswell, J. D. (2021). *Araştırma tasarımı: Nitel, nicel ve karma yöntem yaklaşımları* (H. Özcan, Çev.). Eğiten Kitap. (Orijinal eser 2018'de yayımlanmıştır)
- Eker, S. (2005). *Çağdaş Türk Dili*. Grafiker Yayınları.
- Eşref, Y. (2019). *Türkçe dizi etiketleme için sinir ağı modelleri* (Yayımlanmamış yüksek lisans tezi). Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
- Gibbs, G. R. (2007). *Analyzing qualitative data*. SAGE Publications. <https://doi.org/10.4135/9781849208574>
- Göksel, A., & Kerslake, C. (2005). *Turkish: A comprehensive grammar*. Routledge.
- Greenberg, J. H. (1960). A quantitative approach to the morphological typology of language. *International Journal of American Linguistics*, 26(3), 178-194.

- Harris, Z. S. (1954). Distributional structure. *WORD*, 10(2-3), 146-162. <https://doi.org/10.1080/00437956.1954.11659520>
- Haslett, D. A. (2025). Tokenization changes meaning in large language models: Evidence from Chinese. *Computational Linguistics*, 51(3), 785-814. https://doi.org/10.1162/coli_a_00557
- Kaya, Y. B., & Tantıuđ, A. C. (2024). Türkçe büyük dil modelleri için tokenleştirme ayrıntı düzeyinin etkisi. *Intelligent Systems with Applications*, 21, Article 200335. <https://doi.org/10.1016/j.iswa.2024.200335>
- Maxwell, J. A. (2005). *Qualitative research design: An interactive approach* (2nd ed.). Sage.
- Merriam, S. B. (2013). *Nitel araştırma: Desen ve uygulama için bir rehber*. Nobel.
- Mielke, S. J., Alyafeai, Z., Salesky, E., Raffel, C., Dey, M., Gallé, M., Raja, A., Si, C., Lee, W. Y., Sagot, B., & Tan, S. (2021). *Between words and characters: A brief history of open-vocabulary modeling and tokenization in NLP*. arXiv. <https://doi.org/10.48550/arXiv.2112.10508>
- Onan, B. (2014). Eklemeli dil yapısının Türkçe öğretiminde oluşturduğu bilişsel (kognitif) zeminler/the cognitive backgrounds that agglutinating language structure Form in Turkish teaching. *Mustafa Kemal Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 6(11), 236-263. <https://izlik.org/JA42LD59TK>
- OpenAI. (2026). *tiktoken* (Sürüm 0.13.0) [Bilgisayar yazılımı]. GitHub. <https://github.com/openai/tiktoken>
- Petrov, A., La Malfa, E., Torr, P. H. S., & Bibi, A. (2023). Language model tokenizers introduce unfairness between languages. *Advances in Neural Information Processing Systems*.
- Sapir, E. (1921). *Language: An Introduction to the Study of Speech*. Harcourt, Brace.
- Schuster, M., & Nakajima, K. (2012). Japanese and Korean voice search. *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5149-5152. <https://doi.org/10.1109/ICASSP.2012.6289079>
- Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1), 1-47. <https://doi.org/10.1145/505282.505283>
- Sennrich, R., Haddow, B., & Birch, A. (2016). Neural machine translation of rare words with subword units. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (Cilt 1, ss. 1715-1725). Association for Computational Linguistics. <https://doi.org/10.18653/v1/P16-1162>
- Sezer, T. (2025). *Dizinlerden birimlere: bilişsel dilbilim çerçevesinde birimlendirici tasarımı* (Yayımlanmamış doktora tezi). Hacettepe Üniversitesi Eğitim Bilimleri Enstitüsü, Ankara.
- Shopen, T. (Ed.). (2007). *Language Typology and Syntactic Description* (2. baskı). Cambridge University Press.
- Stryker, C., & Bergmann, D. (t.y.). *What is a transformer model?* IBM. <https://www.ibm.com/think/topics/transformer-model>
- Toraman, C., Yılmaz, E. H., Şahinuç, F., & Özçelik, O. (2022). Impact of tokenization on language models: An analysis for Turkish. *ACM Transactions on Asian and Low-*

Resource Language Information Processing, 22(4), Article 116.
<https://doi.org/10.1145/3578707>

Turan, S. C. (2025). *Türkçe metinlerde sorguya uygun kısımların token seviyesinde tespiti için geç etkileşimli füzyon modeli* (Yayımlanmamış yüksek lisans tezi). Marmara Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). *Attention is all you need*. arXiv. <https://doi.org/10.48550/arXiv.1706.03762>