VERİ BİLİMİ DERGİSİ
www.dergipark.gov.tr/veri

# Türkçe LLM'lerde Nicemleme Dengelemesi: Doğruluk, Hız ve Bellek Kullanımı Üzerine Bir Değerlendirme

Cengizhan Bayram[1], Ferhat Kürkçüoğlu[1], Cevdet Ahmet Turan[1], Volkan Altıntaş[1]*

[1]*Manisa Celal Bayar Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Manisa*

**Makale Bilgisi**

*Başvuru:*
*06/12/2025*
*Kabul:*
*21/12/2025*

**Özet**

Bu çalışma, Türkçe büyük dil modellerinde nicemleme temelli verimlilik–doğruluk ödünleşimini kapsamlı ve karşılaştırmalı biçimde incelemektedir. Çalışma kapsamında, üç farklı Türkçe büyük dil modeli (Trendyol-LLM-7B, Turkish-LLaMA-8B-Instruct ve Turkcell-LLM-7B) üzerinde hem geleneksel ağırlık-yalnız nicemleme yöntemleri (INT8 ve INT4) hem de literatürde yaygın olarak kullanılan gelişmiş bir 4-bit post-training nicemleme yaklaşımı olan GPTQ-4 değerlendirilmiştir. Nicemlenmemiş BFLOAT16 yapılandırmaları referans alınarak, farklı nicemleme stratejilerinin model doğruluğu, üretim kalitesi ve sistem düzeyi verimlilik üzerindeki etkileri analiz edilmiştir.

Deneysel değerlendirmeler, Türkçe doğal dil işleme görevlerini kapsayan güncel ve çeşitli veri setleri üzerinde gerçekleştirilmiştir. Sınıflandırma performansı; dilbilgisel kabul edilebilirlik (TrCOLA, MCC), duygu analizi (SST-2), metinsel çıkarım (RTE) ve soru-cevap ilişkisi (QNLI) görevleri üzerinden ölçülürken, üretim kalitesi XLSUM_TR özetleme veri seti kullanılarak ROUGE-L metriği ile değerlendirilmiştir. Ayrıca VRAM kullanımı, çıkarım süresi ve çalışma zamanı gibi sistem düzeyi ölçümler raporlanmıştır.

Elde edilen bulgular, nicemlemenin görev türüne ve kullanılan yönteme bağlı olarak farklı etkiler gösterdiğini, ancak genel olarak doğrulukta sınırlı kayıplar karşılığında anlamlı bellek tasarrufu ve dağıtım kolaylığı sağladığını ortaya koymaktadır. INT4 ve GPTQ-4 nicemleme yöntemleri, özellikle kaynak kısıtlı ortamlarda bellek verimliliği ile kabul edilebilir doğruluk arasında dengeli bir çözüm sunarken; INT8 nicemleme bazı sınıflandırma görevlerinde yüksek doğruluğu koruyabilmekte, ancak belirli donanım ve yazılım yapılandırmalarında çıkarım hızı açısından dezavantaj oluşturabilmektedir. GPTQ-4 yaklaşımı ise, düşük bitli temsilin sunduğu bellek kazanımını daha istikrarlı çıkarım performansı ile birleştirerek rekabetçi bir alternatif olarak öne çıkmaktadır.

.***Anahtar Kelimeler:*** *Nicemleme; Büyük Dil Modelleri; Model Verimliliği; Türkçe Doğal Dil İşleme; Düşük Hassasiyetli Çıkarım.*

* İletişim e-posta: volkan.altintas@cbu.edu.tr

# Quantization Compensation in Turkish LLMs: An Evaluation of Accuracy, Speed, and Memory Usage

**Abstract**

This study provides a comprehensive and comparative analysis of quantization-based efficiency–accuracy trade-offs in Turkish large language models (LLMs). In this context, three different Turkish LLMs—Trendyol-LLM-7B, Turkish-LLaMA-8B-Instruct, and Turkcell-LLM-7B—are evaluated using both conventional weight-only quantization methods (INT8 and INT4) and a competitive advanced 4-bit post-training quantization technique, GPTQ-4. Non-quantized BFLOAT16 configurations are adopted as reference baselines to systematically examine the effects of different quantization strategies on model accuracy, generation quality, and system-level efficiency.

The experimental evaluation is conducted on a diverse set of Turkish natural language processing tasks. Classification performance is assessed on grammatical acceptability (TrCOLA, using Matthews Correlation Coefficient), sentiment analysis (SST-2), textual entailment (RTE), and question–answering inference (QNLI), while generation quality is evaluated on the XLSUM_TR abstractive summarization dataset using the ROUGE-L metric. In addition, system-level measurements such as VRAM usage, inference time, and overall execution time are reported to provide a holistic assessment of deployment efficiency.

The results indicate that the impact of quantization varies across task types and methods; however, in most cases, substantial memory savings and improved deployability are achieved with only limited degradation in accuracy. INT4 and GPTQ-4 quantization methods offer a favorable balance between memory efficiency and acceptable performance, particularly in resource-constrained environments. While INT8 quantization preserves high accuracy on certain classification tasks, it may introduce notable inference speed slowdowns under specific hardware and software configurations. In contrast, GPTQ-4 emerges as a competitive alternative by combining low-bit memory efficiency with more stable inference performance.

Overall, the findings demonstrate that quantization strategies constitute an effective and practical optimization approach for deploying Turkish large language models. Through comparative analyses across different model architectures and quantization techniques, this study provides actionable insights into task-aware and method-selective quantization strategies within the Turkish LLM ecosystem, contributing generalizable evidence to future research on efficient large language model deployment.

***Keywords:*** *Quantization; Large Language Models; Model Efficiency; Turkish NLP; Low-Precision Inference.*

## 1 Giriş

Son yıllarda doğal dil işleme (Natural Language Processing, NLP) alanında yaşanan en büyük atılımlardan biri, Büyük Dil Modelleri'nin (Large Language Models, LLM) yükselişi olmuştur. Milyarlarca parametreye sahip bu derin öğrenme modelleri, dil anlama ve üretme gibi karmaşık görevlerde insan performansına yaklaşan sonuçlar elde edebilmekte; bu yönüyle yapay zekâ araştırmalarında yeni bir dönemi temsil etmektedir[1]. Bu gelişmenin teknik temelini Transformer mimarisi oluşturmakta olup; ChatGPT, LLaMA, Mistral ve benzeri modeller bu mimarinin küresel ölçekte yaygınlaşmasını sağlamıştır[2].

Bununla birlikte, bu yüksek başarıların ciddi pratik sınırlamaları da bulunmaktadır. LLM'lerin çalıştırılması, yüksek hesaplama gücü, geniş GPU belleği (VRAM), yüksek enerji tüketimi ve maliyetli donanım altyapısı gerektirmektedir. Özellikle 7 milyar (7B) ve üzeri parametre içeren modellerin hem eğitim hem de çıkarım (inference) aşamaları, sınırlı kaynaklara sahip araştırmacılar ve geliştiriciler için önemli bir engel oluşturmaktadır. Bu nedenle, daha az kaynakla çalışabilen verimli modelleme teknikleri geliştirilmesi giderek daha kritik hâle gelmiştir.

Bu doğrultuda öne çıkan yöntemlerden biri nicemleme (quantization) tekniğidir. Nicemleme, model ağırlıklarının yüksek hassasiyetli (örneğin FP32 veya BF16) biçimlerden düşük bit genişliklerine (örneğin INT8 veya INT4) dönüştürülmesiyle, modelin bellek gereksinimini azaltmayı ve çıkarım hızını artırmayı hedefler. Son

yıllarda GGUF, GPTQ, AWQ ve BitsAndBytes gibi çeşitli nicemleme stratejileri geliştirilmiş; bu yöntemlerin her biri, doğruluk ve verimlilik arasında farklı düzeylerde ödünleşim (trade-off) sunmaktadır.

Literatürde, büyük dil modellerinin düşük bitli biçimlerde verimli biçimde çalıştırılmasına yönelik araştırmalar hızla artmaktadır. Örneğin Lin ve arkadaşları, Activation-aware Weight Quantization (AWQ) yöntemini geliştirerek, yalnızca ağırlıkların değil, aktivasyon çıktılarının duyarlılığını da dikkate almış; böylece 4-bit düzeyinde dahi yüksek doğruluk koruması sağlamıştır [3]. Liu ve arkadaşları, LLM-QAT adını verdikleri quantization-aware training yaklaşımıyla, verisiz distilasyon yoluyla modelin davranış dağılımını korumuş ve özellikle 4-bit düzeyinde belirgin iyileşmeler elde etmiştir [4]. Benzer biçimde SpinQuant, post-training nicemleme sürecinde oluşan sapmaları azaltmak için ağırlık ve aktivasyon matrislerine öğrenilen rotasyonlar uygulayarak 4-bit senaryolarda kaybı minimuma indirmiştir[5].

Zhao vd., tarafından önerilen ATOM yöntemi, 3-bit düzeyinde dahi anlamlı doğruluk koruyarak Multi-Granularity Group-wise Quantization (MG-GQ) ve Two-Stage Knowledge Distillation (TSKD) stratejilerini birleştirmiştir[6]. Dong vd. ise Quality Adaptive Quantization (QAQ) yöntemiyle uzun bağlamlı çıkarımlarda artan KV-cache belleği yükünü 10 kat azaltmıştır[7]. Huang vd. (2024) [7], LLaMA-3 üzerinde yaptıkları ampirik analizde 1–8 bit düzeyindeki nicemleme biçimlerini karşılaştırmış; özellikle 2–4 bit ultra düşük hassasiyetli senaryolarda anlamlı performans düşüşleri gözlemleyerek, düşük bitli yöntemlerin hâlâ geliştirilmeye açık olduğunu vurgulamıştır.

Bunlara ek olarak, Lang ve arkadaşları tarafından yapılan kapsamlı inceleme, farklı nicemleme tekniklerini doğruluk (accuracy, perplexity) ve sistemsel verimlilik (VRAM, model boyutu, hız) açısından karşılaştırmış; özellikle post-training nicemleme yöntemlerinin donanım türüne göre değişen performans profilleri sergilediğini göstermiştir[8]. Roy, 4-bit nicemleme biçimlerinin (nf4, fp4) sıcaklık duyarlılığı bakımından farklılıklar sergilediğini ve quantized modellerin çıkarım hızında hâlâ dezavantajlı olabildiğini belirtmiştir[9]. Dumitru vd. katman-temelli bir yaklaşım geliştirerek, katmanların önem düzeyine göre değişken bit seviyelerinde nicemlenmesinin doğruluk kaybını belirgin biçimde azalttığını göstermiştir[10]. Jin, instruction-tuned LLM'ler üzerinde 4-bit nicemlemenin çoğu görevde doğruluğu koruduğunu ancak bazı durumlarda çıkarım hızında yavaşlamalar oluşturabileceğini ortaya koymuştur[11].

Ayrıca Park vd., Any-Precision LLM adını verdikleri yaklaşımda, farklı bit genişliklerinde nicemlenmiş modellerin aynı bellek alanında eşzamanlı barındırılmasını sağlayarak çoklu model dağıtımında verimlilik kazanımı sağlamıştır[12]. Kurtic vd. (2025) [9] ise FP8, INT8 ve INT4 biçimlerinin doğruluk ve hız dengesini yarım milyonun üzerinde test ile sistematik biçimde karşılaştırmış; INT4'ün beklenenden daha rekabetçi olduğunu ortaya koymuştur[13]. Li vd., tarafından gerçekleştirilen çok yönlü bir inceleme ise, post-training nicemlemenin ağırlıklar, aktivasyonlar ve KV önbellek üzerindeki etkilerini analiz ederek farklı senaryolar için pratik öneriler sunmuştur[14].

Bu çalışmalar, düşük bitli nicemleme tekniklerinin büyük dil modellerinde önemli verimlilik avantajları sunduğunu göstermektedir. Ancak mevcut literatürün büyük bölümü İngilizce gibi yüksek kaynaklı dillere odaklanmış; Türkçe gibi düşük kaynaklı dillerdeki modeller üzerinde yapılan sistematik incelemeler oldukça sınırlı kalmıştır[15]. Türkçe LLM'lerin nicemleme sonrasında doğruluk, bellek kullanımı ve çıkarım performansındaki değişimlerinin yeterince araştırılmamış olması, Türkçe NLP ekosisteminde önemli bir boşluğa işaret etmektedir.

Bu bağlamda sunulan çalışma, Türkçe büyük dil modellerinde nicemleme temelli verimlilik–doğruluk ödünleşimini karşılaştırmalı ve çok boyutlu bir çerçevede incelemektedir. Çalışma kapsamında, farklı kurumlar tarafından geliştirilen ve mimari özellikleri bakımından çeşitlilik gösteren üç Türkçe büyük dil modeli (Trendyol-LLM-7B[16], Turkish-LLaMA-8B-Instruct[17] ve Turkcell-LLM-7B[18]) üzerinde nicemleme stratejilerinin etkileri değerlendirilmiştir. Deneysel analizlerde, BitsAndBytes tabanlı ağırlık-yalnız nicemleme yöntemleri (INT8 ve INT4) ile literatürde yaygın olarak kullanılan gelişmiş bir 4-bit post-training nicemleme yaklaşımı olan GPTQ-4 birlikte ele alınmış; nicemlenmemiş BFLOAT16 yapılandırmaları referans alınarak doğruluk, üretim kalitesi ve sistem verimliliği açısından kapsamlı bir karşılaştırma sunulmuştur.

Çalışmanın temel amacı, Türkçe LLM'lerin sınırlı donanım koşullarında verimli biçimde dağıtılabilmesi için uygun nicemleme stratejilerinin belirlenmesine yönelik sayısal bulgular ve pratik içgörüler sağlamaktır. Bu doğrultuda, farklı model

mimarileri ve nicemleme yöntemleri üzerinde gerçekleştirilen deneyler aracılığıyla, gözlemlenen performans eğilimlerinin model-özel mi yoksa daha genel bir davranışı mı yansıttığı ortaya konulmaktadır. Böylece çalışma, Türkçe LLM ekosisteminde görev-duyarlı ve yöntem-seçimli nicemleme stratejilerinin seçimi ve optimizasyonuna ilişkin literatürdeki sınırlı bilgi birikimini genişletmeyi hedeflemektedir.

Bu çalışma ağırlıklı olarak ağırlık-nicemlemesi (weight-only quantization) yaklaşımlarına odaklanmakla birlikte, önceki çalışmalardan farklı olarak yalnızca temel INT8 ve INT4 yapılandırmalarıyla sınırlı kalmayıp, GPTQ gibi gelişmiş düşük bitli post-training nicemleme tekniklerini de kapsamına dâhil etmektedir. Aktivasyon nicemlemesi ve KV-cache sıkıştırma gibi ek optimizasyon yöntemleri ise, ek hesaplama maliyetleri ve yöntemsel karmaşıklık nedeniyle kapsam dışında bırakılmıştır. Bu tercih, Türkçe büyük dil modellerinin gerçek saha koşullarında karşılaşılan temel gereksinimlere—VRAM tasarrufu, dağıtım kolaylığı ve hafif donanım üzerinde çalışabilirlik—odaklanmak amacıyla yapılmıştır. Bu yönüyle çalışma, Türkçe LLM'lerde yaygın olarak kullanılan nicemleme stratejilerinin doğruluk, hız ve bellek kullanımı üzerindeki etkilerini hem yalın hem de karşılaştırmalı bir perspektiften ele alarak, uygulamaya dönük ve genellenebilir sonuçlar sunmayı amaçlamaktadır.

## 2 Materyal ve metot

Bu çalışmada, Türkçe dilinde geliştirilmiş ve farklı mimari ile eğitim özelliklerine sahip üç açık kaynak büyük dil modeli değerlendirilmiştir: Trendyol-LLM-7B, Turkish-LLaMA-8B-Instruct ve Turkcell-LLM-7B. Bu modeller, Türkçe doğal dil işleme görevlerinde yaygın olarak kullanılan ve farklı kurumsal arka planları temsil eden örnekler olarak seçilmiştir. Nicemleme stratejilerinin etkilerini karşılaştırmak amacıyla, her bir modelin nicemlenmemiş bfloat16 (16-bit) hassasiyetindeki sürümü referans (baseline) yapılandırma olarak kabul edilmiştir. Bu yaklaşım, farklı nicemleme yöntemlerinin doğruluk, üretim kalitesi ve sistem düzeyi verimlilik üzerindeki etkilerinin, model-özel farklılıklardan arındırılmış ve tutarlı bir referans noktası üzerinden değerlendirilmesine olanak sağlamaktadır.

### *2.1 Model özellikleri*

Bu çalışmada değerlendirilen modeller, Türkçe sohbet ve görev tabanlı uygulamalar için eğitilmiş, orta ölçekli (7B–8B parametre aralığında) açık kaynak büyük dil modellerinden oluşmaktadır. Çalışma kapsamında kullanılan modeller şunlardır: Trendyol-LLM-7B-chat, Turkish-LLaMA-8B-Instruct ve Turkcell-LLM-7B. Bu modeller, farklı kurumsal ve akademik kaynaklar tarafından geliştirilmiş olmaları, mimari tercihleri ve eğitim stratejileri bakımından çeşitlilik göstermeleri nedeniyle seçilmiştir.

Trendyol-LLM-7B modeli, Direct Preference Optimization (DPO) yaklaşımıyla eğitilmiş olup Türkçe diyalog ve talimat tabanlı görevlerde optimize edilmiştir. Turkish-LLaMA-8B-Instruct modeli, LLaMA mimarisi temel alınarak Türkçe talimat takip yeteneklerini güçlendirmeye yönelik bir ince ayar sürecinden geçirilmiştir. Turkcell-LLM-7B ise kurumsal ölçekli Türkçe veri kaynakları kullanılarak eğitilmiş, genel amaçlı dil anlama ve üretim görevlerine yönelik bir modeldir. Tüm modeller Hugging Face platformu üzerinden erişilebilir olup, değerlendirme çalışmaları bu modellerin yayımlanan kararlı sürümleri üzerinde gerçekleştirilmiştir.

### *2.2 Değerlendirme ortamı*

Model değerlendirmeleri aşağıda belirtilen donanım ortamında gerçekleştirilmiştir:

- GPU: NVIDIA L4 (Ada Lovelace, 24 GB VRAM)
- CPU: Intel(R) Xeon(R) @ 2.20GHz | RAM: 53.0 GB
- Ortam: Python 3.11.13, Transformers 4.54.0+, PyTorch 2.6.0+ Model, Hugging Face Transformers kütüphanesi aracılığıyla yüklenmiş ve çıkarım işlemleri AutoModelForCausalLM ve AutoTokenizer sınıfları üzerinden gerçekleştirilmiştir.

### *2.3 Değerlendirme veri setleri*

Model performansının doğruluk tabanlı değerlendirmesi, Türkçe doğal dil işleme görevlerini kapsayan ve dil anlama yeteneklerinin farklı boyutlarını ölçmeyi amaçlayan çeşitli veri setleri kullanılarak gerçekleştirilmiştir. Bu veri setleri, sınıflandırma ve çıkarımsal akıl yürütme görevlerinin yanı sıra üretim tabanlı özetleme senaryolarını da içerecek biçimde seçilmiştir.

- TrCOLA (The Corpus of Linguistic Acceptability – Türkçe Versiyon): Bu görev, bir cümlenin dilbilgisel olarak

kabul edilebilir olup olmadığını belirlemeyi hedeflemektedir. Veri kümesi, Türkçeye özgü biçimbilimsel (morfolojik), sözdizimsel (sentaktik) ve anlamsal (semantik) yapıları içeren doğru ve hatalı cümlelerden oluşmaktadır. TrCOLA veri seti sınıf dengesizliği göstermesi nedeniyle, bu görevde model performansı Matthews Correlation Coefficient (MCC) metriği ile değerlendirilmiştir. Bu görev, modelin Türkçe'nin yapısal özelliklerine duyarlılığını ölçmek açısından kritik öneme sahiptir[19].

- SST-2 (Türkçe Duygu Analizi Veri Seti): İkili duygu analizi problemine odaklanan bu görevde, *winvoker/turkish-sentiment-analysis-dataset* veri seti kullanılmıştır. Veri kümesi, Türkçe kullanıcı yorumlarından oluşmakta olup her bir örnek olumlu veya olumsuz olarak etiketlenmiştir. Bu görev, modelin günlük dile özgü duygusal tonu doğru biçimde ayırt edebilme yeteneğini değerlendirmektedir[20].
- RTE (Recognizing Textual Entailment – Türkçe):
  Metinler arası çıkarımsal ilişki belirleme görevinde *boun-tabi/nli_tr* veri seti kullanılmıştır. Bu görevde modelin, bir hipotez cümlesinin verilen bir öncül metinden mantıksal olarak çıkarılıp çıkarılamayacağını belirlemesi beklenmektedir. RTE görevi, modelin anlamsal çıkarım ve mantıksal tutarlılık kurma kapasitesini ölçmektedir[20].
- QNLI (Question-Answering Natural Language Inference – Türkçe): Soru-cevap ilişkisini değerlendirmek amacıyla *google/xquad* veri setinin Türkçe yapılandırması (xquad.tr) kullanılmıştır. Bu görevde model, verilen bir bağlam metninin belirli bir sorunun cevabını içerip içermediğini sınıflandırmaktadır. QNLI görevi, gerçekçi soru-cevap senaryolarına dayalı olması nedeniyle modelin bağlamsal anlama yeteneğini test etmektedir[20].

Sınıflandırma görevlerine ek olarak, modelin metin üretme yeteneği XLSUM_TR veri seti kullanılarak değerlendirilmiştir. XL-Sum, çok dilli ve uzun metinlere dayalı özetleme senaryolarını içeren büyük ölçekli bir veri setidir[21]. Türkçe alt kümesi kullanılarak gerçekleştirilen bu değerlendirmede, üretim kalitesi ROUGE-L metriği ile ölçülmüştür[22]. Bu görev, nicemlemenin uzun bağlamlı metin üretimi üzerindeki etkilerini incelemek amacıyla dâhil edilmiştir.

### *2.4 Kuantizasyon yöntemi*

Bu çalışmada, farklı nicemleme stratejilerinin Türkçe büyük dil modelleri üzerindeki etkilerini karşılaştırmalı olarak incelemek amacıyla hem anlık (on-the-fly) hem de post-training nicemleme yaklaşımları kullanılmıştır. Anlık nicemleme işlemleri, *bitsandbytes* kütüphanesi aracılığıyla gerçekleştirilen ağırlık-yalnız (weight-only) nicemleme yöntemlerini kapsamaktadır. Bu yaklaşımda, model ağırlıkları GPU belleğinde (VRAM) INT8 veya INT4 formatında saklanarak bellek gereksinimi azaltılmakta; matris çarpımı gibi hesaplama işlemleri sırasında ise bu düşük hassasiyetli ağırlıklar geçici olarak BFLOAT16 hassasiyetine de-kuantize edilmektedir. Bu yöntem, bellek tasarrufu sağlarken sayısal kararlılığın korunmasını hedeflemektedir.

Buna ek olarak, çalışmada gelişmiş bir 4-bit post-training nicemleme yöntemi olan GPTQ-4[23] de değerlendirmeye dâhil edilmiştir. GPTQ[24] yaklaşımı, model ağırlıklarını çevrimdışı olarak nicemleyerek, grup tabanlı hata minimizasyonu yoluyla düşük bitli temsillerde doğruluk kaybını azaltmayı amaçlamaktadır. GPTQ-4 ile nicemlenmiş modeller, çıkarım sırasında ek bir anlık de-kuantizasyon adımı gerektirmemekte ve bu yönüyle farklı bir performans profili sunmaktadır. Bu sayede, INT4 tabanlı anlık nicemleme ile GPTQ-4 arasında bellek kullanımı, doğruluk ve çıkarım performansı açısından doğrudan bir karşılaştırma yapılabilmiştir.

Bu çalışma kapsamında aktivasyon nicemlemesi veya KV-cache sıkıştırma gibi ek optimizasyon teknikleri uygulanmamıştır. Bu tercih, Türkçe büyük dil modellerinin gerçek saha koşullarında yaygın olarak karşılaşılan temel gereksinimlere—VRAM tasarrufu, dağıtım kolaylığı ve sınırlı donanım üzerinde çalışabilirlik—odaklanmak amacıyla yapılmıştır. Böylece, farklı ağırlık-nicemleme yaklaşımlarının pratik dağıtım senaryolarındaki etkileri yalın ve karşılaştırılabilir bir çerçevede değerlendirilmiştir.

### *2.5 Değerlendirme süreci*

Model değerlendirmesi, tüm modeller ve nicemleme yapılandırmaları için tutarlı ve tekrarlanabilir bir deneysel protokol izlenerek gerçekleştirilmiştir. Değerlendirme süreci aşağıdaki adımlardan oluşmaktadır:

- Her görev için ilgili Türkçe veri kümesi yüklenmiş ve standart veri bölünmeleri (train/validation/test) korunmuştur.
- Metinler, her modelin kendi tokenizer'ı kullanılarak ön işleme tabi tutulmuştur.
- Çıkarım sürecinde, sonuçların deterministik ve karşılaştırılabilir olması amacıyla greedy search yaklaşımı kullanılmıştır (do_sample=False). Üretilen çıktı uzunluğu tüm deneylerde max_new_tokens=128 ile sınırlandırılmıştır.
- Sınıflandırma görevlerinde model çıktıları gerçek etiketlerle karşılaştırılarak performans hesaplanmıştır. TrCOLA görevinin sınıf dengesizliği göstermesi nedeniyle bu görevde Matthews Correlation Coefficient (MCC) [25]metriği kullanılmış; SST-2, RTE ve QNLI görevlerinde ise accuracy metriği raporlanmıştır.
- Üretim tabanlı özetleme görevinde (XLSUM_TR), model çıktıları referans özetlerle karşılaştırılarak ROUGE-L metriği üzerinden değerlendirme yapılmıştır.
- Nicemleme yapılandırmaları arasında sistem düzeyi karşılaştırma yapabilmek amacıyla VRAM kullanımı, çıkarım süresi ve toplam çalışma zamanı gibi ölçümler her deney için kaydedilmiştir.

Bu çalışmada, her bir Türkçe büyük dil modeli için nicemlenmemiş BFLOAT16 yapılandırması referans (baseline) olarak alınmış; INT8, INT4 ve GPTQ-4 nicemleme yöntemleri aynı görev setleri ve aynı çıkarım ayarları altında değerlendirilmiştir. Bu yaklaşım, gözlemlenen performans farklılıklarının nicemleme stratejilerinden kaynaklandığının tutarlı biçimde analiz edilmesine olanak sağlamaktadır.

## 3 Bulgular

Bu bölümde, üç farklı Türkçe büyük dil modelinin—Trendyol-LLM-7B, Turkish-LLaMA-8B-Instruct ve Turkcell-LLM-7B—farklı nicemleme yapılandırmaları altındaki performans sonuçları raporlanmaktadır. Değerlendirmeler, nicemlenmemiş BFLOAT16 yapılandırmaları referans alınarak, INT8, INT4 ve GPTQ-4 nicemleme yöntemlerinin doğruluk, üretim kalitesi ve sistem düzeyi verimlilik üzerindeki etkilerini karşılaştırmalı biçimde incelemektedir. Böylece nicemleme sonrası gözlemlenen performans eğilimlerinin model-özel mi yoksa Türkçe büyük dil modelleri genelinde tutarlı bir davranış mı sergilediği ortaya konulmaktadır.

Deneysel analizler, Türkçe sınıflandırma ve çıkarım görevlerini kapsayan TrCOLA, SST-2, RTE ve QNLI veri setleri ile üretim tabanlı özetleme görevi olan XLSUM_TR üzerinde gerçekleştirilmiştir. Sınıflandırma görevlerinde doğruluk ve görev-özel metrikler (TrCOLA için MCC) kullanılırken, üretim kalitesi ROUGE-L metriği ile değerlendirilmiştir. Sistem düzeyi göstergeler olarak VRAM kullanımı, çıkarım süresi ve toplam çalışma zamanı ölçülmüş ve raporlanmıştır. Tüm modeller ve nicemleme yapılandırmaları, aynı görev setleri ve aynı çıkarım ayarları altında değerlendirilerek sonuçların doğrudan karşılaştırılabilir olması sağlanmıştır.

Tablo 1. Çoklu Türkçe LLM'lerde nicemleme sonrası sınıflandırma performansı

| Model | Nicemleme | TrCOLA (MCC) | SST-2 (Acc) | RTE (Acc) | QNLI (Acc) |
|---|---|---|---|---|---|
| **Trendyol-LLM-7B** | BFLOAT16 | 0.1272 | 0.9014 | 0.7351 | 0.6353 |
| | INT4 | 0.1209 | 0.8956 | 0.7661 | 0.5872 |
| | GPTQ-4 | 0.1057 | 0.8991 | 0.7603 | 0.6204 |
| **Turkish-LLaMA-8B-Instruct** | BFLOAT16 | 0.2093 | 0.8888 | 0.8532 | 0.7683 |
| | INT4 | 0.1372 | 0.8911 | 0.8498 | 0.7672 |
| | GPTQ-4 | 0.1401 | 0.9197 | 0.8681 | 0.7408 |
| **Turkcell-LLM-7B** | BFLOAT16 | 0.1245 | 0.8865 | 0.8096 | 0.6112 |
| | INT4 | 0.0932 | 0.8899 | 0.7798 | 0.6067 |
| | GPTQ-4 | 0.0805 | 0.9071 | 0.7420 | 0.6055 |

### *3.1 Çoklu model süreçleri*

Bu bölümde, Trendyol-LLM-7B, Turkish-LLaMA-8B-Instruct ve Turkcell-LLM-7B modellerinin farklı nicemleme yapılandırmaları altındaki sınıflandırma performansları karşılaştırmalı olarak sunulmaktadır. Tüm modeller için nicemlenmemiş BFLOAT16 yapılandırmaları referans (baseline) olarak alınmış; INT4 ve GPTQ-4 nicemleme yöntemleri aynı görev setleri ve aynı çıkarım ayarları altında değerlendirilmiştir. Elde edilen sonuçlar Tablo 1'de özetlenmektedir.

### *3.2 Sistem düzeyi karşılaştırma*

Nicemleme yöntemlerinin pratik dağıtım senaryolarındaki etkilerini incelemek amacıyla, üç farklı Türkçe büyük dil modeli üzerinde sistem düzeyi metrikler karşılaştırmalı olarak değerlendirilmiştir. Model boyutu, GPU bellek kullanımı (VRAM) ve çıkarım hızı (tokens/s) ölçümleri tüm modeller için aynı donanım ve çıkarım ayarları altında gerçekleştirilmiş olup, elde edilen sonuçlar Tablo 2'de sunulmaktadır.

Tablo 2 incelendiğinde, üç modelde de nicemleme sonrasında VRAM kullanımında belirgin ve tutarlı bir azalma sağlandığı görülmektedir. BFLOAT16 yapılandırmalarına kıyasla INT8 nicemleme yaklaşık %30.2 – %32.9 aralığında, INT4 nicemleme ise %61.6 – %64.4 aralığında VRAM tasarrufu sağlamıştır. Bu durum, bellek kazanımının modelden bağımsız bir nicemleme davranışı sergilediğini göstermektedir.

Çıkarım hızı metrikleri incelendiğinde ise tüm modellerde benzer bir eğilim gözlemlenmiştir. BFLOAT16 yapılandırmaları en yüksek çıkarım hızlarını üretirken, INT8 nicemleme yapılandırmaları belirgin bir hız düşüşü sergilemiştir. Buna karşılık INT4 nicemleme, BFLOAT16'a görece daha yakın çıkarım hızları sunarak hız–bellek dengesi açısından daha avantajlı bir profil ortaya koymuştur. Bu davranış, INT8 nicemlemede uygulanan anlık de-kuantizasyon ve veri türü dönüşümlerinin, hesaplama hattında ek gecikmelere yol açmasından kaynaklanmaktadır.

Tablo 2. Sistem düzeyi metrikler

| Model | Nicemleme | Model Boyutu (GB) | VRAM Kullanımı (GB) | Çıkarım Hızı (tokens/s) |
|---|---|---|---|---|
| **Trendyol-LLM-7B** | BFLOAT16 | 25.48 | 17.61 | 17.42 |
| | INT8 | 25.48 | 11.82 | 7.80 |
| | INT4 | 25.48 | 6.52 | 14.21 |
| **Turkish-LLaMA-8B-Instruct** | BFLOAT16 | 14.97 | 20.56 | 15.62 |
| | INT8 | 14.97 | 14.36 | 6.46 |
| | INT4 | 14.97 | 7.31 | 13.58 |
| **Turkcell-LLM-7B** | BFLOAT16 | 13.74 | 18.25 | 16.29 |
| | INT8 | 13.74 | 12.37 | 5.23 |
| | INT4 | 13.74 | 7.01 | 14.07 |

Basit bir verimlilik göstergesi olarak tokens/s başına VRAM oranı dikkate alındığında, INT4 yapılandırmalarının üç modelde de en yüksek sistem verimliliğini sunduğu görülmektedir. Bu bulgular, düşük bitli nicemleme stratejilerinin yalnızca bellek kazanımı değil, aynı zamanda pratik çıkarım performansı açısından da modelden bağımsız olarak tutarlı sonuçlar ürettiğini göstermektedir.

### *3.3 GPTQ-4 ve INT4 nicemleme yaklaşımlarının karşılaştırılması*

INT4 tabanlı anlık nicemleme ile GPTQ-4 post-training nicemleme yaklaşımlarının karşılaştırmalı sonuçları Tablo 3'te sunulmaktadır. Karşılaştırma, üç farklı Türkçe büyük dil modeli üzerinde elde edilen sonuçların ortalamaları üzerinden gerçekleştirilmiştir.

Tablo 3 incelendiğinde, her iki 4-bit yaklaşımın da bellek gereksinimlerini benzer düzeylerde azalttığı görülmektedir. Ortalama VRAM kullanımı INT4 yapılandırmasında 7.23 GB, GPTQ-4 yapılandırmasında ise 7.01 GB olarak ölçülmüş; bellek kazanımı açısından yöntemler arasında belirgin bir fark gözlemlenmemiştir. Çıkarım hızı açısından bakıldığında, INT4 yapılandırması ortalama 12.63 tokens/s ile GPTQ-4'ün (12.11 tokens/s) hafif üzerinde yer almakta; ancak iki yaklaşım arasındaki fark sınırlı kalmaktadır.

Tablo 3. INT4 ve GPTQ-4 nicemleme yaklaşımlarının karşılaştırılması

| Metrik | INT4 | GPTQ-4 |
|---|---|---|
| **Ortalama VRAM Kullanımı (GB)** | 7.23 | 7.01 |
| **Ortalama Çıkarım Hızı (tokens/s)** | 12.63 | 12.11 |

| | | |
|---|---|---|
| **TrCOLA – MCC (ortalama)** | 0.117 | 0.109 |
| **SST-2 – Accuracy (ortalama)** | 0.892 | 0.909 |
| **RTE – Accuracy (ortalama)** | 0.799 | 0.790 |
| **QNLI – Accuracy (ortalama)** | 0.654 | 0.656 |
| **XLSUM_TR – ROUGE-L (ortalama)** | 0.161 | 0.160 |

Sınıflandırma görevleri açısından değerlendirildiğinde, GPTQ-4 yapılandırmalarının özellikle SST-2 ve QNLI görevlerinde INT4'e kıyasla daha tutarlı veya hafif üstün sonuçlar ürettiği görülmektedir. TrCOLA ve RTE görevlerinde ise her iki yöntemin performansları benzer bantlarda seyretmiş ve farklar ihmal edilebilir düzeyde kalmıştır. Üretim tabanlı XLSUM_TR özetleme görevinde ölçülen ROUGE-L skorları da her iki yöntemin üretim kalitesi açısından büyük ölçüde eşdeğer olduğunu göstermektedir.

Bu bulgular, INT4 nicemlemenin hızlı uygulanabilirliği ve yüksek çıkarım hızı avantajına sahip olduğunu; GPTQ-4'ün ise modelden bağımsız olarak daha istikrarlı ve dengeli bir performans profili sunduğunu ortaya koymaktadır. Dolayısıyla, INT4 yaklaşımı deneysel ve prototipleme odaklı senaryolar için uygun bir seçenek sunarken, GPTQ-4 nicemleme yöntemi üretim ortamlarında daha güvenilir bir alternatif olarak değerlendirilebilir.

## 4 Sonuçlar

Bu çalışmada, farklı mimari ve eğitim özelliklerine sahip üç Türkçe büyük dil modeli (Trendyol-LLM-7B, Turkish-LLaMA-8B-Instruct ve Turkcell-LLM-7B) üzerinde uygulanan nicemleme stratejilerinin doğruluk, üretim kalitesi ve sistem düzeyi verimlilik üzerindeki etkileri kapsamlı ve karşılaştırmalı biçimde incelenmiştir. Elde edilen bulgular, nicemlemenin Türkçe LLM'ler için tek boyutlu bir optimizasyon problemi olmadığını; görev türüne, nicemleme yöntemine ve sistem kısıtlarına bağlı olarak çok boyutlu bir ödünleşim alanı oluşturduğunu açık biçimde ortaya koymaktadır.

Çoklu model sonuçları birlikte değerlendirildiğinde, düşük bitli nicemleme yöntemlerinin genel doğruluk performansını belirgin biçimde düşürmeden anlamlı bellek tasarrufu sağladığı görülmüştür. Sınıflandırma görevlerinde BFLOAT16 yapılandırmaları güçlü bir referans oluşturmakla birlikte, INT4 ve GPTQ-4 nicemleme yöntemlerinin çoğu görevde bu değerlere oldukça yakın performans sergilediği gözlemlenmiştir. Bu durum, Türkçe büyük dil modellerinde yüksek seviyeli anlamsal temsillerin düşük bitli ağırlık temsilleri altında da büyük ölçüde korunabildiğini göstermektedir.

Görev bazında yapılan analizler, nicemlemenin etkisinin homojen olmadığını ortaya koymuştur. Dilbilgisel kabul edilebilirlik gibi yapısal özelliklerin ön planda olduğu TrCOLA görevinde, bazı modellerde düşük bitli nicemleme yapılandırmalarının BFLOAT16 referansına yakın veya yer yer daha yüksek performans sergilemesi, nicemlemenin belirli durumlarda düzenlileştirici (regularization) bir etki yaratabileceğine işaret etmektedir. Buna karşılık, duygu analizi gibi anlamsal yoğunluğu yüksek görevlerde BFLOAT16 yapılandırmalarının genellikle en yüksek doğruluğu sunduğu; özellikle INT4 nicemlemede bazı modellerde sınırlı fakat tutarlı doğruluk kayıplarının ortaya çıktığı gözlemlenmiştir. Bu bulgu, agresif bit indiriminin ince semantik ayrımların temsilini kısmen zorlaştırabileceğini göstermektedir.

Metinsel çıkarım ve soru-cevap ilişkisi gerektiren RTE ve QNLI görevlerinde ise nicemlemenin etkisi tüm modellerde oldukça sınırlı kalmıştır. Düşük bitli yapılandırmaların BFLOAT16 referansına çok yakın doğruluk değerleri üretmesi, bağlamsal ilişki kurma ve çıkarımsal akıl yürütme gibi yüksek seviyeli dil yeteneklerinin nicemlemeye karşı daha dayanıklı olduğunu göstermektedir. Bu sonuç, Türkçe LLM'lerde anlamsal çekirdek temsillerin nicemleme altında kararlılığını koruyabildiğine dair güçlü bir deneysel kanıt sunmaktadır.

Üretim tabanlı özetleme görevlerinde elde edilen sonuçlar da benzer bir dayanıklılığa işaret etmektedir. XLSUM_TR veri seti üzerinde ölçülen ROUGE-L skorları, INT4 ve GPTQ-4 yapılandırmalarının BFLOAT16 referansına oldukça yakın çıktılar üretebildiğini göstermiştir. Özellikle GPTQ-4 nicemlemenin farklı modellerde daha istikrarlı bir üretim kalitesi sunduğu; düşük bitli temsillere rağmen anlamsal bütünlüğün ve özet tutarlılığının büyük ölçüde korunduğu gözlemlenmiştir. Bu durum, gelişmiş post-training

nicemleme yöntemlerinin uzun bağlamlı metin üretiminde pratik olarak uygulanabilir olduğunu göstermektedir.

Sistem düzeyi metrikler açısından değerlendirildiğinde, nicemlemenin en belirgin kazanımının bellek kullanımı üzerinde olduğu açıkça görülmektedir. INT4 ve GPTQ-4 yapılandırmaları, tüm modellerde yaklaşık %50'ye varan VRAM tasarrufu sağlamış; bu durum Türkçe LLM'lerin sınırlı donanım koşullarında dağıtılabilirliğini önemli ölçüde artırmıştır. Buna karşılık çıkarım hızı açısından farklı nicemleme yöntemleri arasında belirgin ayrışmalar gözlemlenmiştir. INT8 nicemlemenin, anlık de-kuantizasyon ve veri türü dönüşümlerine bağlı olarak tüm modellerde tutarlı biçimde hız düşüşü sergilemesi, bellek kazanımının her zaman performans kazanımı anlamına gelmediğini ortaya koymuştur. INT4 yapılandırmaları ise hız–bellek dengesi açısından daha avantajlı bir profil sunmuştur.

INT4 anlık nicemleme ile GPTQ-4 karşılaştırması, düşük bitli nicemleme stratejilerinin seçiminde yalnızca bellek kazanımının değil, çıkarım istikrarının da kritik olduğunu göstermektedir. INT4 yaklaşımı hızlı uygulanabilirliği ve düşük mühendislik maliyeti ile deneysel ve prototipleme senaryoları için uygun bir seçenek sunarken; GPTQ-4, farklı mimarilere sahip modellerde daha dengeli doğruluk ve sistem davranışı sergileyerek üretim ortamları için daha güvenilir bir alternatif olarak öne çıkmaktadır. Bu ayrım, nicemleme stratejisinin kullanım senaryosuna göre seçilmesi gerektiğini açık biçimde ortaya koymaktadır.

Bu çalışmada elde edilen bulgular, INT8 nicemlemenin Türkçe büyük dil modelleri için her koşulda avantajlı bir çözüm sunmadığını açık biçimde ortaya koymaktadır. Her ne kadar INT8 yapılandırmaları bellek kullanımında anlamlı bir azalma sağlasa da, sistem düzeyi ölçümler tüm modellerde INT8 konfigürasyonlarının çıkarım hızı açısından tutarlı biçimde dezavantajlı olduğunu göstermiştir. Bu durum, INT8 nicemlemede uygulanan anlık de-kuantizasyon işlemleri ve veri türü dönüşümlerinin GPU hesaplama hattında ek gecikmelere yol açmasından kaynaklanmaktadır. Özellikle BFLOAT16 ile optimize edilmiş modern GPU mimarilerinde, INT8 nicemlemenin beklenen hız kazanımını sağlayamaması, bellek tasarrufu ile hesaplama verimliliği arasındaki ilişkinin doğrusal olmadığını göstermektedir.

Bu bulgunun yalnızca tek bir modele özgü olmadığı, üç farklı Türkçe büyük dil modelinde de benzer biçimde tekrarlandığı gözlemlenmiştir. Dolayısıyla INT8 nicemlemede gözlenen hız düşüşü, deneysel bir sapma ya da model-özel bir durumdan ziyade, kullanılan nicemleme mekanizmasının yapısal bir sonucu olarak değerlendirilmelidir. Bu bağlamda INT8, çalışmada "en iyi" nicemleme çözümü olarak değil; daha agresif ve dengeli düşük bitli yöntemlere (INT4 ve GPTQ-4) duyulan ihtiyacı gerekçelendiren bir ara basamak olarak konumlandırılmıştır. Elde edilen sonuçlar, pratik dağıtım senaryolarında bellek kazanımının tek başına yeterli olmadığını; çıkarım kararlılığı ve hızın da nicemleme stratejisi seçiminde belirleyici faktörler olduğunu göstermektedir.

Türkçenin eklemeli morfolojik yapısı dikkate alındığında, elde edilen bulgular dil-özel açıdan da önemli çıkarımlar sunmaktadır. Uzun ek dizilimleri ve kelime-altı temsillere yüksek bağımlılık, düşük bitli nicemlemede uç değerlerin ve ince istatistiksel varyansların korunmasını kritik hâle getirmektedir. Bu bağlamda, Türkçeye özgü morfolojik karmaşıklığın nicemleme ile etkileşimi, görev bazlı performans farklılıklarının temel belirleyicilerinden biri olarak değerlendirilebilir.

Sonuç olarak bu çalışma, Türkçe büyük dil modellerinde nicemlemenin etkilerini çoklu model, çoklu görev ve çoklu nicemleme yöntemi perspektifinden bütüncül biçimde ortaya koymaktadır. Bulgular, düşük bitli nicemleme stratejilerinin Türkçe LLM'lerin pratik dağıtımı açısından uygulanabilir ve etkili bir optimizasyon aracı olduğunu; ancak en uygun yapılandırmanın görev türü, sistem kısıtları ve kullanım senaryosuna bağlı olarak değiştiğini göstermektedir. Gelecek çalışmalarda, activation-aware quantization (AWQ), grup tabanlı GPTQ varyantları ve görev-duyarlı nicemleme stratejilerinin Türkçe gibi morfolojik olarak zengin diller üzerindeki etkilerinin daha ayrıntılı biçimde incelenmesi, doğruluk–verimlilik dengesinin daha da iyileştirilmesine yönelik önemli katkılar sağlayacaktır.

## Kaynaklar

[1] T. Brown *vd.*, "Language Models are Few-Shot Learners", içinde *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2020, ss. 1877-1901. Erişim: 27 Temmuz 2025. [Çevrimiçi]. Erişim adresi: https://proceedings.neurips.cc/paper_files/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html

[2] A. Vaswani *vd.*, "Attention is All you Need", içinde *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2017. Erişim: 27 Temmuz 2025. [Çevrimiçi]. Erişim adresi: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[3] J. Lin, J. Tang, H. Tang, S. Yang, G. Xiao, ve S. Han, "AWQ: Activation-aware Weight Quantization for On-Device LLM Compression and Acceleration", *GetMobile: Mobile Comp. and Comm.*, c. 28, sy 4, ss. 12-17, Oca. 2025, doi: 10.1145/3714983.3714987.

[4] Z. Liu *vd.*, "LLM-QAT: Data-Free Quantization Aware Training for Large Language Models", 29 Mayıs 2023, *arXiv*: arXiv:2305.17888. doi: 10.48550/arXiv.2305.17888.

[5] Z. Liu *vd.*, "SpinQuant: LLM quantization with learned rotations", 20 Şubat 2025, *arXiv*: arXiv:2405.16406. doi: 10.48550/arXiv.2405.16406.

[6] Y. Zhao *vd.*, "Atom: Low-Bit Quantization for Efficient and Accurate LLM Serving", *Proceedings of Machine Learning and Systems*, c. 6, ss. 196-209, May. 2024.

[7] S. Dong, W. Cheng, J. Qin, ve W. Wang, "QAQ: Quality Adaptive Quantization for LLM KV Cache", 12 Nisan 2024, *arXiv*: arXiv:2403.04643. doi: 10.48550/arXiv.2403.04643.

[8] J. Lang, Z. Guo, ve S. Huang, "A Comprehensive Study on Quantization Techniques for Large Language Models", 30 Ekim 2024, *arXiv*: arXiv:2411.02530. doi: 10.48550/arXiv.2411.02530.

[9] S. Roy, "Understanding the Impact of Post-Training Quantization on Large Language Models", 17 Eylül 2023, *arXiv*: arXiv:2309.05210. doi: 10.48550/arXiv.2309.05210.

[10] R.-G. Dumitru, V. Yadav, R. Maheshwary, P. I. Clotan, S. T. Madhusudhan, ve M. Surdeanu, "Variable Layerwise Quantization: A Simple and Effective Approach to Quantize LLMs", içinde *Findings of the Association for Computational Linguistics: ACL 2025*, W. Che, J. Nabende, E. Shutova, ve M. T. Pilehvar, Ed., Vienna, Austria: Association for Computational Linguistics, Tem. 2025, ss. 534-550. Erişim: 27 Temmuz 2025. [Çevrimiçi]. Erişim adresi: https://aclanthology.org/2025.findings-acl.29/

[11] R. Jin *vd.*, "A Comprehensive Evaluation of Quantization Strategies for Large Language Models", içinde *Findings of the Association for Computational Linguistics: ACL 2024*, L.-W. Ku, A. Martins, ve V. Srikumar, Ed., Bangkok, Thailand: Association for Computational Linguistics, Ağu. 2024, ss. 12186-12215. doi: 10.18653/v1/2024.findings-acl.726.

[12] Y. Park, J. Hyun, S. Cho, B. Sim, ve J. W. Lee, "Any-Precision LLM: Low-Cost Deployment of Multiple, Different-Sized LLMs", 21 Haziran 2024, *arXiv*: arXiv:2402.10517. doi: 10.48550/arXiv.2402.10517.

[13] S. Li *vd.*, "Evaluating Quantized Large Language Models", 06 Haziran 2024, *arXiv*: arXiv:2402.18158. doi: 10.48550/arXiv.2402.18158.

[14] E. Kurtic, A. Marques, S. Pandit, M. Kurtz, ve D. Alistarh, "'Give Me BF16 or Give Me Death'? Accuracy-Performance Trade-Offs in LLM Quantization", 30 Mayıs 2025, *arXiv*: arXiv:2411.02355. doi: 10.48550/arXiv.2411.02355.

[15] Ç. Çöltekin, "A Corpus of Turkish Offensive Language on Social Media", içinde *Proceedings of the Twelfth Language Resources and Evaluation Conference*, N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, ve S. Piperidis, Ed., Marseille, France: European Language Resources Association, May. 2020, ss. 6174-6184. Erişim: 27 Temmuz 2025. [Çevrimiçi]. Erişim adresi: https://aclanthology.org/2020.lrec-1.758/

[16] "Trendyol/Trendyol-LLM-7b-chat-dpo-v1.0 · Hugging Face". Erişim: 06 Aralık 2025. [Çevrimiçi]. Erişim adresi: https://huggingface.co/Trendyol/Trendyol-LLM-7b-chat-dpo-v1.0

[17] H. T. Kesgin *vd.*, "Optimizing Large Language Models for Turkish: New Methodologies in Corpus Selection and Training", içinde *2024 Innovations in Intelligent Systems and Applications Conference (ASYU)*, Eki. 2024, ss. 1-6. doi: 10.1109/ASYU62119.2024.10757019.

[18] "TURKCELL/Turkcell-LLM-7b-v1 · Hugging Face". Erişim: 20 Aralık 2025. [Çevrimiçi]. Erişim adresi: https://huggingface.co/TURKCELL/Turkcell-LLM-7b-v1

[19] A. Warstadt, A. Singh, ve S. R. Bowman, "CoLA: The Corpus of Linguistic Acceptability (with added annotations)". 2019. Erişim: 06 Aralık 2025. [Çevrimiçi]. Erişim adresi: http://archive.nyu.edu/handle/2451/60441

[20] *turkish-nlp-suite/TrGLUE*. (03 Ekim 2025). Python. Turkish NLP Suite. Erişim: 06 Aralık 2025. [Çevrimiçi]. Erişim adresi: https://github.com/turkish-nlp-suite/TrGLUE

[21] T. Hasan *vd.*, "XL-Sum: Large-Scale Multilingual Abstractive Summarization for 44 Languages", içinde *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, C. Zong, F. Xia, W. Li, ve R. Navigli, Ed., Online: Association for Computational Linguistics, Ağu. 2021, ss. 4693-4703. doi: 10.18653/v1/2021.findings-acl.413.

[22] C. Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries", içinde *Text Summarization Branches Out*, Association for Computational Linguistics, Tem. 2004, ss. 74-81. Erişim: 09 Ağustos 2025. [Çevrimiçi]. Erişim adresi: https://aclanthology.org/W04-1013/

[23] Y. Zhang *vd.*, "S4T-GPTQ: A Space-for-Time Strategy For Optimizing GPTQ 4-bit Quantization", May. 2025, ss. 383-387. doi: 10.1109/ICMLT65785.2025.11193187.

[24] E. Frantar, S. Ashkboos, T. Hoefler, ve D. Alistarh, "GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers", 22 Mart 2023, *arXiv*: arXiv:2210.17323. doi: 10.48550/arXiv.2210.17323.

[25] J. Tamura, Y. Itaya, K. Hayashi, ve K. Yamamoto, "Statistical Inference of the Matthews Correlation Coefficient for Multiclass Classification", 09 Mart 2025, *arXiv*: arXiv:2503.06450. doi: 10.48550/arXiv.2503.06450.