

**TÜRKİYE CUMHURİYETİ**  
**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



**CEVAPTAN SORUYU ÜRETMEK**

21011036 – TALHA ÇELİK  
20011015 – HÜSEYİN SAİD ARICI

**BİLGİSAYAR PROJESİ**

Danışman  
Himmet Toprak KESGİN

Ocak, 2025



## TEŞEKKÜR

---

Proje boyunca yaptığımız toplantılarda her daim sorularımızı cevaplayan ve bize yol gösteren sayın Himmet Toprak Kesgin hocamıza teşekkür etmek istiyoruz.

TALHA ÇELİK  
HÜSEYİN SAİD ARICI

# İÇİNDEKİLER

---

|  |             |
|--|-------------|
| <b>KISALTMA LİSTESİ</b>                                    | <b>v</b>    |
| <b>ŞEKİL LİSTESİ</b>                                       | <b>vi</b>   |
| <b>TABLO LİSTESİ</b>                                       | <b>vii</b>  |
| <b>ÖZET</b>  | <b>viii</b> |
| <b>ABSTRACT</b>  | <b>ix</b>   |
| <b>1 GİRİŞ</b>   | <b>1</b>    |
| 1.1 Amaç . . . . .   | 1           |
| 1.2 Ön İnceleme . . . . .                                  | 2           |
| <b>2 LİTERATÜR ANALİZİ</b>                                 | <b>3</b>    |
| <b>3 SİSTEM ANALİZİ ve FİZİBİLİTE</b>                      | <b>6</b>    |
| 3.1 Sistem Analizi . . . . .                               | 6           |
| 3.2 Fizibilite . . . . .                                   | 7           |
| 3.2.1 Teknik Fizibilite . . . . .                          | 7           |
| 3.2.2 Yasal Fizibilite . . . . .                           | 8           |
| 3.2.3 Ekonomik Fizibilite . . . . .                        | 8           |
| 3.2.4 İş Gücü ve Zaman Fizibilitesi . . . . .              | 8           |
| <b>4 SİSTEM TASARIMI</b>                                   | <b>9</b>    |
| 4.1 Materyal ve Veri Kümesi . . . . .                      | 9           |
| 4.2 Metot . . . . .  | 10          |
| <b>5 DENEYSEL SONUÇLAR</b>                                 | <b>12</b>   |
| 5.1 Performans Analizi . . . . .                           | 12          |
| 5.1.1 Modellerin Aynı Cevaba Ürettikleri Sorular . . . . . | 14          |
| 5.2 Karşılaştırmalı Analiz . . . . .                       | 16          |
| <b>6 SONUÇ VE TARTIŞMA</b>                                 | <b>17</b>   |

|                    |           |
|--------------------|-----------|
| <b>Referanslar</b> | <b>18</b> |
| <b>Özgeçmiş</b>    | <b>20</b> |

## KISALTMA LİSTESİ

---

|       |                                     |
|-------|-------------------------------------|
| NLP   | Natural Language Processing         |
| QG    | Question Generation                 |
| AQG   | Automatic Question Generation       |
| AI    | Artificial Intelligence             |
| SQuAD | Stanford Question Answering Dataset |
| NQG   | Neural Question Generation          |
| RNN   | Recurrent Neural Network            |

## ŞEKİL LİSTESİ

---

|           |                                    |    |
|-----------|------------------------------------|----|
| Şekil 1.1 | mT5'in yapısı . . . . .            | 2  |
| Şekil 3.1 | Model Akış Diyagramı . . . . .     | 7  |
| Şekil 3.2 | Donanım Özellikleri . . . . .      | 7  |
| Şekil 3.3 | Gantt Şeması . . . . .             | 8  |
| Şekil 4.1 | Sistem Çalışma Metotları . . . . . | 10 |
| Şekil 5.1 | Arayüz . . . . .                   | 13 |
| Şekil 5.2 | Gelişmiş Ayarlar . . . . .         | 14 |

## TABLO LİSTESİ

---

|           |   |    |
|-----------|---|----|
| Tablo 5.1 | Model 1 için Training Loss ve Validation Loss Bilgileri . . . . . | 12 |
| Tablo 5.2 | Model 2 için Training Loss ve Validation Loss Bilgileri . . . . . | 13 |
| Tablo 5.3 | Model-1 ve Model-2 için Değerlendirme Metrikleri . . . . .        | 16 |



# CEVAPTAN SORUYU ÜRETMEK

TALHA ÇELİK  
HÜSEYİN SAİD ARICI

Bilgisayar Mühendisliği Bölümü  
Bilgisayar Projesi

Danışman: Himmet Toprak KESGİN

Bu çalışmada, verilen bir cevaptan soruyu üretmek amacıyla bir doğal dil işleme modeli geliştirilmiştir. Bu doğrultuda, modelin giriş verisi olarak metnin tokenlerinin tersine çevrilmiş hali kullanılarak dil modelinin eğitimi gerçekleştirilmiştir. Tersine çevrilmiş token dizileriyle çalışmanın temel amacı, baz alınan gpt modelinde kullanılan önceki token tahmin sistemidir. Eğitim sürecinde farklı öğrenme oranları kullanılarak farklı modeller eğitilmiştir. Eğitim boyunca her iki modelin eğitim ve doğrulama kayıpları takip edilerek performansları değerlendirilmiştir. Model performansını değerlendirmek için metrikler kullanılmıştır. Modelden çıktı alma noktasında farklı parametre seçenekleri sunulmuştur. Bu sayede daha yaratıcı cevaplara ulaşılabilmesi sağlanmıştır. Bu modeller, bir yerel arayüz aracılığıyla kullanıma sunulmuştur.

**Anahtar Kelimeler:** doğal dil işleme, yapay zeka, sorudan cevap üretimi, dil modelleri

## ABSTRACT

---

# GENERATING THE QUESTION FROM THE ANSWER

TALHA ÇELİK  
HÜSEYİN SAİD ARICI

Department of Computer Engineering  
Computer Project

Advisor: Himmet Toprak KESGİN

In this study, a natural language processing (NLP) model was developed with the aim of generating a question based on a given answer. Accordingly, the model was trained using reversed token sequences as input. The primary purpose of working with reversed token sequences is to adopt the previous token prediction system used in the baseline GPT model. Different models were trained using various learning rates during the training process. Throughout the training, the training and validation losses of both models were tracked, and their performance was evaluated. Metrics were used to assess model performance. Additionally, various parameter options were provided for generating outputs from the model, allowing for more creative responses. These models were made available through a local interface for user interaction.

**Keywords:** natural language processing, artificial intelligence, question generation, language models

Son yıllarda, metin üretme odaklı yapay zeka teknolojileri, dijital dünyanın hızla gelişen ihtiyaçlarına yanıt verebilmek ve bilgiye erişimi daha hızlı ve verimli bir hale getirmek amacıyla önemli bir odak noktası haline gelmiştir. Yapay zeka, özellikle NLP ve makine öğrenimi gibi alanlardaki ilerlemeler sayesinde, metin oluşturma süreçlerinde devrim yaratmış ve bu alandaki potansiyelini gözler önüne sermiştir. Artık sadece metinleri anlamakla sınırlı kalmayıp, karmaşık dil yapıları ve bağlam ilişkilerini analiz ederek özgün içerikler oluşturabilen sistemler, içerik üretiminde hem bireylerin hem de kurumların iş yükünü hafifletmektedir.

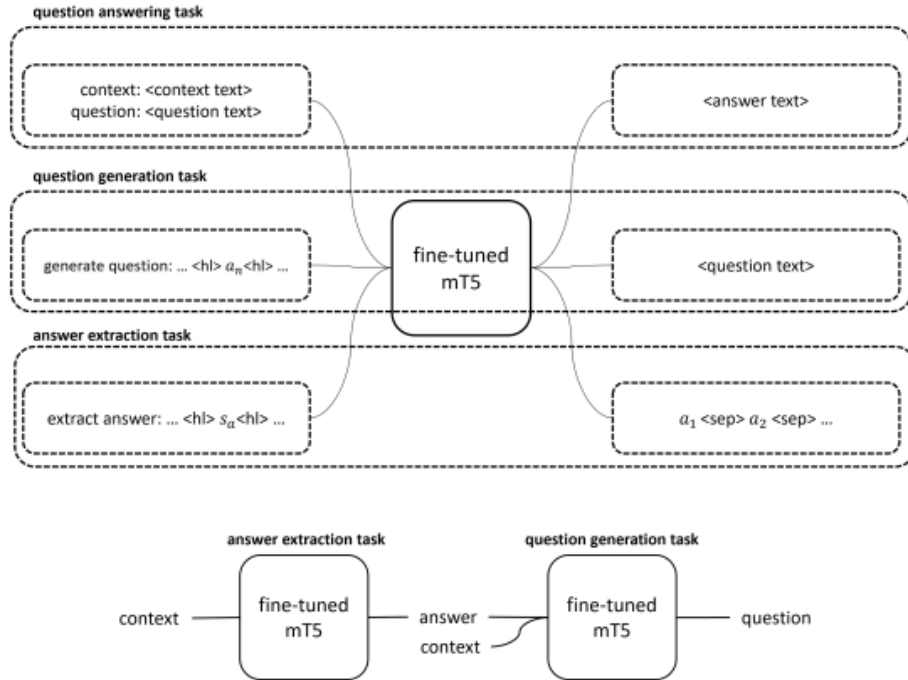
Yapay zeka destekli metin üretim teknolojileri, metin tabanlı veri analizinden müşteri hizmetlerine, pazarlamadan eğitime kadar geniş bir yelpazede uygulama alanı bulmaktadır. Bu teknolojiler, özellikle içerik üretiminin hızla tüketildiği sosyal medya, dijital pazarlama ve haber siteleri gibi platformlarda ve akademik kurumlarda; neredeyse sınırsız sayıda ve çeşitlilikte içerik üretimine olanak tanıyarak büyük bir avantaj sağlamaktadır. Örneğin, E-öğrenme endüstrisine göre öğrenme yönetimi araçlarının yüzde 47'si önümüzdeki üç yıl içinde yapay zekâ becerileriyle çalıştırılacaktır [1].

## 1.1 Amaç

Projenin amacı, mevcut doğal dil işleme tekniklerini kullanarak, bir modelin verilen cevaptan soruyu üretebilmesini sağlamaktır. Bu model, örnek olarak GPT benzeri bir modelin ters işleyişini kullanarak bir sorunun cevabı ile o soruyu üretecek, böylece AI temelli soru üretimi konusunda yeni bir yaklaşım ortaya koyulacaktır. Proje kapsamında eğitim verilerini kullanarak modelin doğru soruyu tahmin etme kabiliyetini arttırmaya çalışılacaktır. Eğitim sonrası finetuning işleminin etkisi gözlenecektir.

## 1.2 Ön İnceleme

Ön inceleme sonucunda yapılacak projeye benzer bir çalışmanın 2022 yılında TÜBİTAK'a bir araştırma makalesi olarak sunulduğu görülmüştür[2]. Çalışmada, çok dilli bir dil modeli olan mT5 modeli kullanılmıştır. mT5, metin-işleme görevlerinde esneklik sağlayan, encoder-decoder mimarisine sahip bir modeldir. Model, daha önce farklı dillerde eğitim gördüğü için Türkçe'de kullanılmaya da uygundur. Türkçe veri kümeleri ile yeniden eğitilerek Türkçe metinlerden soru üretme ve cevap çıkarma yeteneği kazandırılmıştır. Model, hem soru üretme hem de cevap çıkarma görevlerini aynı anda yapabilmesi için çok görevli bir şekilde eğitilmiştir. Bu sayede model, verilen bir bağlamdan hem soruları üretebilmekte hem de cevapları çıkarabilmektedir. Bu yaklaşım, otomatik soru üretim sürecinde herhangi bir manuel işleme gerek kalmadan tam otomatik bir sistem elde edilmesini sağlamaktadır. Eğitimi desteklemek için, TQuADv1, TQuADv2 ve XQuAD gibi Türkçe veri kümeleri kullanılmıştır. TQuAD, Türk ve İslam Bilim Tarihi üzerine oluşturulmuş Türkçe bir SQuAD benzeri bir soru-cevap veri kümesidir. Bu veri kümeleri, modelin eğitiminde ve doğrulama işlemlerinde kullanılmıştır. Her bir veri kümesinde bağlam, soru ve cevap üçlüsü yer almaktadır. Sonuç olarak Model, çok görevli yapı sayesinde verilen bir bağlamdan otomatik olarak anlamlı soru ve cevaplar üretebilmektedir. Hem soru üretimi hem de cevap çıkarımı görevlerinde yüksek performans sergileyerek, Türkçe soru-cevap veri setlerinde başarılı sonuçlar elde etmiştir.



Şekil 1.1 mT5'in yapısı

## 2 LİTERATÜR ANALİZİ

---

Yapay zeka günden güne kendini geliştirmektedir. Metin üretme odaklı eğitilen yapay zekalar da aynı doğrultuda her geçen gün daha isabetli ve daha hızlı şekilde metin üretebilmektedir. Yapay zekanın soru ve cevap üretmesi üzerine birçok araştırma yapılmıştır. Proje kapsamındaki soru üretme üzerine yapılan çalışmalar bu kısımda derlenmiştir.

Ege Üniversitesinden Katira Soleymanzadeh'in yapmış olduğu bir çalışma mevcut. Bu çalışmada biyoloji alanındaki Türkçe metinlerden otomatik soru üretimi amaçlanmıştır. Soru üretimi süreci, örnek olabilecek anlamlı cümleleri anlamak ve bunlardan soru cümleleri oluşturmak için analiz süreçlerine girilmesine dayanmaktadır. Sözdizimsel analizde bağımlılık çözümleyici kullanılarak cümledeki kelime ve ögelerin ilişkileri belirlenir. Anlamsal analiz ise biyolojik alana özgü bir önerme bankası ve anlamsal rol etiketleme sistemi ile gerçekleştirilir. Bu süreçte, her cümlede "kim", "ne", "nerede", "nasıl" ve "neden" gibi soruların oluşturulabilmesi için ögelerin anlamsal rolleri belirlenir ve etiketlenir. Böylece biyoloji metinlerinden eğitimde kullanılabilecek anlamlı ve bilgi verici soruların otomatik olarak üretilmesi hedeflenmiştir.

Çalışma, Türkçe metinlerde anlamsal analiz yaparak soru oluşturma girişiminde bulunan ilk çalışmalardan biri olup, bu amaçla biyoloji alanına özgü bir anlamsal rol etiketleme sistemi geliştirilmiştir. Geliştirilen sistem, üç aşamada işlemler gerçekleştirir: Öncelikle biyoloji alanındaki metinlerden bir anlamsal rol etiketleme veri kümesi oluşturulmuştur. Bu aşamada, metinlerdeki önemli biyolojik kavram ve terimler seçilerek etiketlenmiş, her bir cümlelerin anlamsal rollerine göre sınıflandırılması sağlanmıştır. Sonrasında, bağımlılık çözümleyiciden elde edilen verilerle birlikte anlamsal rol etiketleme sistemi kullanılarak cümlelerin yapısı çıkarılmıştır. Son aşamada ise, sözdizimsel ve anlamsal açıdan analiz edilmiş cümlelerden, uygun soru tipleri seçilerek soru oluşturma şablonlarıyla sorular üretilmiştir.

Sonuç olarak, bu otomatik soru üretim sistemi, öğretmenlerin öğrenci bilgi düzeyini değerlendirmede kullanabileceği sorular üretmenin yanı sıra öğrencilere bireysel çalışma ve kendi kendini değerlendirme imkanı da sunmaktadır. Çalışmanın başarısı, oluşturulan soruların anlamsal doğruluğu ve metinle olan ilişkisi göz önünde bulundurularak değerlendirilmektedir. Bu sistemin, biyoloji gibi karmaşık ilişkiler içeren bir alan için başarılı sonuçlar vermesi, yöntemin diğer alanlara da uyarlanabileceğini göstermektedir. Türkçe metinlerde otomatik soru üretimi üzerine yapılmış sınırlı sayıda çalışma olduğundan, bu çalışma hem eğitimde kullanılabilecek nitelikli içerik sağlama hem de Türkçe doğal dil işleme alanına katkı sunma açısından önemlidir.[3]

Seul National University'den Yanghoon Kim, Hwanhee Lee, Joongbo Shin ve Kyomin Jung yaptıkları başka bir çalışmada soru hedefini daha iyi kullanarak neural soru üretiminde performans iyileştirmesi amaçlanmıştır. Mevcut NQG modelleri, sıklıkla soru hedefindeki kelimeleri soruya dahil ederek istenmeyen sonuçlar üretmektedir. Bu çalışmada, "cevap ayrımı" yöntemine dayalı bir model geliştirilmiştir. Hedef cevabı özel bir sembolle değiştirerek, modelin bağlamı daha iyi anlaması ve doğru soru türünü belirlemesi sağlanmıştır. Ek olarak, modelin hedef cevaptaki anahtar bilgiyi yakalaması için "anahtar ağ" modülü kullanılmıştır.

Cevap ayrımlı seq2seq modeli, metindeki cevabı özel bir sembolle değiştirerek bu bilgiye odaklanır ve bu yöntem sayesinde sorular daha doğru ve beklenen şekilde üretilir. Model, SQuAD veri setinde yapılan deneylerde, önceki NQG modellerine göre daha başarılı sonuçlar elde etmiştir. Üretilen sorularda hedef cevabı içermeme oranı artmış, soru türlerini daha doğru tahmin etme başarısı gösterilmiştir. Ayrıca modelin "anahtar ağ" modülü sayesinde hedef cevaptan gerekli bilgileri doğru bir şekilde alması sağlanmış ve bu da performansı artırmıştır.[4]

Bu çalışmaya benzer NQG için başka bir çalışma da Xingwu Sun ve arkadaşları tarafından yazılan bir makalede ele alınmıştır. Bu makalede, neural ağ tabanlı soru üretimi modelindeki soru türünün cevap türüne uygun olmaması ve bağlama uzak, alakasız kelimelerin soruya eklenmesi gibi iki ana sorun ele alınmıştır. Yazarlar, bu sorunları çözmek için "cevap odaklı" ve "konum farkındalıklı" bir model önermiştir. Cevap odaklı yapı, sorunun cevapla uyumlu soru kelimeleri oluşturmasını sağlarken, konum farkındalıklı yapı, cevap çevresindeki bağlama odaklanarak soruya alakasız kelimelerin eklenmesini önler.

Bu model, SQuAD ve MARCO veri setleri üzerinde test edilmiştir ve önceki yöntemlere göre daha yüksek başarı sağlamıştır. Deneysel sonuçlar, modelin üretilen sorulardaki hata oranını azalttığını, soru kelimesi seçimini iyileştirdiğini ve daha isabetli sonuçlar

sunduğunu göstermiştir.[5]

Bir başka çalışma Kettip Kriangchaivech ve Artit Wangperawong tarafından yapılmıştır. Çalışma, transformers tabanlı bir model ile Wikipedia metinlerinden otomatik soru üretimini hedeflemektedir. Model, SQuAD veriseti üzerinde eğitilmiş ve yeni metinler üzerinden sorular üretebilmektedir. Transformers, eski RNN tabanlı modellere kıyasla daha hızlı eğitilmekte ve daha iyi performans göstermektedir. Bu yaklaşımın, eğitimcilerin sınav ve test soruları hazırlama sürecini kolaylaştıracağı öngörülmektedir.

Model, SQuAD veri setindeki bağlam ve cevaplara dayanarak sorular üretmekte ve üretilen soruların orijinal SQuAD sorularıyla karşılaştırılması için kelime hata oranı metriği kullanılmaktadır. Modelin ürettiği sorular, anlam açısından çoğunlukla doğru ve dilbilgisi açısından doğru yapıdadır, ancak orijinal sorulardan farklı olabilir.

Sonuç olarak, transformer tabanlı model, doğru dil bilgisiyle ilgili ve bağlamla uyumlu sorular üretmektedir. Gelecekteki çalışmalar, veri artırımı ve daha dengeli eğitim veri setleri ile modelin performansını daha da geliştirmeyi hedeflemektedir.[6]

Soru üretme ile alakalı başka bir çalışma da Ying-Hong Chan ve Yao-Chung Fan tarafından yapılmıştır. Çalışmada, soru üretim görevlerinde BERT tabanlı bir modelin kullanımını inceleyerek, performansı artırmayı amaçlayan yeni mimariler geliştirilmiştir. İlk olarak, BERT modelinin doğrudan kullanımı denense de, sıralı bilgi eksikliğinden dolayı yetersiz kalmıştır. Bu nedenle, önceki adımlardaki çıktıları dikkate alan sıralı bir yapı olan BERT-SQG modeli geliştirilmiştir. Daha sonra, cevap belirsizliklerini azaltmak amacıyla özel işaretleme tokenları eklenmiş ve performansı daha da artıran BERT-HLSQG modeli tasarlanmıştır.

Bu modeller, SQuAD veri setinde test edilerek, mevcut en iyi sonuçları elde etmiş ve BLEU 4 skorunu 16.85'ten 22.17'ye yükseltmiştir. Ayrıca, paragraf seviyesindeki bağlamlarda da yüksek performans göstermiştir. Değerlendirmelerde, insan tarafından üretilen sorulara yakın kalitede sorular üretebildiği gözlemlenmiştir.

Sonuç olarak, çalışma, BERT'in sıralı ve işaretlenmiş versiyonları ile soru üretiminde güçlü bir performans elde edilmesini sağlamış ve BERT-HLSQG modelinin, özellikle uzun metinlerde mevcut yöntemleri geride bıraktığını ortaya koymuştur.[7]

# 3

## SİSTEM ANALİZİ ve FİZİBİLİTE

---

### 3.1 Sistem Analizi

Proje temelde, kullanıcıdan girdi alınan bir cevaptan soruya gitmeyi amaçlamaktadır. Önceki token yöntemi ile çalışacak olan bu sistem, base modelden finetuning işlemiyle instruct model elde edilmesiyle hazırlanır.

Sistemin gereksinim duyduğu yüksek işlem kapasitesi için kişisel bilgisayarlar yerine Google Colab platformu kullanılır. Bu sayede riskler azaltılmış olur. İhtiyaç duyulması halinde Colab sürümü yükseltilerek daha hızlı ve daha çok çalışma yapılabilir.

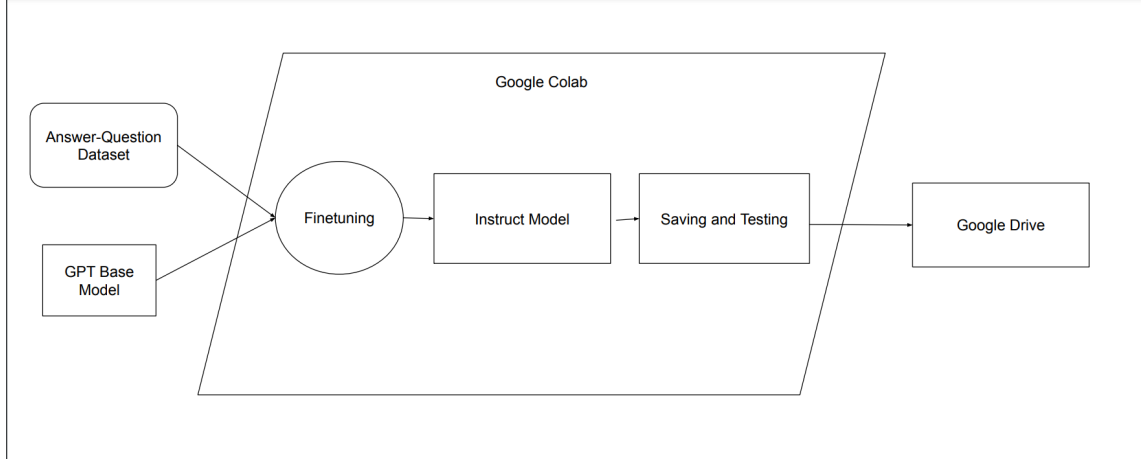
Model ve veri setlerinin Google Colab'e yüklenebilmesi için Google Drive kullanılır. Drive aracılığı ile veri seti dosya yolu olarak programa verilir. Aynı zamanda modelin kaydedileceği konum da Drive'da belirlenir

Sistem öncelikle bir GPT Türkçe dil modelini ele alır. Bu model gerekli eğitim seti ile fine tune edilerek projede kullanılacak olan instruct model elde edilir. Bu eğitim sürecinde "Pytorch Training Tool" kullanılır. Elde edilen model daha sonrasında çalıştırılmak üzere sisteme GPT Türkçe modeli yerine yüklenir. Bu işlemler Google Colab üzerinden gerçekleştirilir

Model boyutunun büyüklüğü problem yaratabileceğinden dolayı bu noktada karşılaşılan bir sorunda "8-Bit Optimizer" kullanılarak model yüklenir.

Test aşamasında ise yüklenen modele girdiler verilerek sistemin çıktıları değerlendirilir. Bu aşamada veri setinin büyüklüğü ve sistemin çıktı kalitesi karşılaştırılabilir.





**Şekil 3.1 Model Akış Diyagramı**

## 3.2 Fizibilite

Proje ihtiyaçları ve gereksinimler birçok farklı metrikte düşünülerek fizibilite maddeleri belirlenmiştir. Hazırlanan bu maddeler sonucunda projenin gerçekleştirilmesi için yeterli kaynaklara sahip olduğu ve herhangi bir engel bulunmadığı belirlenmiştir.

### 3.2.1 Teknik Fizibilite

Teknik Fizibilite yazılım ve donanım olarak ayrılarak incelenmiştir. Bunların birbiriyle uyum içinde çalışabildiği colab ve python kombinasyonu proje için uygun seçilmiştir.

#### 3.2.1.1 Donanım Fizibilitesi

Donanım konusunda 3 seçenek ele alınmıştır. Bunlar arasından hem daha erişilebilir olması, hem de Gpu anlamında çok daha güçlü olması sebebi ile Colab seçeneğinin kullanılmasına karar verilmiştir.

|               | İŞLEMCI                               | RAM      | İŞLETİM SİSTEMİ | GPU                        |
|---------------|---------------------------------------|----------|-----------------|----------------------------|
| 1. BİLGİSAYAR | M1                                    | 8GB      | MACOS           | M1                         |
| 2.BİLGİSAYAR  | (12th Gen Intel(R) Core(TM) i7-12700H | 32GB     | Windows 11 Pro  | NVIDIA GeForce RTX 3050 Ti |
| COLAB         | Intel Xeon 2x vCPU                    | 13GB RAM | Linux           | NVIDIA Tesla K80 12GB VRAM |

**Şekil 3.2 Donanım Özellikleri**

### 3.2.1.2 Yazılım Fizibilitesi

Proje boyunca yapay zeka, doğal dil işleme ve makine öğrenmesi gibi konularda önde gelen ve kullanım kolaylığı sağlayan Python programlama dili, yüksek işlem gücü avantajı sunan Google Colab platformu ile beraber kullanılacaktır. Veri işleme için Python'da bulunan pandas kütüphanesi, model kullanımı için transformers kütüphanesi ve eğitim için yine Python'da bulunan torch kütüphanesi tercih edilmiştir. Gerekli depolama işlemleri için Google Drive platformu kullanılacaktır.

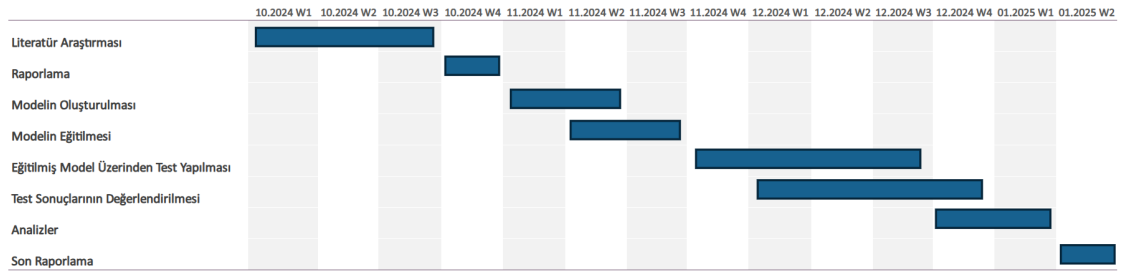
### 3.2.2 Yasal Fizibilite

Projede kullanılan yazılımlar ve işletim sistemleri lisanslıdır. Kullanılan modeller açık kaynak kodludur. Telif hakkı bulunmamaktadır. Herhangi bir yasal yükümlülük bulunmamaktadır.

### 3.2.3 Ekonomik Fizibilite

Projede kullanılacak donanımlar için ek bir ücret ödenmeyecektir. Kullanılan modeller ve yazılımlar ücretsizdir. Colab'de ihtiyaç duyulması halinde Colab Pro proje süresi boyunca aylık 165 Türk Lirasına satın alınacaktır.

### 3.2.4 İş Gücü ve Zaman Fizibilitesi



Şekil 3.3 Gantt Şeması

# 4

## SİSTEM TASARIMI

---

Sistem, hazırlanmış soru cevap dataseti ile base modeli google colab üzerinde finetune ettikten sonra kaydeder ve kullanıma hazır hale getirir.

### 4.0.0.1 Ön İşleme Adımları

Ön işleme adımlarında öncelikle elde olan hazır veri kümesi kullanıma uygun bir dataset haline çevrilir. GPT modeli optimizasyon amaçlı BitsandBytes kullanılarak 8-bit olarak yüklenir.

### 4.0.0.2 Uygulanan Yöntemler

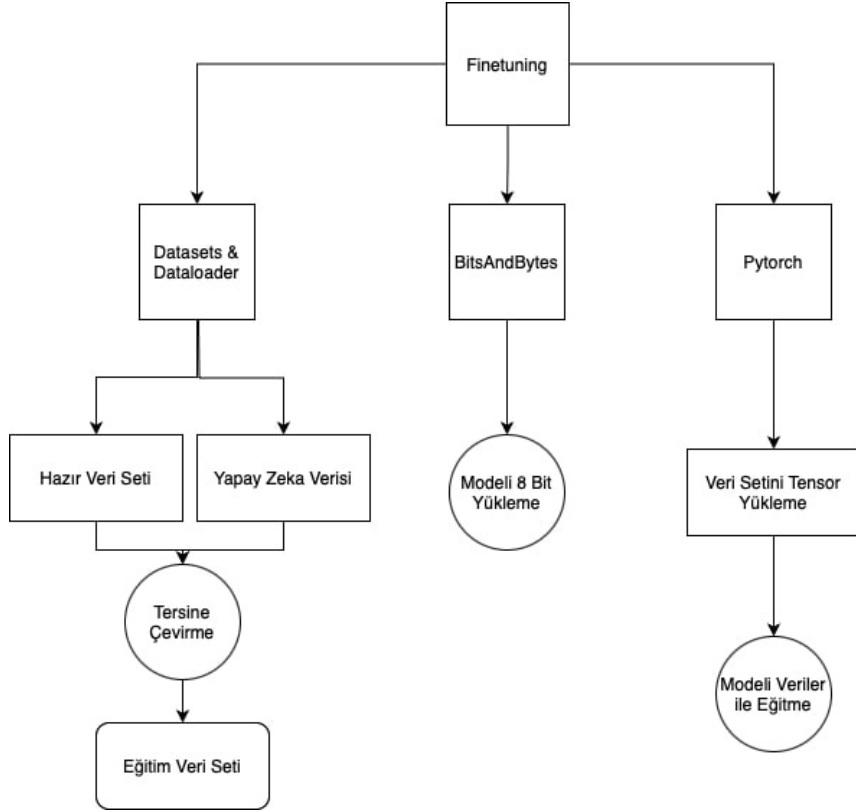
Modelin uygun şekilde çalışabilmesi için birçok yöntem kullanılmıştır. Bunlardan ilki olan modelin önceki token tahmini yapabilmesi için, verisetini oluşturan tokenler ters çevrilmiştir. Bu sayede model sonraki tokeni tahmin ederken aslında geriye doğru çalışmış olacaktır. Ayrıca kullanılan önceki token tahmini yönteminin istenilen soruyu bulduktan sonra durması ve geriye doğru daha fazla devam etmemesi için verinin uygun kısımlarına endoftext tokeni yerleştirilmiştir. Pytorch yöntemi ile birlikte yapay zeka aracılığıyla üretilen dataseti kullanılarak model eğitilmiştir. Eğitilen instruct model daha sonra kullanılabilmesi için Google Drive üzerinde kaydedilmiştir ve tekrar çalıştırılabilir durumdadır.

## 4.1 Materyal ve Veri Kümesi

Kullanılan veriseti yapay zeka tarafından oluşturulan ve daha önce yayımlanan hazır verilerin birleştirilmesiyle oluşturulmuştur. Bu veriler programın dataset yapısına uyacak şekilde düzenlenmiştir. Her bir satır bir soru cevap örneğidir. Veriler soru ve cevap şeklinde ayrılmıştır. Drive üzerindeki bir txt dosyasında saklanmakta ve çalışma esnasında Google Colab'e çekilmektedir. Verilerin çeşitliliği ve soru cevap cümlelerinin uzunluğu eğitimde büyük öneme sahiptir. Cevaplarda sorudaki cümlelere alıntı

kelimelerin kullanılması başarı oranını artırır. Aynı zamanda verilen cevapların ve sorulan soruların uzun tutulması eğitim tarafında modeli daha efektif bir hale getirir. Test tarafında modelden çıktı alınmak istenirken, soruların ise yine aynı şekilde uzun ve zengin tutulması, alınacak olan cevabın doğruluğunu artırabilir.

## 4.2 Metot



Şekil 4.1 Sistem Çalışma Metotları

Projede yapılan finetuning işlemi için data preparing, bitsandbytes, model training gibi metotlar kullanılmıştır. Finetuning işlemi için base gpt modeli baz alınmıştır. Modelin geliştirilmesi ile instruct modele ulaşmak hedeflenmiştir.

Data preparing kısmında, bulunan veri setleri ve yapay zeka cevapları projeye uygun hale getirilmiştir. Yanlış ve anlamsız soru cevap çiftleri temizlenmiştir. Data hazırlama metodu olarak PyTorch'un tensor formatındaki giriş ve çıktı verilerini bir arada tutan Datasets ve dataset'teki verileri mini batch'lere ayırarak eğitime gönderen bir metot olan DataLoader metodu kullanılmıştır[8]. Farklı uzunluktaki diziler padding token kullanarak eşitlenmiş ve PyTorch tensörlerine dönüştürülmüştür. Projenin amacı gereği tokenler eğitime girmeden önce ters çevrilmiştir.

BitsAndBytes, derin öğrenme modellerini düşük bellek kullanarak çalıştırmak için kullanılan bir kütüphanedir[9]. Bu projede modeli 8-bit olarak yüklemek için

kullanılmıştır. BitsAndBytesConfig ile model yapılandırılarak bellek kullanımı azaltılmış, daha az VRAM ile çalışabilecek hale getirilmiştir. Google Colab üzerindeki sınırlı kaynaklar bu sayede daha efektif kullanılabilmiştir.

Eğitim metodu olarak esnekliği düşünülerek Pytorch modeli kullanılmıştır. Parametreleri kullanılarak farklı seçenekler optimize model eğitimi hedeflenmiştir. Pytorch modeli, Torch kütüphanesini baz alan açık kaynaklı bir makine öğrenme kütüphanesidir[10]. Doğal dil işleme alanında yaygın bir kullanıma sahiptir.

# 5

## DENEYSEL SONUÇLAR

---

Bu bölümde projede kullanılan 2 farklı modelin Training Loss ve Validation Loss verileri karşılaştırılıp örnek senaryolarda bu iki modelin verdiği çıktılar belirtilmiştir.

### 5.1 Performans Analizi

Tablo 5.1’de ve Tablo 5.2’de görüldüğü üzere her iki model de başarılı bir şekilde eğitim almış ve düşük train loss değerlerine ulaşmıştır. Model 1, 5e-5 learning rate (LR) ile eğitilmiş ve başlangıçta 1.3449 train loss değeri ile başlayarak 0.2394 seviyesine kadar düşmüştür. Model 2 ise 1e-4 learning rate (LR) ile eğitilmiş, başlangıçta 1.4556 train loss değeri ile başlayarak 0.2149 seviyesine kadar gerilemiştir. Bu durum, her iki modelin de eğitim verisini başarıyla öğrendiğini göstermektedir.

**Tablo 5.1** Model 1 için Training Loss ve Validation Loss Bilgileri

| Epoch | Training Loss | Validation Loss |
|-------|---------------|-----------------|
| 1     | 1.3449        | 1.2842          |
| 2     | 1.0447        | 1.2683          |
| 3     | 0.7768        | 1.3086          |
| 4     | 0.6974        | 1.3401          |
| 5     | 0.6472        | 1.3591          |
| 6     | 0.5772        | 1.4003          |
| 7     | 0.4606        | 1.4303          |
| 8     | 0.3912        | 1.4598          |
| 9     | 0.3112        | 1.4959          |
| 10    | 0.2394        | 1.5131          |

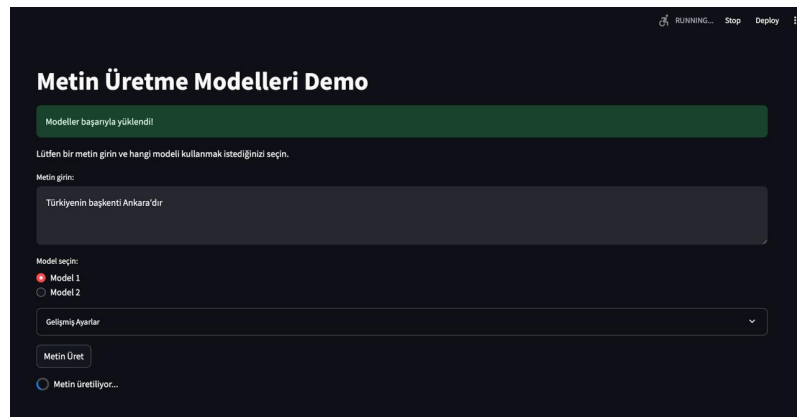
Tablo 5.2’de de görüldüğü üzere, Model 1’in validation loss değerlerinin Model 2’ye göre daha düşük olduğu ve daha yavaş bir artış göstermiştir. Model 1’in validation loss değeri ilk epoch’ta 1.2842 iken, son epoch’ta 1.5131 olarak kaydedilmiştir. Buna karşılık, Model 2’de validation loss başlangıçta 1.4111 olup, son epoch’ta 1.7023 seviyesine kadar yükselmiştir. Bu durum, Model 1’in validation set üzerinde daha iyi genelleme sağladığını ve overfitting etkisinin Model 2’ye göre daha az olduğunu göstermektedir..

**Tablo 5.2** Model 2 için Training Loss ve Validation Loss Bilgileri

| Epoch | Training Loss | Validation Loss |
|-------|---------------|-----------------|
| 1     | 1.4556        | 1.4111          |
| 2     | 1.1534        | 1.3630          |
| 3     | 0.8485        | 1.4173          |
| 4     | 0.7581        | 1.4439          |
| 5     | 0.6993        | 1.4908          |
| 6     | 0.6044        | 1.5508          |
| 7     | 0.5193        | 1.5716          |
| 8     | 0.4393        | 1.6120          |
| 9     | 0.3045        | 1.6507          |
| 10    | 0.2149        | 1.7023          |

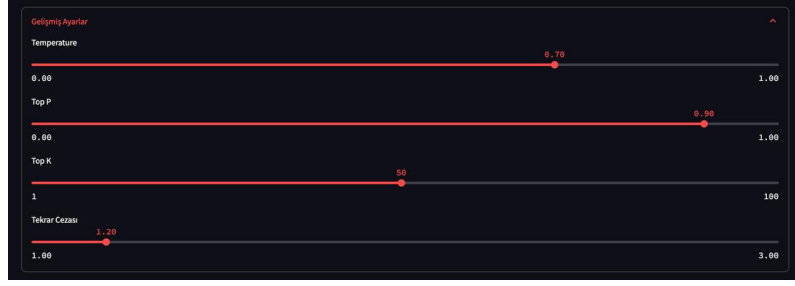
Validation loss değerleri dikkate alındığında, Model 1 daha düşük learning rate ile daha iyi bir genelleme sağlamış ve Model 2’ye göre validation set üzerinde daha başarılı sonuçlar elde etmiştir. Bu nedenle, daha küçük bir learning rate kullanmanın genelleme başarısını artırdığı söylenebilir.

Streamlit kütüphanesi kullanılarak localhost üzerinde çalıştırılan bu arayüzde, şekil 5.1’de görüldüğü gibi istenilen parametreler girilerek iki modelden de çıktı alınabilmektedir.



**Şekil 5.1** Arayüz

Şekil 5.2’de verilen Gelişmiş Ayarlar kısmında ise bu iki modele ait Temperature, Top K, Top P ve Tekrar Cezası verileri ayarlanabilmektedir.



Şekil 5.2 Gelişmiş Ayarlar

Model çıktı üretiminde kullanılan temperature, top-p, top-k ve repetition penalty gibi parametreler, modelin tahminlerini kontrol ederek farklı türde metinler üretilmesini sağlar. Temperature, modelin rastgeleliğini belirler; düşük değerler daha tutarlı, yüksek değerler ise daha yaratıcı ve çeşitli çıktılar üretir. Top-p ve top-k parametreleri, modelin en olası tokenlardan seçim yapmasını sağlar. Top-p, belirli bir kümülatif olasılığa ulaşana kadar seçim yaparken, top-k belirli sayıda en yüksek olasılıklı token arasından seçim yapar. Repetition penalty, modelin aynı ifadeleri tekrar etmesini önleyerek daha doğal ve akıcı metinler oluşturmaya yardımcı olur. Yaratıcı metinler için yüksek temperature, top-p ve top-k değerleri tercih edilirken, tutarlı ve anlamlı metinler için düşük temperature ile birlikte yüksek repetition penalty kullanılmalıdır.

### 5.1.1 Modellerin Aynı Cevaba Ürettikleri Sorular

Bu kısımda 2 modelin de aynı cevap için ürettikleri sorular incelenmiştir

---

Cevap: Türkiye Cumhuriyeti’nin başkenti Ankara’dır.

Soru - Model 1: Türkiye Cumhuriyeti’nin başkent neresidir?

Soru - Model 2: Türkiye’nin başkenti nedir?

---

Cevap: Resmi dili Çince dir.

Soru - Model 1: En çok konuşulan Çin dili nedir?

Soru - Model 2: Çin’in resmin dili nedir?

---



Cevap: Alexander Graham Bell icat etmiştir.

Soru - Model 1: İlk uçağı kim icat etti?

Soru - Model 2: Bu telefon kim tarafından icat edildi?

---

Cevap: Atatürk Selanik'te doğmuştur.

Soru - Model 1: Atatürk nerede doğdu?

Soru - Model 2: Atatürk nerede vefat etti?

---

Cevap: Güneşe en yakın gezegen Merkür'dür.

Soru - Model 1: Güneşe en yakın üçüncü nesne nedir?

Soru - Model 2: Güneşe en yakın gezegen hangisidir?

---

Cevap: İsviçre'nin 5 adet sınır komşusu vardır.

Soru - Model 1: İsviçre'nin kaç tane sınır komşusu vardır?

Soru - Model 2: İsviçre'nin yaklaşık kaç sınırı vardır?

---

Cevap: Tokyo, Japonya'nın en kalabalık şehridir.

Soru - Model 1: Japonya'nın başkenti neresidir?

Soru - Model 2: Ülkenin en kalabalık şehri Tokyo'dur?

---

Cevap: Bir yılda 12 ay vardır.

Soru - Model 1: Bir yılın süresi nedir?

Soru - Model 2: Bir yılda kaç ay var?

---

Cevap: Süper Lig’de en çok şampiyon olan takım Galatasaray’dır.

Soru - Model 1: Süper Lig’in en çok şampiyon olan takım hangisidir?

Soru - Model 2: 2018’de en çok takipçi kazanan takım neresidir?

## 5.2 Karşılaştırmalı Analiz

**Tablo 5.3** Model-1 ve Model-2 için Değerlendirme Metrikleri

| Metric  | Model-1 | Model-2 |
|---------|---------|---------|
| CIDEr   | 0.825   | 0.780   |
| METEOR  | 0.210   | 0.198   |
| ROUGE-L | 0.342   | 0.331   |
| BLEU-1  | 0.388   | 0.375   |
| BLEU-2  | 0.252   | 0.240   |
| BLEU-3  | 0.168   | 0.155   |
| BLEU-4  | 0.102   | 0.093   |

BLEU Skoru: Model-1, her bir BLEU metriğinde Model-2’den daha yüksek skor elde etmiştir. Bu, Model-1’in tahmin ettiği cümlelerin referans cümlelerle daha yüksek oranda örtüştüğünü gösterir.

ROUGE-L Skoru: ROUGE-L metriği, en uzun ortak alt diziyi dikkate alarak modelin ürettiği cümlelerin referans cümlelere olan benzerliğini ölçer. Model-1, bu metrikte de daha iyi bir performans sergilemiştir.

METEOR Skoru: METEOR metriği, eş anlamlı kelimeleri ve kök biçimleri dikkate alarak benzerliği ölçer. Model-1’in bu metrikte daha yüksek skor alması, tahminlerinin anlam açısından referanslara daha yakın olduğunu gösterir.

CIDEr Skoru: CIDEr, özellikle dil üretim görevlerinde çok önemli bir metrik olup, Model-1’in daha iyi genelleme yaptığına işaret eden en yüksek farkın gözlemlendiği metriktir.

Tablo 5.3, Model-1’in daha düşük learning rate ile daha iyi genelleme yapabildiğini ve genel performans açısından Model-2’ye kıyasla üstün olduğunu göstermektedir.

## 6 SONUÇ VE TARTIŞMA

---

Bu projede, verilen cevaptan otomatik olarak soru üretebilen bir yapay zeka modeli geliştirilmiştir. Proje kapsamında kullanılan Türkçe veri kümeleri ve finetuning yöntemleri ile modelin performansı artırılmış, üretilen soruların anlam doğruluğu ve bağlam uyumluluğu yüksek sonuçlar vermesi sağlanmıştır. Sistemin test aşamasında alınan çıktılar, modelin verilen cevaplardan tutarlı ve anlamlı sorular üretebildiğini göstermiştir.

Projenin uygulanması sırasında en önemli zorluk, büyük veri kümeleri ile çalışırken ortaya çıkan işlem kapasitesi gereksinimlerini karşılamaktı. Bu sorun, Google Colab platformu kullanılarak çözüldü.

Bu çalışmanın sonuçları, doğal dil işleme alanında Türkçe dilinde otomatik soru üretimi konusunda önemli bir adım niteliğindedir. Gelecekte, veri kümelerinin çeşitlendirilmesi, modelin daha geniş veri setleri üzerinde eğitilmesi ve farklı dillerde soru üretimi gibi konularda çalışmalar yapılabilir.

- [1] C. Yuan and X. Zhang, “Analysis of diversification of intelligent teaching in english literacy integrated classroom empowered by artificial intelligence technology in colleges and universities,” *Applied Mathematics and Nonlinear Sciences*, vol. 9, May 2024. DOI: 10.2478/amns-2024-1183.
- [2] F. Ç. Akyön, D. Çavuşoğlu, C. Cengiz, S. O. Altınuç, and A. Temizel, “Automated question generation and question answering from turkish texts,” *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 30, no. 5, pp. 1931–1940, 2022, ISSN: 1300-0632. [Online]. Available: <https://hdl.handle.net/11511/101785>.
- [3] K. Soleymanzadeh, “Domain specific automatic question generation from text,” Jan. 2017, pp. 82–88. DOI: 10.18653/v1/P17-3014.
- [4] Y. Kim, H. Lee, J. Shin, and K. Jung, “Improving neural question generation using answer separation,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 6602–6609, Jul. 2019. DOI: 10.1609/aaai.v33i01.33016602. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/4629>.
- [5] X. Sun, J. Liu, Y. Lyu, W. He, Y. Ma, and S. Wang, “Answer-focused and position-aware neural question generation,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds., Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 3930–3939. DOI: 10.18653/v1/D18-1427. [Online]. Available: <https://aclanthology.org/D18-1427>.
- [6] K. Kriangchaivech and A. Wangperawong, “Question generation by transformers,” *CoRR*, vol. abs/1909.05017, 2019. arXiv: 1909.05017. [Online]. Available: <http://arxiv.org/abs/1909.05017>.
- [7] Y.-H. Chan and Y.-C. Fan, “A recurrent BERT-based model for question generation,” in *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, A. Fisch, A. Talmor, R. Jia, M. Seo, E. Choi, and D. Chen, Eds., Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 154–162. DOI: 10.18653/v1/D19-5821. [Online]. Available: <https://aclanthology.org/D19-5821>.
- [8] S. Shen and C. M. Kittivorawong, “Efficient distributed data loading for large-scale machine learning model training with parax,”

- [9] Y. Zhao *et al.*, “Atom: Low-bit quantization for efficient and accurate llm serving,” in *Proceedings of Machine Learning and Systems*, P. Gibbons, G. Pekhimenko, and C. D. Sa, Eds., vol. 6, 2024, pp. 196–209. [Online]. Available: [https://proceedings.mlsys.org/paper\\_files/paper/2024/file/5edb57c05c81d04beb716ef1d542fe9e-Paper-Conference.pdf](https://proceedings.mlsys.org/paper_files/paper/2024/file/5edb57c05c81d04beb716ef1d542fe9e-Paper-Conference.pdf).
- [10] S. Li *et al.*, *Pytorch distributed: Experiences on accelerating data parallel training*, 2020. arXiv: 2006.15704 [cs.DC]. [Online]. Available: <https://arxiv.org/abs/2006.15704>.

### BİRİNCİ ÜYE

**İsim-Soyisim:** TALHA ÇELİK

**Doğum Tarihi ve Yeri:** 16.07.2003, İstanbul

**E-mail:** talha.celik1@std.yildiz.edu.tr

**Telefon:** 0537 456 46 90

**Staj Tecrübeleri:** Turkcell Teknoloji - Yapay Zeka ve Analitik Çözümler  
Bosch Türkiye - Bilgi Teknolojileri

### İKİNCİ ÜYE

**İsim-Soyisim:** HÜSEYİN SAİD ARICI

**Doğum Tarihi ve Yeri:** 19.02.2002, İstanbul

**E-mail:** said.arici@std.yildiz.tr

**Telefon:** 0536 073 39 25

**Staj Tecrübeleri:** SOFT İş Çözümleri A.Ş - Yazılım Departmanı

### Proje Sistem Bilgileri

**Sistem ve Yazılım:** MacOS İşletim Sistemi, Google Colab, Python

**Gerekli RAM:** 32GB

**Gerekli Disk:** 64GB