Sağlık Alanında Büyük Dil Modellerinin Adaptasyonu ve Veri Toplama ve Hazırlama Süreci: Bir Pratik Örnek

Yazar [*] Yazar [†] Yazar [‡]

24 Mayıs 2024

Özet

Büyük dil modelleri (BDM), tıbbi bilgiye erişimi iyileştirmek, hastalarla iletişimi güçlendirmek ve yeni tedaviler geliştirmek gibi potansiyelleri ile sağlık alanında devrim yaratma potansiyeline sahiptir. Ancak, BDM'lerin sağlık alanında etkili ve güvenli bir şekilde kullanılabilmesi için yüksek kaliteli veri toplamak ve hazırlamak gereklidir. Bu makalede, doktorlar tarafından hastaların sorduğu sorulara cevaplar verilen ve herkese açık olarak paylaşılan bir web sitesinden elde edilen doktor anonim profilleri ve soru-cevap verileri kullanılarak BDM adaptasyonu için veri toplama ve hazırlama süreci ele alınmıştır. Veri toplama, veri birleştirme, boş değerlerin işlenmesi, veri tipi dönüşümü, veri temizleme, veri kalite kontrolü, BDM eğitimine hazırlık ve metin ön işleme gibi adımlar detaylı olarak açıklanmıştır. Hazırlanan veriler kullanılarak Meta şirketinin geliştirdiği LLAMA 3 modeli ve YTÜ COSMOS yapay zeka araştırma grubunun geliştirdiği cosmosGPT v0.1 modeli üzerinde fine-tuning işlemi gerçekleştirilmiştir. Böylece modeller, sağlık alanında Türkçe sorulan sorulara daha iyi cevaplar verebilmeye başlamıştır. Bu çalışma, BDM'lerin sağlık alanında kullanımı için kaliteli veri toplama ve hazırlamanın önemini vurgulamaktadır.

Anahtar kelimeler: Büyük Dil Modelleri, Sağlık Alanı, Veri Toplama, Veri Hazırlama, Kalite Kontrol, BDM Eğitimi, Fine-tuning.

Abstract

Large language models (LLMs) have the potential to revolutionize the field of healthcare with their capabilities to improve access to medical information, strengthen communication with patients, and develop new treatments. However, for LLMs to be effectively and safely used in the healthcare field, it is necessary to collect and prepare high-quality data. This article discusses the process of data collection and preparation for LLM adaptation using doctor anonymous profiles and question-answer data obtained from a website where doctors answer questions asked by patients and shared publicly. Steps such as data collection, data merging, handling missing values, data type conversion, data cleaning, data quality control, preparation for LLM training, and text preprocessing are detailed. The prepared data is used for fine-tuning the LLAMA 3 model developed by Meta company and the cosmosGPT v0.1 model developed by YTU COSMOS artificial intelligence research group. As a result, the models have started to provide better answers to questions asked in Turkish in the healthcare field. This study highlights the importance of collecting and preparing quality data for the use of LLMs in healthcare.

Keywords: Large Language Models, Healthcare Field, Data Collection, Data Preparation, Quality Control, LLM Training, Fine-tuning.

1 Giriş

Sağlık hizmetleri, insan yaşamının en önemli ve hassas alanlarından biridir. Sağlıklı bir yaşam sürdürebilmek için doğru ve zamanında tıbbi bilgiye erişim, hastalıkların doğru teşhisi ve etkili tedavi yöntemlerinin uygulanması büyük önem taşımaktadır. Son yıllarda teknolojinin hızla gelişmesi, sağlık hizmetlerinin sunumunda, teşhis ve tedavi süreçlerinde yeni ve heyecan verici olanaklar yaratmaktadır. Yapay zeka (YZ), bu dönüşümün ön saflarında yer alan teknolojilerden biridir. YZ, karmaşık tıbbi verileri analiz ederek, hastalıkları teşhis etmede, kişiselleştirilmiş tedavi planları oluşturmada ve hatta yeni ilaçlar keşfetmede kullanılabilme potansiyeline sahiptir. Bu potansiyelin en önemli temsilcilerinden biri de Büyük Dil Modelleri'dir (BDM). BDM'ler, devasa metin veri kümeleri üzerinde eğitilmiş derin öğrenme algoritmalarıdır. Bu modeller, doğal dili anlama, yorumlama ve üretme konusunda son derece yeteneklidirler. İnsanlar gibi metinleri okuyabilir, yazabilir, özetleyebilir ve hatta farklı diller arasında çeviri yapabilirler. Sağlık alanında BDM'lerin potansiyel uygulamaları oldukça geniştir. Örneğin:

Tıbbi Bilgiye Erişim: Hastalar, BDM'ler aracılığıyla tıbbi bilgilerine kolayca erişebilir, hastalıkları hakkında bilgi edinebilir, semptomlarını değerlendirebilir ve tedavi seçenekleri hakkında bilgi alabilirler.

Hasta-Doktor İletişimi: BDM'ler, hasta-doktor iletişimini kolaylaştırmak için kullanılabilir. Örneğin, hastaların sorularını yanıtlayarak, randevu planlamasına yardımcı olarak ve doktorlara hastaların tıbbi geçmişleri hakkında bilgi sağlayarak iletişim süreçlerini daha verimli hale getirebilirler.

Tıbbi Teşhis: BDM'ler, hastaların tıbbi kayıtlarını, semptomlarını ve tıbbi geçmişlerini analiz ederek doktorlara

[*]Üniversitesi, Bilgisayar Mühendisliği Bölümü, İstanbul, Türkiye
[†]Üniversitesi, Bilgisayar Mühendisliği Bölümü, İstanbul, Türkiye
[‡]Üniversitesi, Bilgisayar Mühendisliği Bölümü, İstanbul, Türkiye

teşhis koymada yardımcı olabilirler. **Tedavi Planlaması:** BDM'ler, hastaların tıbbi geçmişlerini ve semptomlarını analiz ederek, kişiselleştirilmiş tedavi planları oluşturabilir ve tedavi süreçlerini optimize edebilirler. Ancak, BDM'lerin sağlık alanındaki tüm bu potansiyel faydalarını gerçekleştirebilmesi için, bu alana özgü yüksek kaliteli verilerle eğitilmeleri gerekmektedir. Tıbbi metinler, karmaşık terminolojiye, hastalık sınıflandırmalarına, tedavi yöntemlerine ve hasta-doktor iletişim dinamiklerine sahiptir. BDM'lerin bu alandaki verileri doğru bir şekilde anlayabilmesi ve işleyebilmesi için, bu verilere özgü bir şekilde eğitilmeleri gerekmektedir. Bu noktada, veri toplama ve hazırlama süreçleri büyük önem kazanmaktadır. BDM'lerin sağlık alanına adaptasyonu, bu modellerin sadece genel dil yapısını değil, aynı zamanda sağlık alanına özgü terminolojiyi, hastalık sınıflandırmalarını, tedavi yöntemlerini ve hasta-doktor iletişim dinamiklerini anlamalarını gerektirir. Bu nedenle, BDM'lerin sağlık alanında etkili bir şekilde kullanılabilmesi için, bu alana özgü verilerin toplanması, temizlenmesi, yapılandırılması ve özenle hazırlanması gerekmektedir. Bu çalışmanın temel amacı, sağlık alanında özelleştirilmiş bir BDM oluşturmak için gerekli veri toplama ve hazırlama süreçlerini detaylı bir şekilde açıklamak ve bu sürecin, BDM'lerin sağlık alanındaki performansını nasıl etkilediğini göstermektir. Bu amaçla iki farklı BDM modeli kullanılacaktır: Meta tarafından geliştirilen LLAMA 3 [1] ve YTÜ COSMOS yapay zeka araştırma grubunun geliştirdiği cosmosGPT v0.1 [2]. Bu modellerin farklı yetenekleri ve eğitim verileri, BDM adaptasyon süreci ve sağlık alanına özgü verilerin etkisini daha iyi anlamamızı sağlayacaktır. **LLAMA 3**, Meta tarafından geliştirilen açık kaynak kodlu bir BDM'dir. LLAMA 3, geniş bir metin veri kümesi üzerinde eğitilmiş olup, doğal dil işleme görevlerinde etkileyici bir performans sergilemektedir. Ancak, genel amaçlı bir model olması nedeniyle, sağlık alanına özgü terminoloji, hastalık sınıflandırmaları, tedavi yöntemleri ve hasta-doktor iletişim dinamikleri konusunda yeterli bilgiye sahip değildir. **cosmosGPT v0.1** ise, YTÜ COSMOS yapay zeka araştırma grubu tarafından geliştirilmiş ve özellikle Türkçe metinler üzerinde eğitilmiş bir BDM'dir. Bu model, Türkçe dil yapısını ve yaygın kullanılan Türkçe kelime dağarcığını iyi bir şekilde anlamakta ve işlemektedir. Ancak, LLAMA 3 gibi, cosmosGPT v0.1 de sağlık alanına özgü bilgiler konusunda eksikliklere sahiptir. Bu çalışmada, LLAMA 3 ve cosmosGPT v0.1 modellerinin sağlık alanına adaptasyonu için fine-tuning yöntemi kullanılacaktır. Fine-tuning, önceden eğitilmiş bir BDM'nin, yeni bir göreve veya alana özgü verilerle ek olarak eğitilmesi işlemidir. Bu sayede, model belirli bir alandaki performansını artırabilir. Bu çalışmada kullanılan sağlık alanına özgü veri seti, doktorlar tarafından hastaların sorduğu sorulara verilen cevaplardan oluşmaktadır. Bu veri seti, BDM'lerin sağlık alanındaki terminolojiyi, hastalık sınıflandırmalarını ve hasta-doktor iletişim dinamiklerini öğrenmelerini sağlayacaktır. Fine-tuning işlemi sırasında, LLAMA 3 ve cosmosGPT v0.1 modelleri bu veri seti üzerinde eğitilecek ve sağlık alanına özgü sorulara daha doğru ve alakalı cevaplar vermesi hedeflenecektir. Modellerin performansı, eğitim ve test veri setleri kullanılarak değerlendirilecektir. Fine-tuning işlemi sonucunda, LLAMA 3 ve cosmosGPT v0.1 modellerinin sağlık alanındaki performansının artması ve sağlık hizmetleri alanında kullanılabilecek özelleştirilmiş BDM'ler olabilme potansiyelleri gözlemlenecektir. Başarımın test edilmesi için ayrıca küçük bir veri seti üzerinden modellerin verdiği cevaplar sağlık profesyonelleri tarafından değerlendirilecektir. Bu değerlendirme sonucunda, modellerin sağlık alanındaki terminolojiyi, hastalık sınıflandırmalarını ve hasta-doktor iletişim dinamiklerini ne kadar iyi anladığı ve doğru cevaplar verdiği belirlenecektir. Bu çalışma, BDM'lerin sağlık hizmetlerinde kullanımı için atılan önemli bir adımdır. BDM'lerin sağlık alanındaki potansiyel faydaları çok çeşitlidir. Ancak, bu potansiyeli tam olarak gerçekleştirebilmek için, doğru ve etkili veri toplama ve hazırlama süreçlerine dikkat edilmesi gerekmektedir. Bu çalışma, bu süreçleri detaylı bir şekilde açıklayarak, BDM'lerin sağlık alanındaki uygulamalarına yönelik araştırmalara katkı sağlamayı amaçlamaktadır.

2 Veri Toplama ve Hazırlama Süreci

Veri toplama süreci, doktorların hastalara verdiği cevapları içeren doktor profilleri ve soru-cevap verilerini içeren bir veri seti üzerinde gerçekleştirilecektir. Bu veri seti, doktorsitesi.com adlı bir web sitesinden elde edilmiştir. Bu web sitesi, doktorların doğrulanmış profilleri ve hastaların sorduğu sorulara verilen cevapları içeren bir platformdur. Bu web sitesi, doktorların uzmanlık alanları, eğitim geçmişleri, çalıştıkları hastaneler ve hastaların sorduğu sorulara verdikleri cevaplar gibi bilgileri içermektedir. Bu bilgiler, BDM'lerin sağlık alanındaki terminolojiyi, hastalık sınıflandırmalarını ve hasta-doktor iletişim dinamiklerini öğrenmeleri için gerekli verileri sağlayacaktır. Veri seti, web scraping yöntemleri kullanılarak elde edilmiş ve belirli bir formata dönüştürülmüştür. Bu formatta, her bir doktor profili ve hastaların sorduğu sorulara verilen cevaplar ayrı ayrı kaydedilmiştir. Bu veri seti, BDM'lerin eğitiminde kullanılmak üzere hazırlanmıştır. Veri toplama süreci, web scraping yöntemleri kullanılarak gerçekleştirilecektir.

2.1 Verilerin Toplanması için Gerekli Araçların Hazırlanması

Kullanılan Bilgisayar: Macbook Pro 16-inch, 2023, Apple M2 Max chip, 64 GB RAM, 1 TB SSD depolama, macOS Version 14.5 (23F79) işletim sistemi.
Veri Kaynağı: doktorsitesi.com
Programlama Dili: Python 3.9.6
Kodlama Ortamı: Visual Studio Code Versiyon: 1.89.1 (MacOS) ve Eklenti olarak Jupiter Notebook Versiyon: 2024.4.0
Kullanılan Kütüphaneler:

- requests: Internete açılan bağlantılar üzerinden veri alışverişi yapmak için kullanılacak.
- BeautifulSoup: Web sayfalarından alınan HTML verilerini parse etmek ve analiz etmek için kullanılacak. [3]
- asyncio: Asenkron programlama yapmak için kullanılacak.
- aiohttp: Asenkron HTTP istekleri yapmak için kullanılacak.

- pandas: Veri analizi ve işleme için kullanılacak.
- json: JSON verileri işlemek için kullanılacak.

2.2 Verilerin Toplanması

İlk olarak https://www.doktorsitesi.com/tumuzmanlar sayfasına 2024 Mayıs ayı itibariyle ulaşıldı. Sayfada her bir alt sayfada 20 doktor profilinin ünvanları ile birlikte listelendiği toplam 998 alt sayfa olduğu görüldü. Bu alt sayfalardan her birine istek atılarak isimleri ve profil linkleri alındı. Bu linkler ve doktor isimleri bir DataFrame'e kaydedildi. Daha sonra kullanılmak üzere de csv formatında diske kaydedildi.

```
from bs4 import BeautifulSoup


def htmlden_doktorlari_al(html):
    soup = BeautifulSoup(html, 'html.parser')
    az_content = soup.find('div', class_='az-content')
    az_main_wrappers = az_content.find_all('div', class_='az-main-wrapper')
    doktorlar = []
    for az_main_wrapper in az_main_wrappers:
        verified = az_main_wrapper.find('div', class_='verified')
        dogrulanmis_profil = False
        if verified is not None:
            dogrulanmis_profil = verified.text
        resim = az_main_wrapper.find('img')
        resim_linki = resim['src']
        cinsiyet = resim['data-gender']
        profil = az_main_wrapper.find('a')
        profil_linki = profil['href']
        konum = profil_linki.split('/')[-1]
        uzmanlik_alani = profil_linki.split('/')[-2]
        unvan = profil.find('span').text
        isim = profil.text.split('\n')[2].strip()

        doktorlar.append({
            'resim_linki': resim_linki,
            'unvan': unvan,
            'isim': isim,
            'dogrulanmis_profil': dogrulanmis_profil,
            'profil_linki': profil_linki,
            'cinsiyet': cinsiyet,
            'konum': konum,
            'uzmanlik_alani': uzmanlik_alani
        })
    return doktorlar
```

Listing 1: HTML'den Doktorları Alma

```
import requests

tum_doktorlar = []
for i in range(1, 999):
    response = requests.get("https://www.doktorsitesi.com/doktorlar?sayfa=" + str(i))
    doktorlar = htmlden_doktorlari_al(response.text)
    tum_doktorlar.extend(doktorlar)
```

Listing 2: Doktor Profil Özetlerini Request ile Getirme

Paralelleştirme olmadan 16 dakika süren bu işlem, süreyi azaltmak amacıyla asyncio ve aiohttp kütüphaneleri kullanılarak asenkron bir şekilde yapıldığında 55 saniyeye indirildi.

```
import asyncio
import time

import aiohttp

hatali_islemler = []
tum_doktorlar = []

async def doktor_profil_ozetini_getir(session, sayfa_no):
    url = "https://www.doktorsitesi.com/tumuzmanlar?sayfa=" + str(sayfa_no)
    try:
        async with session.get(url) as response:
            html = await response.text()
            doktorlar = htmlden_doktorlari_al(html)
            tum_doktorlar.extend(doktorlar)
    except Exception as e:
        print(f"Error: {e}")
        hatali_islemler.append(sayfa_no)

baslama_zamani = time.time()
async with aiohttp.ClientSession() as session:
    gorevler = []
    for i in range(1, 9):
        gorevler.append(doktor_profil_ozetini_getir(session, i))
    await asyncio.gather(*gorevler)
bitis_zamani = time.time()
print(f"Toplam {len(tum_doktorlar)} doktor profili cekildi. Hatali islem sayisi: {len(hatali_islemler)} Toplam sure: {bitis_zamani - baslama_zamani} saniye")
```

Listing 3: Doktor Profil Özetlerini Getirme ve Paralelleştirme

```
import pandas as pd

tum_doktorlar_df = pd.DataFrame(tum_doktorlar)
tum_doktorlar_df.to_csv('tum_doktorlar.csv', index=False)
tum_doktorlar_df.head()
```

Listing 4: Doktor Profil Özetlerini DataFrame ve CSV'e Kaydetme

Tablo 1: Doktor Profil Özetleri

unvan	isim	profil_linki	cinsiyet	konum	uzmanlik_alani
Uzm. Dr.	*** ***	https://doktorsitesi...ve-dogum/ankara	female	ankara	kadin-hastaliklari-ve-dogum
Prof. Dr.	*** ***	https://doktorsitesi...koloji/izmir	female	izmir	psikoloji
Dyt.	*** ***	https://doktorsitesi...syen/zonguldak	female	zonguldak	diyetisyen
Op. Dr.	*** ***	https://doktorsitesi...ogum/sakarya	male	sakarya	kadin-hastaliklari-ve-dogum
Dt.	*** ***	https://doktorsitesi...onti/istanbul	female	istanbul	dis-hekimi-ortodonti

3 Yöntemler

Lorem ipsum dolor sit amet, consectetur adipiscing elit

4 Bulgular

Lorem ipsum dolor sit amet, consectetur adipiscing elit

5 Tartışma

Lorem ipsum dolor sit amet, consectetur adipiscing elit

6 Sonuç

Lorem ipsum dolor sit amet, consectetur adipiscing elit

Kaynaklar

[1] Meta, "Introducing meta llama 3: The most capable openly available llm to date," 2024.

[2] H. T. Kesgin, M. K. Yuce, E. Dogan, M. E. Uzun, A. Uz, H. E. Seyrek, A. Zeer, and M. F. Amasyali, "Introducing cosmosgpt: Monolingual training for turkish language models," *arXiv preprint arXiv:2404.17336*, 2024.

[3] L. Richardson, "Beautiful soup documentation," *April*, 2007.