

Veri Seti Zenginleştirme ve Geliştirme Rehberi

1. Giriş

Mevcut `telekom_dataset_2500.json` veri setimiz, çok turlu diyalog yapısı ve yapısal araç çağırma formatı ile model eğitimi için **sağlam bir temel** oluşturmaktadır. Ancak, "çalışan" bir model ile "olağanüstü" bir yapay zeka asistanı arasındaki fark, veri setinin zenginliği, çeşitliliği ve gerçek dünya senaryolarını ne kadar yansıttığı ile doğrudan ilişkilidir.

Bu rehber, mevcut veri setimizi bir sonraki seviyeye taşımak için eklenmesi ve geliştirilmesi gereken kritik senaryo türlerini açıklamaktadır.

2. Geliştirilmesi Gereken Alanlar

Alan 1: "Mutlu Son" Senaryolarının Ötesine Geçmek

- **Mevcut Durum:** Veri setimiz, kullanıcının ne istediğini net bir şekilde söylediği ve asistanın tek seferde doğru aracı çağırarak sorunu çözdüğü "ideal" senaryolarla doludur.
- **Problem:** Gerçek kullanıcılar genellikle kararsızdır, fikir değiştirir, eksik bilgi verir veya ne istediğini tam olarak bilmez. Modelimiz bu belirsizlik durumlarını yönetebilmelidir.
- **Çözüm:** Modelin sadece komut uygulayan değil, aynı zamanda durumu anlayıp kullanıcıya yardımcı olan bir asistan olmasını sağlamak için karmaşık ve kararsız kullanıcı diyalogları eklenmelidir.
- **Örnek Senaryo ("Kararsız Kullanıcı"):**

```
{
  "rol": "kullanici", "icerik": "Paketimi yükseltmek istiyorum."
},
{
  "rol": "asistan", "icerik": "Elbette, 'Süper Hız 50GB' ve 'Mega İnternet 100GB' paketlerimiz mevcut. Hangisiyle ilgilenirsiniz?"
},
{
  "rol": "kullanici", "icerik": "Hmm, aslında emin olamadım. Önce mevcut kullanımımı öğrenebilir miyim?"
},
{
  "rol": "asistan",
  "arac_cagrilari": [{ "fonksiyon": "get_data_usage", "parametreler": {} }]
}
```

Alan 2: Pasif Asistandan Proaktif Asistana Geçiş

- **Mevcut Durum:** Asistan, çoğunlukla kullanıcının komutlarına birebir yanıt veren reaktif bir roledir.

- **Problem:** Değer katan bir asistan, sadece söyleneni yapmakla kalmaz, aynı zamanda kullanıcıya fayda sağlayacak önerilerde bulunur ve onu proaktif olarak yönlendirir.
- **Çözüm:** Asistanın, kullanıcının durumunu analiz ederek ona özel teklifler veya hatırlatmalar yaptığı senaryolar eklenmelidir.
- **Örnek Senaryo ("Proaktif Öneri"):**

```
{
  "rol": "kullanici", "icerik": "Faturamı öde."
},
{
  "rol": "asistan",
  "arac_cagrilari": [{ "fonksiyon": "pay_bill", "parametreler": {"amount":
"150TL"} }]
},
{
  "rol": "asistan", "icerik": "Faturanız başarıyla ödendi. Bu arada, son 3
aydır faturanızı ayın son gününde ödediğinizi fark ettim. Dilerseniz
otomatik ödeme talimatı vererek gelecekteki faturalarınızda %5 indirim
kazanabilirsiniz. Kurulum yapmamı ister misiniz?"
}
```

Alan 3: Tekil Araç Kullanımından Zincirleme Araç Kullanımına

- **Mevcut Durum:** Diyaloglar genellikle tek bir amaca hizmet eden tek bir araç çağrısı içeriyor.
- **Problem:** Gerçek hayattaki karmaşık talepler, sonuca ulaşmak için birden fazla aracın mantıksal bir sıra ile çağrılmasını gerektirebilir.
- **Çözüm:** Bir kullanıcı isteğini çözmek için asistanın birden fazla adımı (ve dolayısıyla birden fazla araç çağrısını) takip ettiği zincirleme senaryolar oluşturulmalıdır.
- **Örnek Senaryo ("Zincirleme Araç Kullanımı"):**

```
{
  "rol": "kullanici", "icerik": "Yurtdışına çıkacağım, hattımı kullanıma
açtırıp bana en uygun roaming paketini önerir misin?"
},
{
  "rol": "asistan",
  "arac_cagrilari": [{ "fonksiyon": "enable_roaming", "parametreler":
{"status": true} }]
},
{
  "rol": "asistan", "icerik": "Hattınızı yurtdışı kullanımına başarıyla
açtım. Size en uygun paketi önerebilmem için hangi ülkeye seyahat
edeceğinizi öğrenebilir miyim?"
},
{
```

```
"rol": "kullanici", "icerik": "Almanya'ya gidiyorum."
},
{
  "rol": "asistan",
  "arac_cagrilari": [{ "fonksiyon": "get_available_roaming_packages",
    "parametreler": {"country": "Almanya"} }]
}
```

Alan 4: Hata ve İstisna Durumlarının Yönetimi

- **Mevcut Durum:** Veri setimiz, tüm API çağrılarının her zaman başarılı bir şekilde çalıştığı varsayımına dayanmaktadır.
- **Problem:** Gerçek dünyada sistemler hata verebilir, API'lar ulaşılamaz olabilir veya istenen veri bulunamayabilir. Model, bu tür istisnai durumları zarafetle yönetebilmelidir.
- **Çözüm:** Araç çağırma denemesinin başarısız olduğu ve asistanın kullanıcıya durumu düzgün bir şekilde açıkladığı senaryolar eklenmelidir.
- **Örnek Senaryo ("Başarısız API Çağırısı"):**

```
{
  "rol": "kullanici", "icerik": "T-12345 numaralı arıza kaydımın durumunu
  öğrenebilir miyim?"
},
{
  "rol": "asistan",
  "arac_cagrilari": [{ "fonksiyon": "get_fault_ticket_status",
    "parametreler": {"ticket_id": "T-12345"} }]
  // Bu çağrının bir 'API_ERROR' döndüğünü varsayalım.
},
{
  "rol": "asistan", "icerik": "Talebinizi işleme almaya çalışırken anlık bir
  sorun oluştu ve arıza kayıt sistemine ulaşamadım. Çok özür dilerim. Lütfen
  birkaç dakika sonra tekrar dener misiniz?"
}
```

Alan 5: Araç Gerektirmeyen Doğal Sohbet Akışları

- **Mevcut Durum:** Veri seti, neredeyse tamamen bir araç çağırısı ile sonuçlanan görev odaklı diyaloglardan oluşmaktadır.
- **Problem:** Model, her kullanıcı girdisini bir araç çağırma fırsatı olarak görmemelidir. Genel sohbeti ("Merhaba", "Teşekkürler", "Siz kimsiniz?") yönetebilmeli ve anlamsız isteklere uygun cevaplar verebilmelidir.
- **Çözüm:** Asistanın herhangi bir araç çağırmadan sohbet ettiği, soru yanıtladığı veya yardımcı olamayacağını belirttiği diyaloglar eklenmelidir.
- **Örnek Senaryo ("Normal Sohbet"):**

```
{
  "rol": "kullanici", "icerik": "Siz bir yapay zeka mısınız?"
},
{
  "rol": "asistan", "icerik": "Evet, size telekomünikasyon hizmetlerinizle ilgili yardımcı olmak için tasarlanmış bir yapay zeka asistanıyım. Size nasıl destek olabilirim?"
}
```