# LLM Kullanarak Sentetik Metin Veri Üretimi için Hazır Python Kütüphaneleri ve GitHub Depoları

**1. Giriş**

Sentetik veri, gerçek dünya verilerinin özelliklerini taklit etmek üzere yapay olarak oluşturulan bilgilerdir ve çeşitli alanlarda giderek artan bir öneme sahiptir. Özellikle, gerçek dünya verilerinin kıt, hassas veya elde edilmesi maliyetli olduğu durumlarda, Büyük Dil Modellerini (LLM'ler) eğitmek ve değerlendirmek için sentetik veri kullanımı önemli avantajlar sunmaktadır.1 LLM'lerin çeşitli Doğal Dil İşleme (NLP) görevlerindeki başarısı, sağlam ve çeşitli veri kümelerine olan ihtiyacı artırmıştır. Sentetik veri, bu ihtiyacı karşılamak için umut verici bir çözüm sunmaktadır.

LLM'lerin sentetik metin verisi üretimindeki özel uygulaması, geleneksel yöntemlere (veri artırma veya manuel oluşturma gibi) kıyasla zenginlik, nüans ve bağlamsal uygunluk açısından önemli avantajlar sunar.3 LLM'ler, dilin yapısı ve bağlamı hakkındaki doğal anlayışları sayesinde, kural tabanlı veya daha basit üretken modellerin oluşturabileceğinden çok daha karmaşık sentetik metinler üretebilirler. Bu raporun amacı, LLM tabanlı sentetik metin veri üretimini kolaylaştıran, kullanıma hazır Python kütüphanelerine ve GitHub depolarına ayrıntılı bir genel bakış sunmaktır. Rapor, bu araçların temel amaçlarını, sundukları ana özellikleri, ilgili web bağlantılarını ve sentetik veri üretimi sürecini nasıl kolaylaştırdıklarına dair örnek kullanımlarını içerecektir.

Bu raporun yapısı şu şekildedir: İlk olarak, LLM tabanlı sentetik metin veri üretimi için temel Python kütüphaneleri incelenecektir. Ardından, bu alandaki ilgili GitHub depoları ele alınacaktır. Raporun ilerleyen bölümlerinde, LLM sentetik veri üretiminde dikkate alınması gereken önemli teknikler ve hususlar tartışılacaktır. Son olarak, elde edilen bulgular özetlenecek ve bu alandaki gelecekteki yönelimlere dair çıkarımlar sunulacaktır.

**2. LLM Tabanlı Sentetik Metin Veri Üretimi için Python Kütüphaneleri**

* **2.1. Hugging Face Transformers** 6  
  Hugging Face Transformers kütüphanesi, önceden eğitilmiş LLM'ler ve ilgili araçlar için merkezi bir platform olarak hizmet vermektedir. Kütüphane, metin üretimi de dahil olmak üzere çeşitli NLP görevleri için yaygın olarak kullanılmaktadır. Transformers, BERT, T5, Falcon, LLaMA gibi binlerce önceden eğitilmiş açık kaynaklı modeli kullanıma sunmaktadır ve Hugging Face'in genişleyen LLM ekosisteminin amiral gemisi kütüphanesidir.16  
  Kütüphanenin temel özelliklerinden biri, çeşitli modellerden metin üretmek için kullanılan generate() yöntemidir.8 Bu yöntem, GPT, Mistral ve Llama serisi gibi popüler modellerle uyumludur. generate() yönteminin temel kullanımı, bir başlangıç metni (prompt) sağlamayı ve modelin bu metni devam ettirmesine veya tamamlamasına izin vermeyi içerir.7 Örneğin, aşağıdaki gibi basit bir kodla metin üretilebilir:  
  Python  
  from transformers import pipeline  
    
  generator = pipeline('text-generation', model='gpt2')  
  result = generator("Merhaba dünya, devam et...")  
  print(result)  
    
  Transformers kütüphanesi, sentetik veri üretimi için çeşitli kod çözme stratejileri sunar.8 Açgözlü arama (greedy search), her adımda en yüksek olasılığa sahip olan kelimeyi seçerken, ışın arama (beam search) birden fazla olası kelime dizisini takip ederek daha tutarlı sonuçlar elde etmeyi hedefler. Örnekleme (sampling) yöntemleri ise, olasılık dağılımından rastgele kelimeler seçerek daha çeşitli çıktılar üretir. Bu stratejiler, sentetik verinin istenen özelliklerine göre ayarlanabilir.  
  GenerationConfig sınıfındaki parametreler aracılığıyla metin üretimi özelleştirilebilir.8 max\_new\_tokens parametresi, üretilecek maksimum yeni token sayısını belirlerken, num\_beams ışın arama sırasında takip edilecek hipotez sayısını kontrol eder. do\_sample parametresi etkinleştirildiğinde, olasılık dağılımından örnekleme yapılır ve top\_k ile top\_p parametreleri, örnekleme sırasında dikkate alınacak olası kelimelerin sayısını ve olasılık kütlesini sınırlar.  
  Transformers, metin sınıflandırma veya diyalog gibi belirli görevler için sentetik veri üretmek amacıyla da kullanılabilir.6 LLM'lere uygun promptlar veya talimatlar sağlanarak, istenen formatta ve içerikte sentetik veri oluşturulabilir. Transformers kütüphanesinin esnekliği ve çok çeşitli önceden eğitilmiş modellerin mevcudiyeti, farklı ihtiyaçlara yönelik sentetik metin oluşturmak için onu çok yönlü bir araç haline getirir. Hugging Face Hub entegrasyonu sayesinde, çok sayıda önceden eğitilmiş modele ve veri kümesine kolayca erişilebilir.3 Bu, belirli alanlara veya stillere uygun sentetik veri üretmek için özellikle faydalıdır.  
  Örneğin, ürün açıklamaları oluşturmak için basit bir örnek verilebilir:  
  Python  
  from transformers import pipeline  
    
  generator = pipeline('text-generation', model='gpt2')  
  prompt = "Anahtar kelimeler: yüksek çözünürlük, 15 inç ekran, hafif. Ürün açıklaması:"  
  result = generator(prompt, max\_length=50, num\_return\_sequences=3)  
  for res in result:  
   print(res['generated\_text'])  
    
  Bu örnekte, verilen anahtar kelimelerden yola çıkarak kısa ürün açıklamaları üretilmektedir.  
  Resmi dokümantasyon bağlantısı: <https://huggingface.co/docs/transformers/>
* **2.2. OpenAI Python Kütüphanesi** 17  
  OpenAI Python kütüphanesi, metin üretimi ve diğer NLP görevleri için OpenAI'nin güçlü dil modellerine (GPT-3.5, GPT-4 vb.) bir arayüz sağlar. Bu kütüphane, OpenAI API'si ile etkileşimi kolaylaştırarak, geliştiricilerin uygulamalarına son teknoloji ürünü yapay zeka modellerini entegre etmelerine olanak tanır.19  
  Kütüphanenin temel işlevleri arasında, promptlara dayalı olarak metin üretmek için kullanılan openai.ChatCompletion.create() ve openai.Completion.create() yöntemleri bulunur.19 openai.ChatCompletion.create() yöntemi, özellikle sohbet modelleri için tasarlanmıştır ve model, messages (kullanıcı ve sistem rol mesajlarının bir listesi), temperature (rastgelelik seviyesi), max\_tokens (maksimum çıktı uzunluğu) gibi önemli parametreleri içerir.19 openai.Completion.create() yöntemi ise, daha genel metin tamamlama görevleri için kullanılır ve benzer parametrelere sahiptir, ancak messages yerine doğrudan bir prompt parametresi alır.  
  Örneğin, bir hikaye oluşturmak için aşağıdaki gibi bir kod kullanılabilir:  
  Python  
  from openai import OpenAI  
  import os  
    
  client = OpenAI(api\_key=os.environ.get("OPENAI\_API\_KEY"))  
    
  response = client.chat.completions.create(  
   model="gpt-3.5-turbo",  
   messages=[  
   {"role": "user", "content": "Arkadaşlık hakkında 3 cümlelik bir hikaye oluştur."},  
   ],  
   temperature=0.7,  
   max\_tokens=100  
  )  
  print(response.choices.message.content)  
    
  OpenAI API'si, yapılandırılmış sentetik veri (örneğin, CSV formatında) oluşturmak için de kullanılabilir.3 LLM'lere CSV formatını, şemayı ve sütunlar arasındaki ilişkileri belirten yapılandırılmış promptlar sağlanarak, anlamlı ve tutarlı sentetik tablolar oluşturulabilir. OpenAI Cookbook, bu tür uygulamalar için pratik örnekler sunmaktadır.3  
  OpenAI API'si ayrıca, web araması gibi araçları kullanma ve fonksiyon çağırma gibi gelişmiş özellikler sunar.20 Bu özellikler, üretilen sentetik verinin gerçekçiliğini ve kullanışlılığını artırabilir. Örneğin, bir LLM'ye güncel bir olay hakkında sentetik bir haber makalesi yazması istendiğinde, web arama aracı kullanılarak en son bilgilere erişilebilir.  
  Resmi dokümantasyon bağlantısı: <https://platform.openai.com/docs/api-reference>
* **2.3. LangChain** 16  
  LangChain, LLM'ler tarafından desteklenen uygulamalar oluşturmak için bir çerçeve olarak tanımlanabilir ve karmaşık sentetik veri üretimi iş akışları oluşturmak için kullanılabilir.16 LangChain, birden fazla promptun zincirlenmesine, harici veri kaynaklarının kullanılmasına ve daha karmaşık sentetik veri üretimi süreçleri oluşturmak için farklı bileşenlerin entegre edilmesine olanak tanır.29  
  Tipik bir LangChain tabanlı sentetik veri üretimi süreci, belge parçalama (document chunking), bağlam oluşturma (context generation), sorgu oluşturma (query generation) ve veri evrimi (data evolution) gibi adımları içerir.29 Belge parçalama, büyük belgeleri daha küçük, anlamlı parçalara ayırarak bağlamın korunmasına yardımcı olur. Bağlam oluşturma aşamasında, ilgili bilgileri bulmak için rastgele bir veri parçası seçilir ve benzerlik algoritmaları kullanılarak ilişkili parçalar gruplandırılır. Sorgu oluşturma adımında, oluşturulan bağlam kullanılarak bir LLM aracılığıyla bir dizi soru veya ifade üretilir. Veri evrimi ise, üretilen sorguların karmaşıklığını ve gerçekçiliğini artırmak için çeşitli şablonlar uygulanarak gerçekleştirilir.  
  LangChain iş akışlarında bilgi grafikleri veya benzerlik algoritmaları entegre edilerek daha gerçekçi ve çeşitli sentetik veri kümeleri oluşturma potansiyeli bulunmaktadır.29 LangChain'in modülerliği ve entegrasyon yetenekleri, basit LLM promptlamanın ötesine geçen, yüksek düzeyde özelleştirilmiş sentetik veri üretimi süreçlerinin oluşturulmasına olanak tanır.  
  Örneğin, belirli bir belgeye dayalı olarak sentetik soru-cevap çiftleri oluşturmak için yüksek düzeyli bir LangChain iş akışı şu şekilde olabilir: İlk olarak, belge parçalara ayrılır. Ardından, her parça için ilgili bağlamlar oluşturulur. Bu bağlamlar kullanılarak, LLM'den belge içeriğiyle ilgili sorular ve cevaplar üretmesi istenir. Son olarak, üretilen soru-cevap çiftleri veri evrimi teknikleriyle zenginleştirilebilir.  
  Resmi dokümantasyon bağlantısı: <https://www.langchain.com/>
* **2.4. Diğer İlgili Kütüphaneler** 16  
  Belirli sentetik veri üretimi senaryolarında ilgili olabilecek diğer Python kütüphaneleri şunlardır:
  + **ydata-synthetic**: GAN tabanlı modeller (CTGAN, TimeGAN) dahil olmak üzere çeşitli sentetik veri üretimi teknikleri sunar.31
  + **SDV (Synthetic Data Vault)**: Klasik makine öğrenimi ve derin öğrenme modellerini içeren çeşitli sentezleyiciler sağlar ve tek ve çok tablolu verileri destekler.31
  + **Mostly AI's Synthetic Data SDK**: Yüksek doğruluklu, gizlilik açısından güvenli sentetik veri üretmeye odaklanan açık kaynaklı bir araç kitidir.32
  + **NLTK (Natural Language Toolkit)** ve **SpaCy**: Öncelikli olarak sentetik veri üretimi için olmasa da, sentetik veri iş akışlarında metin verilerinin ön ve son işlenmesi için faydalı olabilirler.16
  + **SentenceTransformers**: Cümle gömülmeleri oluşturmak için kullanılabilir, bu da sentetik verinin semantik benzerliğini ve çeşitliliğini analiz etmek için kullanılabilir.16

Sentetik veri üretimi için çeşitli Python kütüphaneleri mevcuttur ve her birinin kendine özgü güçlü yönleri ve odak noktaları vardır. Kütüphane seçimi, veri türü (tablolu, zaman serisi, metin), istenen kontrol seviyesi ve performans değerlendirmeleri gibi görevin özel gereksinimlerine bağlıdır. LLM'ler metin üretimi için güçlü olsa da, diğer kütüphaneler farklı veri modaliteleri veya gizlilik koruma gibi belirli kalite özelliklerine yönelik özel teknikler sunar. Bu kütüphanelerin LLM tabanlı yaklaşımlarla entegre edilmesi, daha kapsamlı sentetik veri çözümlerine yol açabilir.

**3. LLM Tabanlı Sentetik Metin Veri Üretimi için GitHub Depoları**

* **3.1. Awesome-LLM-Synthetic-Data** ((<https://github.com/wasiahmad/Awesome-LLM-Synthetic-Data>)) 33  
  Bu depo, LLM'lerin sentetik verileri, LLM'ler tarafından sentetik veri üretimi ve LLM'ler için sentetik veriler hakkında makalelerin, araçların ve blogların derlenmiş bir listesidir.35 Depo, anketler, yöntemler, uygulama alanları (matematiksel muhakeme, kod üretimi, metin-SQL, hizalama vb.), veri kümeleri, araçlar ve bloglar gibi çeşitli kategorileri kapsamaktadır. Depoda listelenen, özellikle sentetik metin veri üretimiyle ilgili olan belirli teknikler ve uygulama alanları (örneğin, yüksek kaliteli/karmaşıklıkta talimat üretimi, hizalama) dikkat çekicidir. Bu depo, LLM tabanlı sentetik veri üretimi alanındaki en son araştırmaları, metodolojileri ve araçları takip etmek için değerli bir kaynak olarak hizmet vermektedir. Deponun kapsamlı yapısı, kullanıcıların ilgili akademik makaleleri, pratik araçları ve bilgilendirici blog gönderilerini hızlı bir şekilde keşfetmelerini sağlayarak, sentetik veri üretimi tekniklerini keşfetmelerini ve uygulamalarını kolaylaştırır.
* **3.2. LLM-Synthetic-Data** ((<https://github.com/pengr/LLM-Synthetic-Data>)) 33  
  Bu depo, LLM odaklı sentetik verilere odaklanan GitHub bağlantıları ve blog gönderileri dahil olmak üzere bir başka kaynak koleksiyonudur.33 Depo, LLM'lerin sentetik veri üretimi bağlamını ve zorluklarını anlamak için ilgili olan ön eğitim, talimat ayarlama, model çöküşü ve uzun bağlam gibi konuları içermektedir. Depoda bahsedilen, sentetik metin veri üretimi hakkında pratik rehberlik veya içgörü sağlayan belirli blog gönderileri veya kaynaklar bulunmaktadır. Önceki depoya benzer şekilde, bu koleksiyon da LLM tabanlı sentetik veri üretimi hakkında kaynak arayan araştırmacılar ve uygulayıcılar için değerli bir başlangıç noktası sunmaktadır. İlgili bağlantıları ve makaleleri bir araya getirerek, depo kullanıcıların bilgi arama çabalarını azaltır ve alanındaki temel kaynaklara merkezi bir erişim noktası sağlar.
* **3.3. Zhen-Tan-dmml/LLM4Annotation** ((<https://github.com/Zhen-Tan-dmml/LLM4Annotation>)) 36  
  Bu depo, LLM'leri açıklama (annotation) için kullanmaya odaklandığı için potansiyel olarak ilgili olabilir ve bu, LLM'leri veri oluşturmak için kullanma açısından sentetik veri üretimiyle benzerlikler taşır.36 Deponun belirli yönleri (örneğin, teknikler, betikler), sentetik metin veri üretimi için uyarlanabilir veya bilgilendirici olabilir. Birincil odak noktası açıklama olsa da, bu depoda LLM'leri veri oluşturma için promptlama ve yönlendirme ile ilgili teknikler ve içgörüler, sentetik veri üretimi görevlerine aktarılabilir. LLM'lerden belirli veri türlerini elde etme zorlukları, hem açıklama hem de sentetik veri üretimi için ortaktır. İlgili görevlere odaklanan depoları keşfetmek, değerli stratejiler ve yaklaşımlar ortaya çıkarabilir.
* **3.4. Kiln-AI/Kiln** (<https://github.com/Kiln-AI/Kiln>) 37  
  Bu depo, LLM'leri potansiyel olarak kullanan etkileşimli bir sentetik veri üretimi özelliği içeren bir araç olarak tanımlanmaktadır.37 Depoda bahsedilen, sentetik metin veri üretimiyle ilgili belirli işlevler veya özellikler bulunmaktadır. Kiln gibi, açıkça sentetik veri üretimi yetenekleri sunan araçlar, genel amaçlı kütüphaneler kullanarak sentetik veri üretimi iş akışlarını sıfırdan oluşturmak yerine, LLM'leri kullanarak sentetik veri kümeleri oluşturmak için daha kullanıcı dostu bir arayüz ve önceden oluşturulmuş işlevler sağlayabilir.
* **3.5. SciPhi-AI/synthesizer** ((<https://github.com/SciPhi-AI/synthesizer>)) 37  
  Bu depo, sentetik veriyi mevcut verilere dayandırmak için AgentSearch ile entegre edilebilen bir sentezleyici aracı olarak tanımlanmaktadır.37 Mevcut verilere dayandırma yeteneği, daha gerçekçi ve bağlamsal olarak ilgili sentetik metin üretmek için faydalı olabilir. Sentetik veri üretimini gerçek dünya bilgileriyle dayandırma yeteneği, LLM tabanlı sentetik veri oluşturmada temel bir zorluk olan, üretilen metnin kalitesini ve doğruluğunu önemli ölçüde artırabilir. LLM'ler bazen yanlış veya anlamsız bilgiler üretebilir (halüsinasyonlar). Bu tür sentezleyiciler, onlara mevcut veri kümelerinden ilgili bağlam sağlayarak bu sorunu hafifletmeye yardımcı olabilir.
* **3.6. Open R1** (<https://github.com/grpo727/open_r1>) 38  
  Bu depo, DeepSeek-R1'i yeniden üretmeye odaklanmakta ve Distilabel tekniğini kullanarak sentetik veri üretimi için betikler içermektedir.38 Bu tür araştırma odaklı depolardan gelişmiş sentetik veri üretimi tekniklerini öğrenme potansiyeli bulunmaktadır. En son modelleri ve teknikleri yeniden üretmeyi amaçlayan depoları incelemek, LLM'leri kullanarak sentetik veri üretimi için en son yaklaşımlar hakkında bilgi sağlayabilir. Araştırmacılar genellikle model performansını artırmak için yeni sentetik veri üretimi yöntemleri denemektedir. Onların uygulamalarını keşfetmek, değerli öğrenme fırsatları ve belirli sentetik veri zorlukları için potansiyel çözümler sunabilir.
* **3.7. mlabonne/llm-datasets** (<https://github.com/mlabonne/llm-datasets>) 39  
  Bu depo, bazıları sentetik olarak oluşturulmuş olabilecek bir LLM veri kümeleri koleksiyonu olarak tanımlanmaktadır.39 Özellikle sentetik olarak işaretlenmiş veya LLM tabanlı yöntemler kullanılarak oluşturulmuş ve örnek veya kıyaslama olarak hizmet edebilecek veri kümeleri bulunmaktadır. Mevcut sentetik veri kümelerini keşfetmek, oluşturulan veri türleri, kullanılan yöntemler ve potansiyel uygulamalar hakkında bilgi sağlayabilir. Herkese açık sentetik veri kümelerini analiz etmek, sentetik veri üretimi alanının mevcut durumunu anlamaya ve kendi görevleri için ilgili olabilecek veri kümelerini belirlemeye yardımcı olabilir.

**4. LLM Sentetik Veri Üretiminde Dikkat Edilmesi Gerekenler ve Teknikler**

* **4.1. Prompt Mühendisliği** 1  
  İyi tasarlanmış promptlar, LLM'leri istenen sentetik metni üretmeye yönlendirmede çok önemli bir rol oynar.1 Bağlam sağlama, formatı belirtme, örnekler verme (birkaç atış öğrenimi) ve persona tabanlı promptlar kullanma gibi teknikler bu süreçte önemlidir. Promptun kalitesi ve özgüllüğü, üretilen sentetik verinin kalitesini ve alaka düzeyini önemli ölçüde etkiler. Etkili prompt mühendisliği, istenen sonuçları elde etmek için gereklidir. LLM'ler, promptta sağlanan talimatlara ve bağlama göre yanıt verir. Görevi, istenen formatı ve belirli kısıtlamaları açıkça tanımlayan promptlar oluşturmak, faydalı sentetik veri üretmek için hayati öneme sahiptir.
* **4.2. Veri Kalitesi ve Çeşitliliği** 1  
  LLM tarafından üretilen sentetik verinin kalitesini, doğruluğunu ve çeşitliliğini sağlama zorlukları ele alınmalıdır. Halüsinasyonlar, stilistik gerçekçilik eksikliği ve doğal önyargılar gibi potansiyel sorunlar göz önünde bulundurulmalıdır.1 Veri kalitesini iyileştirmek için filtreleme, çıktıları ağırlıklandırma, bilgiye erişimli üretim ve yinelemeli kendini iyileştirme gibi teknikler tartışılmalıdır.1 Farklı promptlama teknikleri kullanma, üretim parametrelerini değiştirme ve kümeleme tabanlı yaklaşımlar kullanma gibi çeşitliliği artırma stratejileri açıklanmalıdır.3 Sentetik verinin etkinliği için veri kalitesi ve çeşitliliği arasında bir denge kurmak önemlidir. Aşırı homojen veya yanlış sentetik veri, model eğitimi veya değerlendirmesi için istenen faydaları sağlamayabilir. LLM'ler çeşitli metinler üretebilse de, üretilen verinin yüksek kalitede olması ve gerçek dünya verilerinin özelliklerini doğru bir şekilde yansıtması önemlidir. Çıktıyı filtreleme, iyileştirme ve çeşitlendirme teknikleri gereklidir.
* **4.3. Önyargıları Ele Alma** 1  
  Önyargılı gerçek dünya verileri üzerinde eğitilmiş LLM'ler tarafından üretilen sentetik verilerde önyargıların artma riski vurgulanmalı ve bu önyargıları azaltma yöntemleri tartışılmalıdır.1 Sentetik veri, LLM'nin eğitim verilerinde bulunan önyargıları istemeden devralabilir ve hatta artırabilir. Adalet ve zararlı sonuçları önlemek için dikkatli değerlendirme ve azaltma stratejileri gereklidir. LLM'ler eğitildikleri verilerden öğrenirler. Bu veriler cinsiyet, ırk veya diğer hassas özelliklerle ilgili önyargılar içeriyorsa, model tarafından üretilen sentetik veri de bu önyargıları yansıtabilir. Bu sorunun farkında olmak ve sentetik veriyi önyargılardan arındırmak için teknikler kullanmak önemlidir.
* **4.4. Sentetik Verinin Değerlendirilmesi** 4  
  Sentetik verinin faydasını ve doğruluğunu değerlendirmenin önemi kısaca değinilmeli, kalitesini ve amaçlanan görev için uygunluğunu değerlendirmek için kullanılan metrikler ve yöntemler belirtilmelidir.4 Model eğitimi veya değerlendirmesi gibi kritik görevler için kullanmadan önce sentetik verinin kalitesini ve kullanışlılığını değerlendirmek esastır. Sentetik verinin gerekli standartları karşıladığından emin olmak için uygun metrikler ve değerlendirme yöntemleri kullanılmalıdır. Her sentetik veri eşit kalitede değildir. Gerçek dünya verilerine benzerliğini, aşağı akış model performansına etkisini ve istenen özellikleri temsil etme yeteneğini değerlendirmek, sentetik veri üretimi sürecinde çok önemli adımlardır.

**5. Sonuç**

Bu rapor, LLM tabanlı sentetik metin veri üretimi için değerli kaynaklar sağlayan temel Python kütüphanelerini (Hugging Face Transformers, OpenAI Python Kütüphanesi, LangChain ve diğerleri) ve GitHub depolarını özetlemektedir. Prompt mühendisliği gibi tekniklerin ve sentetik veri üretimi sürecinde veri kalitesi, çeşitliliği ve önyargı konularındaki hususların önemi vurgulanmıştır. LLM tabanlı sentetik metin veri üretiminin, veri kıtlığını gidermek, maliyetleri düşürmek ve NLP modellerinin performansını ve dayanıklılığını artırmak için taşıdığı potansiyel yeniden teyit edilmiştir. Sonuç olarak, bu alanın yapay zeka alanındaki sürekli ilerlemeler ve artan önemiyle birlikte geleceğe yönelik umut verici bir görünüm sunmaktadır.

**Tablo 1: LLM Tabanlı Sentetik Metin Veri Üretimi için Temel Python Kütüphanelerinin Özeti**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kütüphane Adı** | **Açıklama** | **Sentetik Veri Üretimi için Temel Özellikler** | **Bağlantı** |
| Hugging Face Transformers | Önceden eğitilmiş LLM'ler ve NLP görevleri için araçlar sunan merkezi kütüphane. | generate() metodu, çeşitli kod çözme stratejileri, GenerationConfig ile özelleştirme. | <https://huggingface.co/docs/transformers/> |
| OpenAI Python Kütüphanesi | OpenAI'nin dil modellerine (GPT-3.5, GPT-4 vb.) arayüz sağlar. | openai.ChatCompletion.create() ve openai.Completion.create() metotları, çeşitli parametrelerle kontrol. | <https://platform.openai.com/docs/api-reference> |
| LangChain | LLM'ler tarafından desteklenen uygulamalar oluşturmak için çerçeve. | Prompt zincirleme, harici veri kaynakları entegrasyonu, belge parçalama, bağlam ve sorgu oluşturma. | <https://www.langchain.com/> |
| ydata-synthetic | GAN tabanlı modeller dahil çeşitli sentetik veri üretimi teknikleri sunar. | CTGAN, TimeGAN, GMM tabanlı modeller. | <https://ydata.ai/solutions/synthetic-data> |
| SDV (Synthetic Data Vault) | Klasik ML ve derin öğrenme modellerini içeren çeşitli sentezleyiciler sağlar. | Tek ve çok tablolu veri desteği, anonimleştirme kontrolü, kısıtlama uygulama. | <https://sdv.dev/> |

**Tablo 2: LLM Tabanlı Sentetik Metin Veri Üretimi için Temel GitHub Depolarının Özeti**

|  |  |  |  |
| --- | --- | --- | --- |
| **Depo Adı** | **Açıklama** | **Temel Özellikler/Önemli Noktalar** | **Bağlantı** |
| Awesome-LLM-Synthetic-Data | LLM'lerin sentetik verileri hakkında derlenmiş makaleler, araçlar ve bloglar listesi. | Anketler, yöntemler, uygulama alanları, veri kümeleri, araçlar ve bloglar. | (https://github.com/wasiahmad/Awesome-LLM-Synthetic-Data) |
| LLM-Synthetic-Data | LLM odaklı sentetik verilere odaklanan kaynak koleksiyonu. | GitHub bağlantıları, blog gönderileri, ön eğitim, talimat ayarlama konuları. | (https://github.com/pengr/LLM-Synthetic-Data) |
| Zhen-Tan-dmml/LLM4Annotation | LLM'leri açıklama için kullanmaya odaklanır. | LLM'leri veri oluşturma için promptlama ve yönlendirme teknikleri. | (https://github.com/Zhen-Tan-dmml/LLM4Annotation) |
| Kiln-AI/Kiln | Etkileşimli bir sentetik veri üretimi özelliği içeren bir araç. | Kullanıcı dostu arayüz, sentetik veri üretimi için önceden oluşturulmuş işlevler. | <https://github.com/Kiln-AI/Kiln> |
| SciPhi-AI/synthesizer | Sentetik veriyi mevcut verilere dayandırmak için AgentSearch ile entegre edilebilir. | Sentetik veri üretimini gerçek dünya bilgileriyle dayandırma yeteneği. | (https://github.com/SciPhi-AI/synthesizer) |
| Open R1 | DeepSeek-R1'i yeniden üretmeye odaklanır. | Sentetik veri üretimi için Distilabel tekniğini kullanan betikler. | <https://github.com/grpo727/open_r1> |
| mlabonne/llm-datasets | Bazıları sentetik olarak oluşturulmuş LLM veri kümeleri koleksiyonu. | Sentetik olarak işaretlenmiş veya LLM tabanlı yöntemlerle oluşturulmuş veri kümeleri. | <https://github.com/mlabonne/llm-datasets> |

#### Alıntılanan çalışmalar

1. [2503.14023] Synthetic Data Generation Using Large Language Models: Advances in Text and Code - arXiv, erişim tarihi Nisan 10, 2025, <https://arxiv.org/abs/2503.14023>
2. Large language models generating synthetic clinical datasets: a feasibility and comparative analysis with real-world perioperative data - Frontiers, erişim tarihi Nisan 10, 2025, <https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2025.1533508/full>
3. Synthetic data generation (Part 1) | OpenAI Cookbook, erişim tarihi Nisan 10, 2025, <https://cookbook.openai.com/examples/sdg1>
4. LLM-Driven Synthetic Data Generation, Curation & Evaluation | by ..., erişim tarihi Nisan 10, 2025, <https://cobusgreyling.medium.com/llm-driven-synthetic-data-generation-curation-evaluation-33731e33b525>
5. Synthetic Data Generation Using Large Language Models: Advances in Text and Code, erişim tarihi Nisan 10, 2025, <https://arxiv.org/html/2503.14023v1>
6. Synthetic Data Generator Simplifies Dataset Creation with Large Language Models - InfoQ, erişim tarihi Nisan 10, 2025, <https://www.infoq.com/news/2025/01/synthetic-data-generator/>
7. What is Text Generation? - Hugging Face, erişim tarihi Nisan 10, 2025, <https://huggingface.co/tasks/text-generation>
8. Generation strategies - Hugging Face, erişim tarihi Nisan 10, 2025, <https://huggingface.co/docs/transformers/v4.27.0/generation_strategies>
9. Text generation - Hugging Face, erişim tarihi Nisan 10, 2025, <https://huggingface.co/docs/transformers/llm_tutorial>
10. Generating text word by word - Transformers - Hugging Face Forums, erişim tarihi Nisan 10, 2025, <https://discuss.huggingface.co/t/generating-text-word-by-word/29312>
11. Generation - Hugging Face, erişim tarihi Nisan 10, 2025, <https://huggingface.co/docs/transformers/main_classes/text_generation>
12. Text Generation with Hugging Face Transformers: A Beginner's Guide | by Ganesh Lokare, erişim tarihi Nisan 10, 2025, <https://medium.com/@lokaregns/text-generation-with-hugging-face-transformers-a-beginners-guide-6b0b4b957379>
13. How to generate text: using different decoding methods for language generation with Transformers - Hugging Face, erişim tarihi Nisan 10, 2025, <https://huggingface.co/blog/how-to-generate>
14. How to generate text using GPT2 model with Huggingface transformers? - Stack Overflow, erişim tarihi Nisan 10, 2025, <https://stackoverflow.com/questions/76663419/how-to-generate-text-using-gpt2-model-with-huggingface-transformers>
15. Generating text while model is still training - Beginners - Hugging Face Forums, erişim tarihi Nisan 10, 2025, <https://discuss.huggingface.co/t/generating-text-while-model-is-still-training/57482>
16. 10 Must-Know Python Libraries for LLMs in 2025 - MachineLearningMastery.com, erişim tarihi Nisan 10, 2025, <https://machinelearningmastery.com/10-must-know-python-libraries-for-llms-in-2025/>
17. Top 50 Python Libraries to Know in 2025 - Analytics Vidhya, erişim tarihi Nisan 10, 2025, <https://www.analyticsvidhya.com/blog/2024/12/python-libraries/>
18. 8 Top Open-Source LLMs for 2024 and Their Uses - DataCamp, erişim tarihi Nisan 10, 2025, <https://www.datacamp.com/blog/top-open-source-llms>
19. How can I use OpenAI for text generation? - Milvus, erişim tarihi Nisan 10, 2025, <https://milvus.io/ai-quick-reference/how-can-i-use-openai-for-text-generation>
20. Developer quickstart - OpenAI API, erişim tarihi Nisan 10, 2025, <https://platform.openai.com/docs/quickstart>
21. Advanced usage - OpenAI API, erişim tarihi Nisan 10, 2025, <https://platform.openai.com/docs/advanced-usage>
22. How to Generate Text with OpenAI, GPT-3, and Python - Matt on ML.NET - Accessible AI, erişim tarihi Nisan 10, 2025, <https://accessibleai.dev/post/generating_text_with_gpt_and_python/>
23. Building a Tool To Generate Text With OpenAI's GPT-4 Model - DZone, erişim tarihi Nisan 10, 2025, <https://dzone.com/articles/building-a-tool-to-generate-text-with-gpt4>
24. data-winners/generation-api-openai/openai-text-generation-examples-in-python.ipynb at main - GitHub, erişim tarihi Nisan 10, 2025, <https://github.com/FrontAnalyticsInc/data-winners/blob/main/generation-api-openai/openai-text-generation-examples-in-python.ipynb>
25. Text Generation: I'm struggling with Output Consistency - API - OpenAI Developer Forum, erişim tarihi Nisan 10, 2025, <https://community.openai.com/t/text-generation-im-struggling-with-output-consistency/539537>
26. GPT-4 API Guide With Examples (All Scripts Included), erişim tarihi Nisan 10, 2025, <https://hasanaboulhasan.medium.com/gpt-4-api-guide-with-examples-all-scripts-included-afa1b6a5c3d9>
27. OpenAI Chat Completion API - OpenAI Platform, erişim tarihi Nisan 10, 2025, <https://platform.openai.com/docs/guides/text-generation>
28. Text-Generation App with OpenAI Python API & Streamlit - YouTube, erişim tarihi Nisan 10, 2025, <https://www.youtube.com/watch?v=WPlSpjiv4tk>
29. Using LLMs for Synthetic Data Generation: The Definitive Guide ..., erişim tarihi Nisan 10, 2025, <https://www.confident-ai.com/blog/the-definitive-guide-to-synthetic-data-generation-using-llms>
30. Generating Synthetic Dataset Using LLM - DeepLearning.AI, erişim tarihi Nisan 10, 2025, <https://community.deeplearning.ai/t/generating-synthetic-dataset-using-llm/737049>
31. The Top 5 Python Packages to Generate Realistic Synthetic Data - YData, erişim tarihi Nisan 10, 2025, <https://ydata.ai/resources/top-5-packages-python-synthetic-data>
32. The Synthetic Data SDK - An open-source python toolkit for high-fidelity privacy-safe ... - Mostly AI, erişim tarihi Nisan 10, 2025, <https://mostly.ai/blog/the-synthetic-data-sdk-an-open-source-python-toolkit-for-high-fidelity-privacy-safe-synthetic-data>
33. Real-time updated, fine-grained reading list on LLM-synthetic-data. - GitHub, erişim tarihi Nisan 10, 2025, <https://github.com/pengr/LLM-Synthetic-Data>
34. README.md - pengr/LLM-Synthetic-Data - GitHub, erişim tarihi Nisan 10, 2025, <https://github.com/pengr/LLM-Synthetic-Data/blob/main/README.md>
35. wasiahmad/Awesome-LLM-Synthetic-Data - GitHub, erişim tarihi Nisan 10, 2025, <https://github.com/wasiahmad/Awesome-LLM-Synthetic-Data>
36. Zhen-Tan-dmml/LLM4Annotation - GitHub, erişim tarihi Nisan 10, 2025, <https://github.com/Zhen-Tan-dmml/LLM4Annotation>
37. In 2024, what is the best tool/framework for creating synthetic data that can then be used to fine-tune with? - Reddit, erişim tarihi Nisan 10, 2025, <https://www.reddit.com/r/LocalLLaMA/comments/194m01m/in_2024_what_is_the_best_toolframework_for/>
38. Top GitHub Projects of Jan 2025 - OpenCV, erişim tarihi Nisan 10, 2025, <https://opencv.org/blog/top-github-projects-jan-2025/>
39. mlabonne/llm-datasets: Curated list of datasets and tools for post-training. - GitHub, erişim tarihi Nisan 10, 2025, <https://github.com/mlabonne/llm-datasets>
40. Synthetic Data Generation with Large Language Models for Text Classification: Potential and Limitations | OpenReview, erişim tarihi Nisan 10, 2025, [https://openreview.net/forum?id=MmBjKmHIND&referrer=%5Bthe%20profile%20of%20Zhuoyan%20Li%5D(%2Fprofile%3Fid%3D~Zhuoyan\_Li2)](https://openreview.net/forum?id=MmBjKmHIND&referrer=%5Bthe+profile+of+Zhuoyan+Li%5D(/profile?id%3D~Zhuoyan_Li2))
41. Synthetic Data Generation with Large Language Models for Text Classification: Potential and Limitations - ACL Anthology, erişim tarihi Nisan 10, 2025, <https://aclanthology.org/2023.emnlp-main.647.pdf>
42. Few-shot LLM Synthetic Data with Distribution Matching - arXiv, erişim tarihi Nisan 10, 2025, <https://arxiv.org/html/2502.08661v1>
43. Data Generation Using Large Language Models for Text Classification: An Empirical Case Study - arXiv, erişim tarihi Nisan 10, 2025, <https://arxiv.org/html/2407.12813v2>
44. [2310.07849] Synthetic Data Generation with Large Language Models for Text Classification: Potential and Limitations - arXiv, erişim tarihi Nisan 10, 2025, <https://arxiv.org/abs/2310.07849>
45. On LLMs-Driven Synthetic Data Generation, Curation, and Evaluation: A Survey - arXiv, erişim tarihi Nisan 10, 2025, <https://arxiv.org/html/2406.15126v1>

Elbette, LLM tabanlı sentetik metin veri üretimi konusunda öne çıkan tüm önemli GitHub depolarının kapsamlı bir özetini hazırlayacağım. Bu özet; açıklama, kullanım alanı, desteklenen modeller, aktiflik durumu ve bağlantıları içerecek şekilde yapılandırılacak. Popüler, aktif, niş ve deneysel projeleri de dahil edeceğim. Sonuçlar hazır olduğunda seninle paylaşacağım.

# LLM Tabanlı Sentetik Metin Veri Üretimi: Önemli GitHub Projeleri

Büyük dil modelleri (LLM) kullanarak sentetik metin veri setleri oluşturmayı kolaylaştıran çeşitli açık kaynak projeler mevcuttur. Aşağıda, hem popüler/aktif projeler hem de niş veya deneysel ama ilgi çekici projeler ayrı başlıklar altında özetlenmiştir. Her biri için proje amacı, üretebildiği veri türleri, desteklediği LLM’ler, kullanım kolaylığı (örnekler) ile güncellik durumu ve GitHub bağlantısı verilmiştir.

## Argilla – Distilabel & Synthetic Data Generator

([GitHub - argilla-io/synthetic-data-generator: Build datasets using natural language](https://github.com/argilla-io/synthetic-data-generator)) Argilla Synthetic Data Generator arayüzü, kullanıcıların ***doğal dil*** ile veri kümesi tanımlamalarına ve ***LLM***’ler aracılığıyla örnekler üretmesine olanak tanır. Arayüzde istenen veri kümesinin tanımı girilir, sistem bir ***talimat (prompt)*** ve isteğe bağlı bağlam bilgisi alarak belirtilen sayıda sentetik veri örneği üretir. Oluşan veri, Argilla veya Hugging Face Hub’a aktarılabilir.

**Açıklama:** Argilla, metin veri kümelerini etiketleme ve yönetme platformuyla tanınır. Bu platformun bir parçası olan **Distilabel** kütüphanesi ve **Synthetic Data Generator** aracı, LLM’ler yardımıyla kullanıcı tanımına uygun sentetik veri üretmeye odaklanır. Araç, literatürde önerilen **Self-Instruct** ve **Evol-Instruct** gibi teknikleri destekleyecek esneklikte tasarlanmıştır ([GitHub - davanstrien/awesome-synthetic-datasets: awesome synthetic (text) datasets](https://github.com/davanstrien/awesome-synthetic-datasets#:~:text=,data%20ownership%2C%20and%20overall%20efficiency)). Kullanıcı, üretmek istediği veri setinin özelliklerini **doğal dilde tarif ederek** kısa sürede örnekler oluşturabilir.

**Üretilen Veri Tipleri:** Metin sınıflandırma için etiketli cümleler, sohbet/diyalog verileri (talimat-yanıt çiftleri) ve **Retrieval-Augmented Generation (RAG)** senaryoları için soru-cevap/veri çiftleri gibi çeşitli veri türlerini destekler ([GitHub - argilla-io/synthetic-data-generator: Build datasets using natural language](https://github.com/argilla-io/synthetic-data-generator#:~:text=Supported%20Tasks%3A)). Örneğin, sınıflandırma görevleri için farklı kategorilerde cümle örnekleri ya da sohbet modelleri için kullanıcı talimatları ve asistan cevapları üretebilir.

**Desteklenen LLM’ler:** Argilla’nın aracı, birden fazla LLM sağlayıcısını destekleyecek şekilde esnektir. OpenAI API (GPT-3.5, GPT-4 vb.), Hugging Face Inference Endpoint’leri, yerel **Ollama** veya **vLLM** sunucuları gibi farklı modeller/servisler entegre edilebilir ([GitHub - argilla-io/synthetic-data-generator: Build datasets using natural language](https://github.com/argilla-io/synthetic-data-generator#:~:text=,http%3A%2F%2F127.0.0.1%3A11434)). Varsayılan olarak, Hugging Face üzerinden açık modeller (örn. Llama türevleri) veya OpenAI’nin büyük modelleri kullanılabilir. Bu sayede hem bulut tabanlı hem yerel LLM’lerle çalışmak mümkün olur.

**Kullanım & Örnek:** Aracın kurulumu pip install synthetic-dataset-generator ile yapılabilir ([GitHub - argilla-io/synthetic-data-generator: Build datasets using natural language](https://github.com/argilla-io/synthetic-data-generator#:~:text=Installation)). Python API’si ile launch() fonksiyonu çağrılarak bir Gradio arayüzü başlatılır ([GitHub - argilla-io/synthetic-data-generator: Build datasets using natural language](https://github.com/argilla-io/synthetic-data-generator#:~:text=Quickstart)). Arayüz üzerinden üç adımda veri kümesi oluşturulur: (1) **İstediğiniz veri setini tanımlayın:** Örneğin “Python ile matematik bulmacaları çözen bir asistan” gibi bir açıklama girilir. (2) **Veri setini yapılandırın:** Gerekirse modelin rolü için sistem mesajı ve birkaç örnek belirtilebilir. (3) **Veri setini üretin:** Kaç adet örnek istenildiği girilerek üretim başlatılır ve sonuçlar görüntülenir (yandaki görselde görüldüğü gibi) ([GitHub - argilla-io/synthetic-data-generator: Build datasets using natural language](https://github.com/argilla-io/synthetic-data-generator)). Üretilen veri, tek tuşla Argilla platformuna veya Hugging Face Hub’a gönderilebilmektedir. Bu aracın kullanımı için teknik bilgi gerekmeden, arayüz sayesinde kolayca sentetik veri oluşturmak mümkündür.

**Güncellik & Aktiflik:** Distilabel kütüphanesi ve Argilla Sentetik Veri aracı 2024 itibariyle aktiftir ve geliştirilmeye devam edilmektedir ([GitHub - davanstrien/awesome-synthetic-datasets: awesome synthetic (text) datasets](https://github.com/davanstrien/awesome-synthetic-datasets#:~:text=This%20is%20a%20very%20flexible,such%20as%20%2084%20Self)). Argilla ekibi ve açık kaynak katkıcıları projeyi güncel tutmaktadır. Argilla’nın GitHub deposunda düzenli güncellemeler ve örnekler yer almaktadır.

**GitHub:** Distilabel kütüphanesi [argilla-io/distilabel](https://github.com/argilla-io/distilabel), arayüz aracı [argilla-io/synthetic-data-generator](https://github.com/argilla-io/synthetic-data-generator).

## Kiln – Veri Üretimi ve İnce Ayar Platformu

**Açıklama:** **Kiln**, uçtan uca bir LLM prototipleme ve veri işbirliği aracıdır. Hem **sentetik veri üretimi** hem de LLM’lerin **fine-tuning (ince ayar)** işlemlerini kullanıcı dostu bir şekilde bir araya getirmeyi hedefler. Masaüstü uygulamaları (Windows, Mac, Linux) üzerinden çalışan sezgisel bir arayüze sahiptir ([GitHub - Kiln-AI/Kiln: The easiest tool for fine-tuning LLM models, synthetic data generation, and collaborating on datasets.](https://github.com/Kiln-AI/Kiln#:~:text=Key%20Features)). Geliştirici olmayan ekiplerle birlikte veri kümeleri üzerinde çalışmayı kolaylaştırır ve süreç içinde model çıktıları ile insan geri bildirimlerini bir araya toplar.

**Üretilen Veri Tipleri:** Kiln içerisinde kullanıcılar çeşitli amaçlar için veri üretebilir. Örneğin, **talimat-yanıt çiftleri**, sohbet diyalogları, soru-cevap veri setleri, özetleme örnekleri veya sınıflandırma için örnek cümleler oluşturmak mümkündür. Araç, bir **etkileşimli arayüz** sunduğundan, kullanıcı hangi türde veriye ihtiyaç duyuyorsa onu tarif ederek uygun formatta sentetik veri üretimini başlatabilir. Bu sayede Q&A, diyalog, duygu analizi gibi değişik görevler için veri sentezlenebilir.

**Desteklenen LLM’ler:** Kiln, arka planda farklı model ve API sağlayıcıları ile çalışabilir. **OpenAI** gibi bulut API’leri ile entegre olup GPT-3.5/4 modellerini kullanabildiği gibi, yerel olarak **Unsloth** altyapısıyla (yerel LLM çalıştırma aracı) da çalışabilir ([In 2024, what is the best tool/framework for creating synthetic data that can then be used to fine-tune with? : r/LocalLLaMA](https://www.reddit.com/r/LocalLLaMA/comments/194m01m/in_2024_what_is_the_best_toolframework_for/#:~:text=Late%20addition%2C%20but%20I%20created,AI%2FKiln)). Ayrıca kendi belgelendirmesinde “herhangi bir AI modelini veya sağlayıcısını kullanma” yöntemleri sunulmuştur ([GitHub - Kiln-AI/Kiln: The easiest tool for fine-tuning LLM models, synthetic data generation, and collaborating on datasets.](https://github.com/Kiln-AI/Kiln#:~:text=,model%20or%20provider%20in%20Kiln)), bu da Hugging Face modelleri veya diğer üçüncü parti LLM servislerinin de adapte edilebileceğini gösterir. Yani hem bulutta hem yerelde geniş bir LLM desteği mevcuttur.

**Kullanım & Örnek:** Kiln’in en dikkat çekici yanı, **tek tıkla kurulabilen masaüstü uygulaması** şeklinde sunulmasıdır. Teknik detaylara girmeden, uygulama açıldıktan sonra bir proje oluşturup ihtiyaç duyulan veri tipini seçebilir veya tarif edebilirsiniz. Örneğin, bir kullanıcı dostu sihirbaz ile “müşteri destek sohbet verisi” üretmek istediğinizde, Kiln size birkaç örnek sorup ona göre birçok sentetik diyalog oluşturabilir. Oluşan veri kümesi üzerinde arayüzden inceleme yapabilir, gerekirse insan geri bildirimiyle düzenleyip doğrudan aynı ortamda bir modeli bu verilerle eğitebilirsiniz. İnce ayar için yine arayüzden model ve parametre seçimleri yapılarak süreç başlatılır; Kiln, bu süreci arka planda otomatikleştirir. Ayrıca Python kütüphanesi olarak da kullanılabilir (pip install kiln ile), böylece kod üzerinden de veri üretimi ve model eğitimi yaptırılabilir.

**Güncellik & Aktiflik:** Kiln projesi 2024 yılı itibarıyla yenidir ve aktiftir. Geliştiricisi projeyi Reddit gibi platformlarda tanıtmış ve açık kaynak olarak GitHub’da yayınlamıştır. Kullanıcı topluluğundan gelen geri bildirimlerle sık sık güncellenmesi muhtemeldir. Masaüstü uygulamaları düzenli olarak iyileştiriliyor ve dokümantasyon ile video rehberleri mevcuttur ([GitHub - Kiln-AI/Kiln: The easiest tool for fine-tuning LLM models, synthetic data generation, and collaborating on datasets.](https://github.com/Kiln-AI/Kiln#:~:text=Docs%20%26%20Guides)).

**GitHub:** [Kiln-AI/Kiln](https://github.com/Kiln-AI/Kiln) – “The easiest tool for fine-tuning LLM models, synthetic data generation, and collaborating on datasets.”

## Fuxion – LLM Tabanlı Veri Oluşturma ve Normalizasyon Kitaplığı

**Açıklama:** **Fuxion**, Python tabanlı bir kütüphane olarak **LangChain** ve LLM’leri kullanarak sentetik veri üretimi yapar ve çıktıları istenen forma normalleştirir ([GitHub - Tobiadefami/fuxion: Sythetic data generation and normalization functions powered by LLMs](https://github.com/tobiadefami/fuxion#:~:text=fuxion)) ([GitHub - Tobiadefami/fuxion: Sythetic data generation and normalization functions powered by LLMs](https://github.com/tobiadefami/fuxion#:~:text=The%20,of%20how%20to%20use%20it)). Özellikle yapılandırılmış veya standart formatta çıktıya ihtiyaç duyulan durumlar için geliştirilmiştir. Örneğin adres, isim, fiyat gibi verileri belirli bir şablonda üretip temizlemek için kullanılabilir. Kullanıcı tarafından tanımlanan bir **şablon (prompt template)** ve birkaç örnek ile, istenen yapıda çok sayıda sentetik veri noktası oluşturabilir. Veri üretimi ve normalizasyon tek adımda gerçekleşir, böylece çıktılar tutarlı bir biçimde formatlanır.

**Üretilen Veri Tipleri:** Fuxion genel amaçlı tasarlanmıştır; doğru prompt şablonları verildiğinde farklı senaryolar için veri üretebilir. **Örnek:** İsim verileri (ad, soyad, unvan parçalarına ayrılmış) üretme, adresleri standart formata dönüştürme, belirli bir formatta Q&A çiftleri oluşturma veya özel bir JSON yapısında veri hazırlama gibi kullanım alanları gösterilmiştir ([GitHub - Tobiadefami/fuxion: Sythetic data generation and normalization functions powered by LLMs](https://github.com/tobiadefami/fuxion#:~:text=fuxion%20is%20a%20Python%20package,and%20watch%20how%20things%20unfold)) ([GitHub - Tobiadefami/fuxion: Sythetic data generation and normalization functions powered by LLMs](https://github.com/tobiadefami/fuxion#:~:text=output_structure%20%3D%20%7B%20,str)). Temelde, kullanıcı **çıktı veri yapısını** tanımlar (örn. JSON alanları) ve modelden bu yapıya uygun metinler üretmesi istenir. Bu sayede sınıflandırma için etiketli cümlelerden diyaloglara kadar pek çok türde veri, önceden tanımlanan şablona uygun şekilde üretilebilir.

**Desteklenen LLM’ler:** Fuxion şu an için OpenAI API tabanlı modellerle çalışmaya odaklanmıştır. Öntanımlı olarak **GPT-3.5-Turbo** ve **GPT-4** ile bunların varyantlarını desteklemektedir ([GitHub - Tobiadefami/fuxion: Sythetic data generation and normalization functions powered by LLMs](https://github.com/tobiadefami/fuxion#:~:text=Models%20supported)). Örneğin model\_name="gpt-4" veya model\_name="gpt-3.5-turbo" şeklinde seçilebiliyor. Dokümantasyonda “gpt-4o” ve “gpt-4o-mini” gibi modeller de listelenmiş – bunlar da OpenAI’nin farklı boyut/tür versiyonlarını ifade etmektedir. Henüz Hugging Face veya yerel LLM entegrasyonu doğrudan yoktur, ancak planlanmaktadır ([GitHub - Tobiadefami/fuxion: Sythetic data generation and normalization functions powered by LLMs](https://github.com/tobiadefami/fuxion#:~:text=fuxion%20is%20still%20a%20work,the%20most%20functional%20and%20reliable)). Geliştiriciler, gelecekte **yerel barındırılan LLM’ler** veya HuggingFace API’leriyle de uyumlu hale getirmek üzere çalışmalar yapmayı hedeflediklerini belirtmiştir.

**Kullanım & Örnek:** Fuxion kütüphanesi pip install fuxion ile kurulabilir ([GitHub - Tobiadefami/fuxion: Sythetic data generation and normalization functions powered by LLMs](https://github.com/tobiadefami/fuxion#:~:text=)). Kullanımda önce bir **şablon dosyası** ve **few-shot örnek dosyası** hazırlanır. Ardından Python kodunda DatasetPipeline sınıfı kullanılarak veri üretim süreci tanımlanır ([GitHub - Tobiadefami/fuxion: Sythetic data generation and normalization functions powered by LLMs](https://github.com/tobiadefami/fuxion#:~:text=from%20fuxion,rich%20import%20print)) ([GitHub - Tobiadefami/fuxion: Sythetic data generation and normalization functions powered by LLMs](https://github.com/tobiadefami/fuxion#:~:text=pipeline_chain%20%3D%20DatasetPipeline%28%20generator_template%3D,4o%22%2C%20cache%3DFalse%2C%20verbose%3DTrue%2C%20temperature%3D1.0)). Örneğin:

from fuxion.pipelines import DatasetPipeline

pipeline = DatasetPipeline(

generator\_template="templates/isim\_olusturucu.template",

few\_shot\_file="examples/isim\_ornekleri.json",

output\_structure={"ad": str, "soyad": str, "unvan": str},

dataset\_name="isim\_verisi",

k=100, # 100 örnek oluştur

model\_name="gpt-3.5-turbo",

)

result = pipeline.execute()

Yukarıdaki kod, belirlenen şablona göre 100 adet isim örneği üretip isim\_verisi.json dosyasına kaydedecektir ([GitHub - Tobiadefami/fuxion: Sythetic data generation and normalization functions powered by LLMs](https://github.com/tobiadefami/fuxion#:~:text=result%20%3D%20pipeline_chain)). Fuxion, ardışık şekilde hem üretimi hem de normalizasyonu yaptığı için çıktı verisi tanımlanan output\_structure yapısına uygun ve tutarlı olur ([GitHub - Tobiadefami/fuxion: Sythetic data generation and normalization functions powered by LLMs](https://github.com/tobiadefami/fuxion#:~:text=This%20pipeline%20generates%20a%20dataset,directory)). Kullanım örnekleri arasında **beşerlik isim listeleri**, formatlı adres listeleri gibi senaryolar bulunmaktadır.

**Güncellik & Aktiflik:** Fuxion görece yeni (2023 sonu itibariyle) ve henüz küçük ölçekli bir projedir (GitHub ~50 yıldız). “Work in progress” olarak belirtilmiştir ([GitHub - Tobiadefami/fuxion: Sythetic data generation and normalization functions powered by LLMs](https://github.com/tobiadefami/fuxion#:~:text=Future%20work%3A)) ancak açık kaynak katkılara açıktır. Geliştiricileri gelecekte daha fazla özellik eklemeyi planladığı için (ör. farklı LLM entegrasyonları) proje aktif sayılabilir. Şu an temel fonksiyonları çalışır durumdadır ve örneklerle belgelenmiştir.

**GitHub:** [Tobiadefami/fuxion](https://github.com/tobiadefami/fuxion)

## DataDreamer – Veriseti Oluşturma ve Eğitim İş Akışları Kütüphanesi

**Açıklama:** **DataDreamer** açık kaynaklı bir Python kütüphanesidir ve amacı LLM’leri kullanarak hızlı bir şekilde sentetik veri hazırlamak ve ardından bu verilerle model eğitimi gibi işlemleri tek bir iş akışında gerçekleştirmektir ([GitHub - davanstrien/awesome-synthetic-datasets: awesome synthetic (text) datasets](https://github.com/davanstrien/awesome-synthetic-datasets#:~:text=)). “Prompting” (modeli yönlendiren komutların yazılması), veri üretimi ve eğitim aşamalarını entegre bir şekilde sunar. Araç, araştırma makalelerinde önerilen teknikleri **verimli ve basit** bir arayüzle kullanıma geçirmeyi hedefler. Özetle, “veri hayal etme” konseptine uygun olarak modelden veri hayali kurmasını ve bunu gerçek bir veri kümesi olarak derlemesini sağlar.

**Üretilen Veri Tipleri:** DataDreamer prensipte herhangi bir metin tabanlı veri tipini üretebilir, çünkü esnek bir şekilde model yönlendirmeye olanak tanır. Örneğin, **talimat ve yanıt çiftleri**, soru-cevap veri noktaları, belirli bir görev için etiketlenmiş cümleler (sınıflandırma veya duygu analizi için), özetleme örnekleri veya hatta metinden SQL sorgusu eşleşmeleri gibi çeşitli görevler için veri oluşturmak mümkün. Bu çeşitlilik, kullanıcı tarafından sağlanan istemlere (prompt’lara) bağlı olarak sağlanır. Kütüphane içerisinde, yaygın sentetik veri üretim teknikleri (örneğin **Self-Instruct** yaklaşımı, **Evol-Instruct** varyasyonları vb.) uygulanabilir durumdadır ([GitHub - davanstrien/awesome-synthetic-datasets: awesome synthetic (text) datasets](https://github.com/davanstrien/awesome-synthetic-datasets#:~:text=This%20is%20a%20very%20flexible,such%20as%20%2084%20Self)).

**Desteklenen LLM’ler:** DataDreamer birden fazla LLM ile çalışabilecek şekilde tasarlanmıştır. “Birçok LLM sağlayıcısını destekler” şeklinde belgelendiği için ([GitHub - davanstrien/awesome-synthetic-datasets: awesome synthetic (text) datasets](https://github.com/davanstrien/awesome-synthetic-datasets#:~:text=This%20is%20a%20very%20flexible,such%20as%20%2084%20Self)), muhtemelen OpenAI’nin modelleri, HuggingFace üzerinden erişilebilen modeller veya belki Azure/OpenAI gibi platformları entegre edebilmektedir. Kütüphane, yüksek performans ve hız için API çağrılarını verimli şekilde yapacak yapıda olduğundan, **toplu istek gönderme (batching)** gibi özellikler sunarak büyük ölçekli veri üretimini hızlandırır. Bu da farklı modellerle çalışırken de verim sağlar. (Not: Spesifik desteklenen modeller listesi dokümantasyonda net verilmemiş, ancak esneklik vurgulandığından çok sayıda LLM API’sinde çalışabilir.)

**Kullanım & Örnek:** DataDreamer odak noktası olarak geliştiricilere yönelik bir kütüphanedir. Bu nedenle bir GUI’den ziyade Python ile entegre kullanılması beklenir. Örneğin, bir Python betiğinde DataDreamer’ın API’ını çağırarak bir **prompt seti** üzerinden veri üretimi tetiklenebilir, ardından çıkan veri yine DataDreamer içinden doğrudan bir modelin fine-tuning işlemi için kullanılabilir. Belgelerde basit ve anlaşılır bir arayüz sunulduğu belirtiliyor; muhtemelen birkaç satır komutla **LLM’den belirli formatta bir veri yığını üret** ve **bunu dosyaya kaydet** gibi fonksiyonları vardır. “Research-grade” (araştırma kalitesinde) olması, deneysel çalışmalar için detaylı ayar imkânları sunduğunu ve çıktıları ürettikten sonra farklı modellere eğitim yapma konusunda esneklik sağladığını gösterir. Örnek komutlar için resmi README incelenebilir.

**Güncellik & Aktiflik:** DataDreamer, 2024 yılında akademik bir çalışma (ACL 2024) olarak duyurulmuştur ve Colin Raffel gibi isimlerin katkısıyla geliştirilmiştir ([GitHub - wasiahmad/Awesome-LLM-Synthetic-Data: A reading list on LLM based Synthetic Data Generation](https://github.com/wasiahmad/Awesome-LLM-Synthetic-Data#:~:text=,GitHub%202024) ). Bu, projenin araştırma camiasında karşılık bulduğunu ve muhtemelen aktif geliştirildiğini gösterir. Henüz geniş bir kullanıcı tabanı olmasa da, amaç “basit ve son derece verimli” olmak olduğu için endüstride de ilgi çekebilir. GitHub reposu (adı “DataDreamer” olması muhtemel) aktif olarak güncelleniyor olabilir.

**GitHub:** DataDreamer projesinin açık kaynak kodu duyurulmuştur, ancak GitHub linki ilgili kaynaklarda açıkça verilmemiştir. Muhtemelen arama ile bulunabilir (ACL 2024 paper’ının yazarları tarafından). Özette bahsedilen şekliyle arayabilirsiniz.

## LLM-Swarm – Ölçeklenebilir Toplu Sentetik Veri Üretimi Aracı

**Açıklama:** **LLM-Swarm**, özellikle büyük ölçekli (milyonlarca satır gibi) sentetik metin üretimi yapmak isteyenler için tasarlanmış bir araçtır. HPC (High Performance Computing) ortamlarında **Slurm** gibi iş kuyruğu yöneticileri kullanarak birden çok LLM örneğini dağıtık biçimde çalıştırmayı sağlar ([GitHub - davanstrien/awesome-synthetic-datasets: awesome synthetic (text) datasets](https://github.com/davanstrien/awesome-synthetic-datasets#:~:text=%23%20llm)). Yani bir sunucu kümesinde, örneğin 100 tane açık kaynak LLM instance’ını paralel çalıştırarak devasa miktarda metni hızlıca üretebilirsiniz. Bu araç, Hugging Face tarafından yayınlanan devasa **Cosmopedia** veri setinin üretilmesinde kullanılmıştır ([GitHub - davanstrien/awesome-synthetic-datasets: awesome synthetic (text) datasets](https://github.com/davanstrien/awesome-synthetic-datasets#:~:text=,inference%20endpoints%20in%20Slurm%20clusters)) (Cosmopedia, 25 milyar token’lık bir sentetik veri derlemesidir).

**Üretilen Veri Tipleri:** Aslında LLM-Swarm belirli bir veri tipiyle sınırlı değildir; hangi prompt’u verirseniz o içeriği üretir, fakat asıl faydası **çok büyük hacimli** veriyi etkin şekilde oluşturmasıdır. Örneğin bir dil modeliyle yüzbinlerce sahte sohbet diyaloğu ya da Wikipedia benzeri makaleler üretmek istiyorsanız, LLM-Swarm bunları cluster üzerinde dağıtarak üretir. Cosmopedia örneğinde, ders kitabı tarzında makaleler, blog yazıları, çeşitli temalarda metinler sentezlendi ([GitHub - davanstrien/awesome-synthetic-datasets: awesome synthetic (text) datasets](https://github.com/davanstrien/awesome-synthetic-datasets#:~:text=)). Kısaca, **metin türü ne olursa olsun**, büyük miktarda üretimde kullanılır.

**Desteklenen LLM’ler:** Genellikle **açık kaynak LLM** modelleri (ör. LLaMA-türevleri, Bloom, GPT-J, vs.) kendi donanımınızda çalıştırmak üzere uygundur. LLM-Swarm bu modelleri Slurm kümesi içinde dağıtık bir şekilde koordine eder. Ayrıca **HuggingFace’in text-generation-inference** veya benzeri sunucularıyla da entegre çalışabilir. Temelde model bağımsızdır; sizin belirlediğiniz LLM ağırlıkları ve kodu üzerinde birden fazla örnek başlatıp görevleri paylaştırır. Örneğin 8 GPU’lu bir sunucuda 8 kopya model çalıştırıp her birine farklı prompt kuyruğu atamak gibi.

**Kullanım & Örnek:** Bu araç daha ileri seviye kullanıcılar içindir. Bir kullanım şekli: Önce Slurm kümenizde bir ortam kurarsınız, ardından LLM-Swarm’ın betiğini çalıştırarak bir istek kuyruğu tanımlarsınız. Komut satırından veya betikten, üretmek istediğiniz veri seti için gereken prompt şablonlarını ve adetlerini belirlersiniz. LLM-Swarm bunları cluster’da dağıtır, her bir parça çıktı düzenli olarak toplanıp birleşik bir veri seti haline getirilir. Örneğin “1 milyon farklı başlık altında kısa makaleler üret” dediğinizde, bu görevler yüzlerce alt göreve bölünüp paralel gerçekleştirilir. LLM-Swarm kodu, her bir işin durumunu ve çıktısını takip eder. Geliştiriciler, HPC altyapısına aşina kişilere yönelik örnek konfigürasyonlar sunmuştur.

**Güncellik & Aktiflik:** Bu proje özellikle büyük veri seti üretimi gerektiren araştırmalarda ortaya çıkmıştır. Cosmopedia’nın üretimi 2024’te gerçekleştiği için araç güncel sayılır. Ancak bu, niş bir kullanım alanıdır; sürekli bir topluluk geliştirmesi yerine hedefli bir geliştirme olabilir. Yine de Github’da paylaşıldığından, HPC ve LLM ile uğraşanlar için değerli bir referans projesidir.

**GitHub:** [mlabonne/llm-swarm](https://github.com/mlabonne/llm-swarm) (Hugging Face ekosisteminden Maxime Labonne tarafından geliştirilen araç)

## Domain-Specific Dataset Creator (Hugging Face “Data Is Better Together”)

**Açıklama:** Hugging Face topluluğunda yürütülen **Domain Specific Dataset Project**, belirli bir uzmanlık alanına yönelik veri setlerini **LLM’ler yardımıyla sıfırdan oluşturmaya** odaklanan bir girişimdir ([GitHub - davanstrien/awesome-synthetic-datasets: awesome synthetic (text) datasets](https://github.com/davanstrien/awesome-synthetic-datasets#:~:text=)). Fikir olarak, alan uzmanlarıyla iş birliği yaparak o alana dair yüksek kaliteli yapay veriler üretmek ve bu verileri model eğitimi için kullanmak hedeflenir. Örneğin, tıbbi makalelerden otomotiv teknik dökümanlarına kadar dar bir alandaki veriyi LLM destekli üretmek. Proje, bir **grafiksel arayüzlü araç** da içeriyor; bu arayüz, kullanıcıların bir sentetik veri üretim hattı tanımlamasını sağlıyor.

**Üretilen Veri Tipleri:** Bu proje kapsamında üretilecek veriler tamamen hedef alana özgüdür. Eğer alan finans ise, **finansal diyaloglar, müşteri-hizmet senaryoları, rapor özetleri** gibi o domaine uygun içerikler üretilebilir. Alan tıp ise, **doktor-hasta diyalogları, tıbbi vakalar, reçete açıklamaları** vb. söz konusu olabilir. Araç, kullanıcıdan alanla ilgili bazı temel girdiler alıp o doğrultuda LLM’den çeşitli örnekler üretmesini sağlayacaktır. Sonuçta, elde edilen veri seti hedef alandaki gerçek veriye benzer dağılımda ancak sentetik olarak üretilmiş olacak.

**Desteklenen LLM’ler:** Muhtemelen bu proje Hugging Face altyapısını kullandığı için **HuggingFace Hub üzerindeki modeller** ön plandadır. Örneğin, alan özelinde ince ayar yapılmış bir GPT-2/3 türevi model ya da büyük bir genel model (Bloom, LLaMA-2 vb.) kullanılabilir. Proje henüz başlangıç aşamasında olduğundan, arka planda doğrudan API çağrılarıyla OpenAI modelini kullanmak yerine açık modelleri kullanmayı teşvik ediyor olabilir. Ancak kesin desteklenenler, projenin dokümantasyonunda belirtilmelidir.

**Kullanım & Örnek:** Domain-Specific Dataset Creator aracı, bir **web arayüzü** şeklinde düşünülmüştür. Kullanıcı arayüzü üzerinden bir pipeline (iş akışı) tasarlar: Önce alanı ve amacını tanımlar, ardından LLM’in üretmesi istenen veri formatını ve miktarını belirtir. Örneğin, “Otomotiv parçaları hakkında SSS (sıkça sorulan sorular) veri seti oluştur” gibi bir görev verilebilir. Sistem, bu komutu alıp LLM’den farklı soru-cevap çiftleri üretmesini sağlar. Kullanıcı üretilen örnekleri inceleyip gerekiyorsa düzenleyebilir, filtreleyebilir. Bu şekilde etkileşimli bir süreç sonunda, o alana özgü bir sentetik veri kümesi elde edilir. Bu yaklaşım, alan uzmanlarının yönlendirmeleriyle LLM’in yaratıcı gücünü birleştirir.

**Güncellik & Aktiflik:** Bu proje Hugging Face’in “**data-is-better-together**” inisiyatifinin bir parçası olup 2024 civarında ortaya çıkmıştır. Henüz deneysel sayılır; aktif bir geliştirme ekibi ve kullanıcı geri bildirimiyle şekillenmektedir. Büyük bir topluluk projesi olduğundan, sürekliliği muhtemelen Hugging Face tarafından desteklenecektir.

**GitHub:** [huggingface/data-is-better-together (domain-specific-datasets)](https://github.com/huggingface/data-is-better-together) – (Domain specific dataset tool bölümü)

## Stanford Alpaca (Self-Instruct Yöntemi ile Sentetik Talimat Verisi)

**Açıklama:** **Stanford Alpaca** projesi, LLM tabanlı sentetik veri üretiminin çarpıcı bir örneğidir. 2023 başlarında Stanford ekibi, OpenAI’nin **text-davinci-003** modelini kullanarak yaklaşık 52.000 adet talimat-yanıt çiftinden oluşan bir veri setini otomatik olarak oluşturmuştur. Bu yaklaşım, **Self-Instruct** adı verilen bir çerçeveye dayanır ([Synthetic dataset generation techniques: Self-Instruct](https://huggingface.co/blog/davanstrien/self-instruct#:~:text=%3E%20Self,source)). Self-Instruct, küçük bir insan yazımı talimat setini başlangıç alarak modelin kendi ürettiği yeni talimatlar ve bunlara verdiği yanıtlarla veri kümesini büyütme prensibine dayanır. Yani model, kendi kendini talimatlarla eğitmiş olur ([Synthetic dataset generation techniques: Self-Instruct](https://huggingface.co/blog/davanstrien/self-instruct#:~:text=%3E%20Self,source)). Alpaca projesinde de önce bir çekirdek talimat listesi belirlendi, daha sonra GPT-3.5 modeli defalarca çalıştırılarak yeni sorular ve bu soruların cevabı niteliğinde sentetik veriler üretildi. Sonuç, Alpaca-7B modelinin bu 52k sentetik veri ile eğitilerek şaşırtıcı derecede iyi bir talimat takibi yeteneği kazanması oldu.

**Üretilen Veri Tipleri:** Alpaca’nın oluşturduğu veri seti **çeşitli kullanıcı talimatları ve bu talimatlara verilen yanıtları** içeriyordu. Talimatlar oldukça geniş bir yelpazeyi kapsar: genel bilgi soruları, yaratıcı yazma istekleri, tavsiyeler, mantık bulmacaları, kodlama görevleri, özetleme istekleri vb. Bu çeşitlilik, modelin geniş bir kullanım alanında test edilmesini sağlar. Kısacası, Alpaca veri seti **Q&A, sohbet, talimat uygulama, özetleme** gibi birçok görevi içeren çok yönlü bir sentetik derlemedir. Bu da gösteriyor ki Self-Instruct yöntemi, modelden neredeyse insan çeşitliliğinde talimatlar türetebilir.

**Desteklenen LLM’ler:** Alpaca veri setinin üretiminde **OpenAI’nin GPT-3 (davinci-003)** modeli kullanılmıştır. Yani yöntem, güçlü bir harici modele dayanmıştır. Ancak elde edilen veri seti daha sonra **LLaMA 7B** gibi daha küçük açık kaynak modele aktarılarak bu model eğitildi. Self-Instruct yöntemi genel bir yöntem olduğundan, benzer şekilde başka modellerle de yapılabilir: Örneğin Anthropic Claude veya GPT-4 kullanarak daha da kaliteli talimat verileri üretilebilir. Nitekim daha sonraki projelerde GPT-4 ile **Evol-Instruct** (talimatları karmaşıklaştırarak üretme) yaklaşımı da uygulanmıştır (WizardLM projesi gibi). Yani, yöntemin kendisi modelden bağımsız olsa da, **kaliteli sentetik veri** için daha büyük bir “öğretmen” model kullanmak gerekir.

**Kullanım & Örnek:** Stanford Alpaca, bir kütüphane değil bir deneysel projeydi, bu yüzden bir “araç” gibi kullanılmaz ama benzeri bir süreci yineleyebilirsiniz. Alpaca’nın GitHub deposunda veri üretimi için kullanılan prompt şablonları ve kod örnekleri paylaşılmıştır. Örneğin, bir Python betiği içinde OpenAI API’ına bağlanıp şöyle bir istem gönderilir: “Aşağıda bir talimat var. Bu talimatı en iyi şekilde yerine getirecek yanıtı yaz.” Ardından modelin cevabı kaydedilir. Bu şekilde binlerce kez çağrı yapılır (farklı başlangıç talimatlarıyla). Alpaca ekibi, bunu $500 gibi bir API maliyetiyle gerçekleştirmiştir. Elde edilen alpaca\_data.json dosyası, 52k satır talimat/cevap içerir ve bu, Alpaca modelini eğitmek için kullanıldı. Günümüzde, bu veri açıkça mevcuttur; dolayısıyla siz de Alpaca veri setini kullanarak veya aynı kodu OpenAI API anahtarınızla çalıştırarak benzer sentetik veri üretebilirsiniz.

**Güncellik & Aktiflik:** Alpaca projesi, LLM toplumunda bir dönüm noktası olmuş ve pek çok türeve ilham vermiştir. Ancak doğası gereği tek seferlik bir veri üretimidir; yani Alpaca deposu çok aktif gelişmez (veri sabit, model yayınlandı). Yine de Self-Instruct tekniği üzerine birçok yeni çalışma yapıldı: Örneğin **Vicuna**, **WizardLM**, **Orca** gibi modeller ya Alpaca verisine dayandı ya da benzer sentetik veri yöntemlerini kullandı. Bu açıdan, Self-Instruct yaklaşımı güncelliğini koruyor ve çeşitli projelerde farklı şekillerde uygulanıyor. Alpaca’nın kendisi ise 2023’te yayınlanmış tamamlanmış bir projedir.

**GitHub:** [tatsu-lab/stanford\_alpaca](https://github.com/tatsu-lab/stanford_alpaca) – Alpaca veri seti ve modelinin çıktıları (Stanford). Ayrıca Self-Instruct orijinal uygulaması için Yizhong Wang’ın [Self-Instruct reposu](https://github.com/yizhongw/self-instruct) da incelenebilir.

## Self-Contrast – LLM ile Geri Bildirimsiz İyileştirme (Deneysel)

**Açıklama:** **Self-Contrast**, LLM’leri insan geri bildirimi olmadan hizalamak (align etmek) için önerilen yenilikçi bir yöntemdir ([GitHub - davanstrien/awesome-synthetic-datasets: awesome synthetic (text) datasets](https://github.com/davanstrien/awesome-synthetic-datasets#:~:text=%23%20Self)). Bu proje, bir anlamda sentetik tercih verisi üretimi yapar. Starling adlı ekip tarafından geliştirilen bu yaklaşımda, bir modelin belirli konularda kendi içinde çelişen cevaplar üretmesi sağlanır ve bu cevaplar arasından tercih yaparak bir **ödül modeli** eğitilir. Bu ödül modeli daha sonra orijinal dil modelini, **AI Feedback** ile güçlendirilmiş şekilde ince ayar yapmak için kullanılır (Reinforcement Learning with AI Feedback, RLAIF). Kısacası, insan yerine AI’nın ürettiği sahte tercih sinyalleriyle modeli daha iyi hale getirmeye çalışırlar.

**Üretilen Veri Tipleri:** Self-Contrast çerçevesinde üretilen veri, klasik anlamda kullanıcı talimatı – yanıt ikilisi değil, **modelin farklı yanıtları ve bunların karşılaştırmalarıdır**. Örneğin, bir soru için modelden iki farklı yanıt ürettirilir; daha sonra sistem, belirli ölçütlere göre bunlardan hangisinin tercih edilmesi gerektiğine dair bir “kazanan/kaybeden” etiketi verir – bu bir nevi **sentetik karşılaştırmalı tercih verisi** dir. Bu veri, bir ödül modeli eğitmek için kullanılır. Projede üretilen **Nectar** adlı veri seti bu şekilde modelin kendi hatalarından ders çıkarmasını sağlayacak etiketler içerir ([GitHub - davanstrien/awesome-synthetic-datasets: awesome synthetic (text) datasets](https://github.com/davanstrien/awesome-synthetic-datasets#:~:text=%3E%20Self,for%20aligning%20with%20human%20preference)). Bu veriler içerik itibariyle daha çok **iyilik kötülük, doğruluk yanlışlık, uygunluk** gibi yargıları barındırır.

**Desteklenen LLM’ler:** Self-Contrast yönteminde, temel dil modeli hem veri üreticisi hem de iyileştirilecek hedeftir. Örneğin Anthropıc’in Claude veya OpenAI’nın modelleriyle bu yapılabilir. Starling ekibi kendi LLM’lerini hedef almıştı (muhtemelen bir varyant GPT-3 modelini). Yöntemin kendisi, istenen herhangi bir büyük dil modeline uygulanabilir. Kod açık kaynak olarak sunulduğundan, isteyen kendi modeline bu süreci uygulayabilir. Yani model desteği açık; tek ihtiyacınız, yeterince güçlü bir LLM ve hesaplama kaynağı.

**Kullanım & Örnek:** Self-Contrast kodu, ilgili makalenin (Extensive Self-Contrast Enables Feedback-Free LM Alignment) deposunda mevcuttur. Kullanmak için, önce bir dil modeli alıyorsunuz ve çeşitli istemler vererek ondan birden fazla yanıt üretmesini sağlıyorsunuz. Sonra bu yanıt çiftlerini bir **karşılaştırma modülüne** sokup hangi yanıtın tercih edilmesi gerektiğini (hangisinin daha yararlı veya zararsız olduğu gibi) otomatik belirliyorsunuz. Bu aşamada insan yok, bunun yerine önceden tanımlanmış otomatik metrikler veya kurallar olabilir. Ortaya çıkan etiketlenmiş veri (ör: “Cevap A, Cevap B’ye tercih edilir çünkü daha az toksik”) ile bir ödül modeli eğitiliyor. Ardından klasik **PPO** gibi pekiştirmeli öğrenme algoritmalarıyla ana dil modeli bu ödül modele göre optimize ediliyor. Kod örnekleri genellikle PyTorch ile bu döngüyü kurmayı içerir. Bu oldukça ileri düzey bir kullanım olduğundan, deneysel bir uygulama isteyen araştırmacılar içindir. Gündelik veri üretimi aracı olarak kullanılmaz.

**Güncellik & Aktiflik:** Bu yöntem 2024’te ortaya atılmış olup henüz deneysel aşamadadır. İnsan geribildirimi olmadan model hizalama fikri ilgi çekicidir ancak pratikte ne derece başarılı olduğu araştırılıyor. Self-Contrast’ın GitHub kodu (Nectar dataset ile birlikte) yayınlanmıştır ve akademik çevrede tartışılmaktadır ([GitHub - davanstrien/awesome-synthetic-datasets: awesome synthetic (text) datasets](https://github.com/davanstrien/awesome-synthetic-datasets#:~:text=%3E%20Self,for%20aligning%20with%20human%20preference)). Geniş bir kullanıcı tabanı yoktur, daha çok araştırma deneyi niteliğindedir. İleride insan müdahalesini azaltmak adına benzeri yaklaşımlar geliştirilmeye devam edebilir.

**GitHub:** [HazyResearch/nectar](https://github.com/HazyResearch/nectar) (Self-Contrast kod ve “Nectar” veri seti)

**Kaynaklar:** Proje açıklamalarındaki bilgiler ilgili GitHub sayfalarından ve makalelerden derlenmiştir. Örneğin Argilla ve Distilabel için resmi README’ler ([GitHub - davanstrien/awesome-synthetic-datasets: awesome synthetic (text) datasets](https://github.com/davanstrien/awesome-synthetic-datasets#:~:text=,data%20ownership%2C%20and%20overall%20efficiency)) ([GitHub - argilla-io/synthetic-data-generator: Build datasets using natural language](https://github.com/argilla-io/synthetic-data-generator#:~:text=Supported%20Tasks%3A)), Kiln için yazarın Reddit açıklaması ([In 2024, what is the best tool/framework for creating synthetic data that can then be used to fine-tune with? : r/LocalLLaMA](https://www.reddit.com/r/LocalLLaMA/comments/194m01m/in_2024_what_is_the_best_toolframework_for/#:~:text=Late%20addition%2C%20but%20I%20created,AI%2FKiln)), Fuxion için README örnek kodu ve açıklamaları ([GitHub - Tobiadefami/fuxion: Sythetic data generation and normalization functions powered by LLMs](https://github.com/tobiadefami/fuxion#:~:text=This%20pipeline%20generates%20a%20dataset,directory)) ([GitHub - Tobiadefami/fuxion: Sythetic data generation and normalization functions powered by LLMs](https://github.com/tobiadefami/fuxion#:~:text=Models%20supported)), DataDreamer ve diğer “awesome” liste öğeleri için Hugging Face topluluk derlemeleri kullanılmıştır ([GitHub - davanstrien/awesome-synthetic-datasets: awesome synthetic (text) datasets](https://github.com/davanstrien/awesome-synthetic-datasets#:~:text=)) ([GitHub - davanstrien/awesome-synthetic-datasets: awesome synthetic (text) datasets](https://github.com/davanstrien/awesome-synthetic-datasets#:~:text=)). Self-Instruct ve Alpaca ile ilgili genel bilgiler, Hugging Face blog yazısından ve orijinal makaleden özetlenmiştir ([Synthetic dataset generation techniques: Self-Instruct](https://huggingface.co/blog/davanstrien/self-instruct#:~:text=%3E%20Self,source)). Bu sayede, LLM tabanlı sentetik veri üretimi konusundaki önemli girişimler kapsamlı şekilde özetlenmiştir.

# LLM Kullanarak Sentetik Veri Üretme Yöntemleri, Kütüphaneler ve Kaynaklar

**Giriş**

Sentetik veri, gerçek dünya olaylarından ziyade algoritmalar veya modeller aracılığıyla oluşturulan yapay veri olarak tanımlanır.1 Gerçek verinin elde edilmesinin zor, maliyetli veya hassas olduğu durumlarda sentetik veri, önemli bir alternatif sunmaktadır.1 Özellikle Büyük Dil Modelleri'nin (LLM'ler) son yıllarda kaydettiği ilerlemeler, metin, kod ve diğer veri türlerinde bağlamsal olarak tutarlı ve gerçekçi sentetik veri üretme konusunda yeni ve güçlü olanaklar yaratmıştır.4 Yapay zeka eğitim verilerinin önemli bir bölümünü sentetik verinin oluşturması beklenmektedir; Gartner'a göre bu oranın 2024 yılında %60'a ulaşacağı öngörülmektedir.8 Bu durum, sentetik verinin LLM geliştirme ve test süreçlerinde ne kadar kritik bir rol oynadığını açıkça göstermektedir. Gerçek veri toplamanın getirdiği zorluklar ve maliyetler düşünüldüğünde, LLM'lerin gelişmiş yetenekleri sayesinde sentetik verinin kalitesinin artması, bu alana olan ilgiyi ve yatırımı önemli ölçüde artırmaktadır. Sentetik veri, sadece bir veri üretme yöntemi olmanın ötesinde, yapay zeka geliştirme süreçlerindeki temel sorunlara (veri eksikliği, gizlilik, önyargı) çözüm sunan stratejik bir araç haline gelmektedir.

Bu raporun amacı, LLM'ler kullanılarak sentetik veri üretme yöntemlerini, bu amaçla geliştirilmiş Python kütüphanelerini ve Medium platformundaki bu konuları detaylı olarak ele alan kaynakları incelemektir. Rapor kapsamında, farklı LLM modellerinin sentetik veri üretimindeki rolleri, bu süreçte dikkat edilmesi gereken etik hususlar ve üretilen verinin kalitesini ölçmek için kullanılan metrikler de değerlendirilecektir.

**LLM Kullanarak Sentetik Veri Üretme Yöntemleri**

LLM'ler kullanılarak sentetik veri üretimi, çeşitli yöntemler ve yaklaşımlar aracılığıyla gerçekleştirilebilir. Bu yöntemler, LLM'lerin dil anlama ve üretme yeteneklerinden faydalanarak, istenen özelliklere sahip yapay veri setleri oluşturmayı hedefler.

**Prompt Mühendisliği (Prompt Engineering)**

Prompt mühendisliği, LLM'lere verilen girdileri (promptları) dikkatli bir şekilde tasarlayarak, modelin istenen türde ve kalitede sentetik veri üretmesini sağlamayı amaçlar.6 Etkili bir prompt genellikle üç temel bileşeni içerir: görev tanımı (üretilecek verinin ne olduğu), üretim koşulları (verinin hangi formatta veya yapıda olması gerektiği) ve bağlam içi örnekler (modelin nasıl bir çıktı üretmesi gerektiğine dair örnekler).6 LLM'ler, önceden herhangi bir örnek verilmeden (sıfır-shot) veya az sayıda örnek verilerek (few-shot) etiketli veri üretebilirler.6 İyi tasarlanmış promptlar, üretilen sentetik verinin kalitesini, tutarlılığını ve çeşitliliğini önemli ölçüde etkileyebilir. Prompt mühendisliği, sentetik veri üretiminde LLM'lerin davranışını kontrol etmenin ve istenen formatta, yapıda ve içerikte veri üretmesini sağlamanın kritik bir yoludur. LLM'lerin temel yeteneği, verilen metni anlamak ve buna uygun yanıtlar üretmektir. Promptlar, LLM'lere ne tür bir veri üretmeleri gerektiğini söyleyen talimatlardır. Görev tanımı, neyin üretileceğini; üretim koşulları, nasıl üretileceğini; bağlam içi örnekler ise LLM'ye yol göstermesini sağlar.

**Çok Adımlı Üretim (Multi-Step Generation)**

Çok adımlı üretim, karmaşık sentetik veri üretim görevlerini daha küçük ve yönetilebilir adımlara bölmeyi içeren bir yaklaşımdır.6 Bu yöntem, özellikle yapılandırılmış veya birbirine bağlı veri setleri oluşturulurken faydalı olabilir.6 Örneğin, bir belgeden soru-cevap çiftleri oluşturulurken, ilk adım belgenin anlamlı parçalara ayrılması (chunking), ikinci adım ilgili bağlamın oluşturulması, üçüncü adım bu bağlama dayalı soruların üretilmesi ve son adım olarak da bu sorulara uygun cevapların (beklenen çıktılar) oluşturulması olabilir.12 Karmaşık veri üretim görevlerinin tek bir prompt ile yönetilmesi zor olabileceği için, görevi alt görevlere ayırmak ve her bir alt görev için ayrı promptlar kullanmak daha etkili olabilir. Bu, her adımda daha fazla kontrol ve özelleştirme imkanı sunar, böylece üretilen verinin tutarlılığı ve doğruluğu artırılabilir.

**Daha Güçlü Modellerden Distilasyon (Distillation from Stronger Models)**

Distilasyon, daha yetenekli bir modelin (öğretmen model) çıktılarını kullanarak, daha az yetenekli bir modeli (öğrenci model) eğitme sürecidir.5 Bu yöntem, büyük ve karmaşık öğretmen modelin bilgi ve muhakeme becerilerini, daha küçük ve daha verimli öğrenci modellere aktararak, yanıt kalitesini ve hesaplama verimliliğini optimize etmeyi amaçlar.8 Distilasyon, büyük ve maliyetli LLM'lerin yeteneklerini daha küçük ve daha verimli modellere aktarmanın bir yoludur. Bu, kaynak kısıtlaması olan veya daha hızlı çıkarım gerektiren uygulamalar için faydalı olabilir. Ancak, bu yöntemi kullanırken, eğer harici bir modelin API'si kullanılıyorsa, API sağlayıcısının kullanım koşullarını ihlal etme riskine de dikkat etmek gereklidir.8

**Kendi Kendini Geliştirme Teknikleri (Self-Improvement Techniques)**

Kendi kendini geliştirme, bir modelin kendi ürettiği çıktılar üzerinden yinelemeli bir şekilde öğrenmesini içeren bir tekniktir.5 Bu süreç genellikle, küçük bir başlangıç görev havuzu ile başlar. Model, bu havuzdaki görevlerden rastgele örnekler seçerek, yeni talimatlar ve bunlara karşılık gelen örnekler üretmek için kullanılır.8 Kendi kendini geliştirme, etiketlenmiş veri eksikliğinde veya modelin kendi ürettiği verilerle performansını artırmak istendiğinde faydalı olabilir. Ancak, bu yöntemde modelin kendi hatalarını tekrarlama veya mevcut önyargıları pekiştirme riski de bulunmaktadır.

**Alanı Genişletilmiş Üretim (Retrieval-Augmented Generation - RAG)**

Alanı genişletilmiş üretim (RAG), sentetik veri üretim sürecine harici bilgi kaynaklarını (bilgi tabanları, belgeler vb.) entegre ederek, üretilen verinin factuality (olgusal doğruluk) ve bağlamsal doğruluğunu artırmayı hedefleyen bir yöntemdir.7 Bu yaklaşım, özellikle belirli bir alan veya konu hakkında sentetik veri üretmek gerektiğinde büyük fayda sağlar.13 RAG, LLM'lerin geniş genel bilgi birikimini, belirli alanlardaki daha derinlemesine bilgilerle birleştirerek, daha doğru ve bağlamsal olarak zengin sentetik veri üretmeyi mümkün kılar. Bu, özellikle müşteri destek sistemleri, araştırma araçları ve dahili bilgi yönetimi gibi bilgi yoğun uygulamalar için önemlidir.

**Python Sentetik Veri Kütüphaneleri**

LLM'ler kullanılarak sentetik veri üretimi için çeşitli Python kütüphaneleri mevcuttur. Bu kütüphaneler, veri üretme süreçlerini kolaylaştıran ve otomatikleştiren araçlar ve işlevler sunar.

**Promptwright**

Promptwright, Stacklok tarafından geliştirilen bir Python kütüphanesidir ve LLM'ler kullanarak büyük ölçekli sentetik veri setleri oluşturmak için tasarlanmıştır.14 Kütüphane, yerel LLM'ler (Ollama, VLLM) ve çeşitli LLM servis sağlayıcıları (OpenAI, Anthropic, OpenRouter vb.) ile uyumludur.14 Kullanıcılar, YAML yapılandırma dosyaları veya doğrudan Python kodu aracılığıyla veri üretme görevlerini tanımlayabilirler.14 Promptwright ayrıca, komut satırı arayüzü (CLI) üzerinden doğrudan çalıştırma ve üretilen veri setlerini Hugging Face Hub'a otomatik olarak gönderme gibi özelliklere de sahiptir.14 Bu kütüphane, özellikle LLM tabanlı sentetik veri üretimine odaklanması ve farklı LLM sağlayıcıları ile kolay entegrasyon sunması açısından kullanışlıdır. Yerel LLM desteği sayesinde, API maliyetleri ve hassas verilerin dış ortamlara gönderilmesi konusundaki endişeler de azaltılabilir.

**SDV (Synthetic Data Vault)**

SDV (Synthetic Data Vault), tablolar halinde sentetik veri oluşturmak için kapsamlı bir Python kütüphanesidir.15 Tek tablolu ve çok tablolu (ilişkisel) veriler için çeşitli yapay zeka modelleri sunar.15 SDV, sentetik veriyi değerlendirme ve görselleştirme araçlarına da sahiptir, bu sayede üretilen verinin kalitesi ve gerçek veriyle olan benzerliği analiz edilebilir.15 Ayrıca, veri anonimleştirme ve iş mantığına özel kurallar ekleme gibi özelleştirme seçenekleri de sunar.15 SDV, genel amaçlı bir sentetik veri kütüphanesi olmasına rağmen, LLM'lerle doğrudan entegrasyonu olmasa da, LLM'ler tarafından üretilen verilerin yapısını taklit etmek veya LLM'ler tarafından üretilen verilerle birlikte kullanılmak üzere tabular sentetik veri oluşturmak için kullanılabilir.

**ydata-synthetic**

ydata-synthetic, çeşitli sentetik veri üretme stratejileri sunan bir başka Python kütüphanesidir.16 CTGAN (Conditional Tabular GAN), TimeGAN (Time-Series GAN) ve GMM (Gaussian Mixture Model) tabanlı modeller gibi farklı veri türleri için çeşitli modeller içerir.16 Kütüphane, veri okumadan sentetik veri profilini çıkarmaya kadar tüm süreci yöneten kullanıcı dostu bir Streamlit uygulamasına da sahiptir.16 ydata-synthetic, farklı veri türleri için geniş model seçenekleri sunması ve kolay kullanılabilir bir arayüze sahip olması nedeniyle dikkat çekicidir. LLM'ler tarafından üretilen verinin türüne bağlı olarak uygun bir seçenek olabilir.

**Diğer İlgili Kütüphaneler**

* **Gretel Synthetics:** LSTM (Long Short-Term Memory), Time-Series DGAN (Deep Convolutional Generative Adversarial Network) ve ACTGAN gibi modelleri içeren, özelleştirme seçenekleri sunan bir kütüphanedir.16
* **nbsynthetic:** Küçük ve orta ölçekli, hem sayısal hem de kategorik özelliklere sahip karma veri setleri için GAN tabanlı bir mimari kullanır.16
* **synthcity:** Tabular ve zaman serisi verileri için çeşitli GAN ve VAE (Variational Autoencoder) tabanlı modeller sunar. Gizlilik ve adalet gibi belirli odak alanlarına sahip modelleri de bulunmaktadır.16
* **Faker:** Sahte isimler, adresler, e-posta adresleri gibi çeşitli dummy (yer tutucu) veriler üretmek için yaygın olarak kullanılan bir Python paketidir.17 Sentetik veri oluşturmaktan ziyade genellikle test amaçlı veya prototip geliştirme süreçlerinde kullanılır.

Bu diğer kütüphaneler, LLM'ler tarafından üretilen verinin özelliklerine veya projenin özel gereksinimlerine bağlı olarak alternatif veya tamamlayıcı çözümler sunabilir. Faker ise, özellikle uygulama geliştirme ve test aşamalarında faydalı olabilir.

**Medium Platformundaki Kaynaklar**

Medium platformu, LLM'ler kullanılarak sentetik veri üretme yöntemlerini ve araçlarını anlatan çeşitli makalelere ev sahipliği yapmaktadır.8 "Sentetik veri üretimi LLM" ve "synthetic data generation LLM" gibi anahtar kelimelerle yapılan aramalar, bu konuda birçok değerli kaynağı ortaya çıkarmaktadır. Bu makaleler genellikle farklı yaklaşımları, kullanılan prompt mühendisliği tekniklerini ve pratik uygulama örneklerini detaylı bir şekilde açıklamaktadır.8

Bazı Medium makaleleri, belirli LLM modellerini (örneğin Llama 3) kullanarak soru-cevap çiftleri veya diğer metin tabanlı verileri sentetik olarak üretme adımlarını ayrıntılı olarak sunmaktadır.20 Bu tür yazılar, adım adım rehberler ve kod örnekleri sağlayarak, kullanıcıların teorik bilgiyi pratik uygulamalara dönüştürmelerine yardımcı olabilir. Diğer makaleler ise, veri augmentasyonu, distilasyon ve model-in-the-loop gibi farklı sentetik veri üretim tekniklerini LLM'ler bağlamında ele almaktadır.5 Bu yazılar, farklı yöntemlerin avantajlarını, dezavantajlarını ve ne zaman kullanılması gerektiğini tartışarak kullanıcılara daha geniş bir perspektif sunar.

Medium makaleleri genellikle ilgili araştırma makalelerine, GitHub depolarına veya diğer önemli kaynaklara atıfta bulunmaktadır.8 Bu referanslar, kullanıcılara konuyu daha derinlemesine araştırma ve ilgili araçlara erişim imkanı sunarak öğrenme süreçlerini destekler. Medium, sentetik veri üretimi konusunda hem teorik bilgiye hem de pratik uygulamalara yönelik güncel ve erişilebilir içerik sunan değerli bir kaynaktır. Kullanıcılar, bu platformdaki makaleler aracılığıyla farklı yöntemleri ve araçları daha yakından tanıyabilir ve bu alandaki en son gelişmeleri takip edebilirler.

**Farklı LLM Modellerinin Kullanımı**

Sentetik veri üretimi için çeşitli LLM modelleri kullanılmaktadır. Bu modellerin her biri, farklı özelliklere, avantajlara ve dezavantajlara sahiptir. Model seçimi, üretilecek sentetik verinin türüne, kalitesine ve projenin özel gereksinimlerine bağlıdır.

GPT-3 ve GPT-4 gibi modeller, yüksek kaliteli ve bağlamsal olarak tutarlı sentetik veri üretme yetenekleri sayesinde yaygın olarak tercih edilmektedir.3 Bu modeller, çeşitli promptlarla insan benzeri metinler üretebilir ve bu sayede farklı amaçlar için sentetik veri setleri oluşturulabilir. Ancak, bu modeller genellikle API üzerinden erişilebilir ve kullanım maliyetleri diğer modellere göre daha yüksek olabilir. Ayrıca, API kullanımında veri gizliliği endişeleri de dikkate alınmalıdır.

Llama ve Mistral gibi açık kaynaklı modeller de, sentetik veri üretiminde giderek daha fazla ilgi görmektedir.14 Bu modeller, kullanıcıların kendi altyapıları üzerinde çalıştırılabilir ve özelleştirilebilir olmaları nedeniyle avantaj sunar. Özellikle büyük ölçekli deneyler yapmak veya hassas verilerle çalışmak isteyen kullanıcılar için bu modeller cazip bir seçenek olabilir. Ancak, bu modellerin ticari kullanımında lisanslama koşullarına dikkat etmek gerekebilir ve bazı durumlarda aynı kalitede sonuçlar elde etmek için daha fazla kaynak gerekebilir.

Aşağıdaki tablo, sentetik veri üretiminde yaygın olarak kullanılan bazı LLM modellerinin avantajlarını ve dezavantajlarını karşılaştırmaktadır:

|  |  |  |
| --- | --- | --- |
| **Model** | **Avantajları** | **Dezavantajları** |
| GPT-3/4 | Yüksek kalite, geniş bilgi, iyi bağlamsal tutarlılık | API maliyeti, veri gizliliği endişeleri |
| Llama | Açık kaynak, özelleştirilebilir, yerel kullanım imkanı | Ticari kullanımda lisanslama koşulları, daha fazla kaynak gerektirebilir |
| Mistral | Açık kaynak, iyi performans, ticari kullanıma uygun lisans | Llama'ya göre daha yeni, topluluk desteği daha az olabilir |
| Diğerleri | Belirli görevler için optimize edilmiş olabilir (örneğin kod üretimi için özel modeller) | Genel amaçlı modeller kadar geniş bilgiye sahip olmayabilir, daha az yaygın olabilir |

Bu karşılaştırma, kullanıcıların projeleri için en uygun LLM modelini seçmelerine yardımcı olabilir. Model seçimi yapılırken, projenin özel gereksinimleri, bütçe, performans beklentileri ve gizlilik ihtiyaçları gibi faktörler dikkate alınmalıdır.

**Etik Hususlar ve Veri Kalitesi**

Sentetik veri üretimi, beraberinde bir dizi etik hususu ve veri kalitesi değerlendirmesini getirmektedir.1 Özellikle LLM tabanlı yöntemlerde bu konuların önemi daha da artmaktadır.

**Etik Hususlar**

Sentetik veri üretimi sürecinde dikkat edilmesi gereken temel etik konular arasında önyargı (bias) ve gizlilik yer almaktadır.5 Gerçek veride var olan önyargılar, sentetik verilere de yansıyabilir ve bu durum, yapay zeka modellerinin adaletsiz veya ayrımcı sonuçlar üretmesine neden olabilir.5 Bu nedenle, sentetik veri üretilirken kullanılan kaynak verinin ve modelin önyargılardan arındırılması veya bu önyargıların farkında olunması ve gerekli önlemlerin alınması büyük önem taşır.

Gizlilik, sentetik veri üretiminde bir diğer önemli etik konudur.4 Her ne kadar sentetik veri, gerçek kişilere ait bilgileri içermemesi gerekse de, bazı durumlarda hassas bilgilerin yanlışlıkla sentetik verilere sızma riski bulunabilir. Özellikle LLM'ler gibi karmaşık modeller kullanıldığında, modelin eğitim verilerinden öğrendiği bilgileri sentetik verilere yansıtma potansiyeli göz önünde bulundurulmalıdır. Bu nedenle, sentetik veri üretiminde gizliliği koruyacak tekniklerin kullanılması (örneğin, diferansiyel gizlilik) ve veri üretim süreçlerinin şeffaf olması önemlidir.

LLM'lerin kendilerinde var olan dilsel önyargılar ve potansiyel olarak yanlış veya zararlı içerik üretme yetenekleri nedeniyle, LLM tabanlı sentetik veri üretiminde etik konuların önemi daha da artmaktadır.25 Bu nedenle, LLM'lerin çıktıları dikkatli bir şekilde izlenmeli, doğrulanmalı ve etik prensiplere uygunluğu sağlanmalıdır.

**Veri Kalitesi Metrikleri**

Sentetik verinin kalitesini değerlendirmek için çeşitli metrikler kullanılmaktadır. Bu metrikler genellikle üç ana boyutta incelenir: doğruluk (fidelity), kullanılabilirlik (utility) ve gizlilik (privacy).6

* **Doğruluk (Fidelity):** Sentetik verinin, orijinal gerçek verinin istatistiksel özelliklerini ve yapısını ne kadar iyi yansıttığını ölçer.6 Bu boyut, sentetik verinin gerçek veriye benzer dağılımlara, korelasyonlara ve diğer istatistiksel özelliklere sahip olup olmadığını değerlendirir. İstatistiksel benzerlik testleri, dağılım benzerliği skorları ve korelasyon katsayıları gibi metrikler bu boyutta kullanılır.30
* **Kullanılabilirlik (Utility):** Sentetik verinin, gerçek dünya uygulamalarında (örneğin, model eğitimi, test etme) ne kadar işe yaradığını ölçer.6 Bu boyut, sentetik veri üzerinde eğitilen modellerin, gerçek veri üzerinde eğitilen modellere benzer performans gösterip göstermediğini değerlendirir. Model performans skorları (doğruluk, F1 skoru vb.) ve özellik önem skorları gibi metrikler bu boyutta kullanılır.30
* **Gizlilik (Privacy):** Sentetik verinin, hassas bilgileri (eğer varsa) ne kadar iyi koruduğunu ölçer.6 Bu boyut, sentetik veriden orijinal verilere ait bilgilerin yeniden çıkarılma riskini değerlendirir. Yeniden tanımlama riski metrikleri, satır yeniliği (row novelty) ve çıkarım saldırılarına karşı koruma gibi metrikler bu boyutta kullanılır.30

Sentetik verinin kalitesini değerlendirmek için bu üç boyutun dengeli bir şekilde ele alınması önemlidir. Yüksek doğruluk, sentetik verinin gerçek veriyi iyi temsil ettiğini gösterirken, yüksek kullanılabilirlik sentetik verinin pratik uygulamalarda faydalı olduğunu gösterir. Yüksek gizlilik ise, hassas bilgilerin korunmasını sağlar.

**Sonuç ve Öneriler**

LLM'ler, sentetik veri üretimi için güçlü ve esnek araçlar sunmaktadır. Prompt mühendisliği, çok adımlı üretim, distilasyon ve kendi kendini geliştirme gibi çeşitli yöntemler, farklı ihtiyaçlara ve senaryolara uygun çözümler sunmaktadır. Promptwright, SDV ve ydata-synthetic gibi Python kütüphaneleri, bu alanda çalışan geliştiriciler ve araştırmacılar için önemli araçlar sağlamaktadır. Medium platformu ise, bu konuda güncel bilgileri ve pratik rehberleri içeren değerli bir kaynaktır. Farklı LLM modellerinin avantaj ve dezavantajları dikkate alınarak, projenin gereksinimlerine en uygun model seçilmelidir.

Sentetik veri üretimi sürecinde etik hususlara ve veri kalitesine özen göstermek, sorumlu ve etkili yapay zeka uygulamaları geliştirmek için hayati önem taşır. Önyargıların azaltılması, gizliliğin korunması ve üretilen verinin kalitesinin düzenli olarak değerlendirilmesi gereklidir.

Sentetik veri üretimi için LLM kullanmak isteyen kullanıcılara yönelik bazı pratik öneriler şunlardır:

* Projenizin özel gereksinimlerini ve hedeflerini net bir şekilde belirleyin ve buna en uygun LLM modelini ve sentetik veri üretim yöntemini seçin.
* LLM'lerden en iyi sonuçları almak için etkili promptlar tasarlamaya özen gösterin ve gerekirse karmaşık görevler için çok adımlı üretim tekniklerini kullanın.
* İhtiyaçlarınıza uygun Python kütüphanelerini keşfedin ve projelerinizde bu araçlardan faydalanın.
* Sentetik veri üretimi ve LLM'ler konusundaki en son gelişmeleri takip etmek için Medium gibi platformlardaki güncel kaynakları düzenli olarak inceleyin.
* Ürettiğiniz sentetik verinin kalitesini, doğruluk, kullanılabilirlik ve gizlilik metriklerini kullanarak düzenli olarak değerlendirin ve etik hususlara her zaman dikkat edin.

#### Alıntılanan çalışmalar

1. Synthetic Data Generation with LLMs: What You Need to Know - Deepchecks, erişim tarihi Nisan 10, 2025, <https://www.deepchecks.com/what-to-know-synthetic-data-generation-llms/>
2. Synthetic Data Generation: Definition, Types, Techniques, & Tools - Turing, erişim tarihi Nisan 10, 2025, <https://www.turing.com/kb/synthetic-data-generation-techniques>
3. Synthetic data generation (Part 1) - OpenAI Cookbook, erişim tarihi Nisan 10, 2025, <https://cookbook.openai.com/examples/sdg1>
4. Large language models generating synthetic clinical datasets: a feasibility and comparative analysis with real-world perioperative data - Frontiers, erişim tarihi Nisan 10, 2025, <https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2025.1533508/full>
5. Synthetic Data: Benefits and Techniques for LLM Fine-Tuning in 2025, erişim tarihi Nisan 10, 2025, <https://labelyourdata.com/articles/llm-fine-tuning/synthetic-data>
6. Leveraging LLMs for Synthetic Data Generation - Deepchecks, erişim tarihi Nisan 10, 2025, <https://www.deepchecks.com/leveraging-llms-synthetic-data-generation/>
7. arxiv.org, erişim tarihi Nisan 10, 2025, <https://arxiv.org/abs/2503.14023>
8. How To Generate Synthetic Data for Fine-Tuning LLMs with AI Alignment - Medium, erişim tarihi Nisan 10, 2025, <https://medium.com/@dongchaochen/how-to-create-synthetic-data-for-fine-tuning-llms-with-ai-alignment-b7e04bb9ebdb>
9. On LLMs-Driven Synthetic Data Generation, Curation, and Evaluation: A Survey - arXiv, erişim tarihi Nisan 10, 2025, <https://arxiv.org/html/2406.15126v1>
10. On LLMs-Driven Synthetic Data Generation, Curation, and Evaluation: A Survey - ACL Anthology, erişim tarihi Nisan 10, 2025, <https://aclanthology.org/2024.findings-acl.658.pdf>
11. Synthetic Data Generation Using Large Language Models for Financial Question Answering - ACL Anthology, erişim tarihi Nisan 10, 2025, <https://aclanthology.org/2025.finnlp-1.7.pdf>
12. Using LLMs for Synthetic Data Generation: The Definitive Guide - Confident AI, erişim tarihi Nisan 10, 2025, <https://www.confident-ai.com/blog/the-definitive-guide-to-synthetic-data-generation-using-llms>
13. Synthetic Data Generation with LLMs - Towards Data Science, erişim tarihi Nisan 10, 2025, <https://towardsdatascience.com/synthetic-data-generation-with-llms/>
14. stacklok/promptwright: Generate large synthetic data using ... - GitHub, erişim tarihi Nisan 10, 2025, <https://github.com/stacklok/promptwright>
15. Welcome to the SDV! | Synthetic Data Vault, erişim tarihi Nisan 10, 2025, <https://docs.sdv.dev/sdv>
16. The Top 5 Python Packages to Generate Realistic Synthetic Data, erişim tarihi Nisan 10, 2025, <https://ydata.ai/resources/top-5-packages-python-synthetic-data>
17. statice/awesome-synthetic-data: A curated list of awesome synthetic data tools (open source and commercial). - GitHub, erişim tarihi Nisan 10, 2025, <https://github.com/statice/awesome-synthetic-data>
18. Generating synthetic data to test LLM applications | by Andrea Rosales | Medium, erişim tarihi Nisan 10, 2025, <https://medium.com/@andrea.rosales08/generating-synthetic-data-to-test-llm-applications-1c2a053cdf7a>
19. LLM-Driven Synthetic Data Generation, Curation & Evaluation | by Cobus Greyling | Medium, erişim tarihi Nisan 10, 2025, <https://cobusgreyling.medium.com/llm-driven-synthetic-data-generation-curation-evaluation-33731e33b525>
20. Three Steps to Create Synthetic Data for LLM Training Using Llama 3.1 405B, erişim tarihi Nisan 10, 2025, <https://sebastian-petrus.medium.com/three-steps-to-create-synthetic-data-for-llm-training-using-llama-3-1-405b-6a6e92d76b48>
21. Generating synthetic data with differentially private LLM inference - Google Research, erişim tarihi Nisan 10, 2025, <https://research.google/blog/generating-synthetic-data-with-differentially-private-llm-inference/>
22. We just Open Sourced Promptwright: Generate large synthetic datasets using a local LLM, erişim tarihi Nisan 10, 2025, <https://www.reddit.com/r/LocalLLaMA/comments/1ge9192/we_just_open_sourced_promptwright_generate_large/>
23. We just Open Sourced Promptwright: Generate large synthetic datasets using a local LLMGeneration - Reddit, erişim tarihi Nisan 10, 2025, <https://www.reddit.com/r/learnmachinelearning/comments/1ge9924/we_just_open_sourced_promptwright_generate_large/>
24. Generating synthetic Data with Large Language Models - Niklas Heidloff, erişim tarihi Nisan 10, 2025, <https://heidloff.net/article/generating-synthetic-data/>
25. Guide to Ethical Fine-Tuning of Large Language Models | Tonic.ai, erişim tarihi Nisan 10, 2025, <https://www.tonic.ai/guides/ethical-fine-tuning-llm-synthetic-data>
26. Ethical and Legal Considerations of Synthetic Data Usage - Keymakr, erişim tarihi Nisan 10, 2025, <https://keymakr.com/blog/ethical-and-legal-considerations-of-synthetic-data-usage/>
27. Ethical Implications of LLM Models in Decision-Making - MyScale, erişim tarihi Nisan 10, 2025, <https://myscale.com/blog/ethical-implications-llm-models-decision-making/>
28. Synthetic Data in AI: Challenges, Applications, and Ethical Implications - arXiv, erişim tarihi Nisan 10, 2025, <https://arxiv.org/html/2401.01629v1>
29. Best Practices and Lessons Learned on Synthetic Data for Language Models - arXiv, erişim tarihi Nisan 10, 2025, <https://arxiv.org/html/2404.07503v1>
30. How to evaluate synthetic data quality - Syntheticus, erişim tarihi Nisan 10, 2025, <https://syntheticus.ai/blog/how-to-evaluate-synthetic-data-quality>
31. How to Visually Evaluate Your Synthetic Data Quality? - YData, erişim tarihi Nisan 10, 2025, <https://ydata.ai/resources/how-to-visually-evaluate-your-synthetic-data-quality>
32. Synthetic Quality & Privacy Report - Gretel!, erişim tarihi Nisan 10, 2025, <https://docs.gretel.ai/optimize-synthetic-data/evaluate/synthetic-quality-privacy-report>
33. How to evaluate the quality of the synthetic data – measuring from the perspective of fidelity, utility, and privacy | AWS Machine Learning Blog, erişim tarihi Nisan 10, 2025, <https://aws.amazon.com/blogs/machine-learning/how-to-evaluate-the-quality-of-the-synthetic-data-measuring-from-the-perspective-of-fidelity-utility-and-privacy/>
34. SDMetrics - Synthetic Data Vault, erişim tarihi Nisan 10, 2025, <https://docs.sdv.dev/sdmetrics>
35. Synthetic Data Quality - Gretel.ai, erişim tarihi Nisan 10, 2025, <https://docs.gretel.ai/optimize-synthetic-data/evaluate/synthetic-data-quality-report>