

2026 Teknik SEO ve Yapay Zeka Entegrasyon Mimarisi: Kapsamlı Analiz, Onarım ve Veri Mühendisliği Raporu

1. Yönetici Özeti ve Yeni Nesil Arama Paradigması

Dijital bilgi erişim sistemleri, geleneksel arama motoru dizinlemesinden (indexing), büyük dil modellerinin (LLM) anlık çıkarımına (inference) dayalı yanıt üretme süreçlerine doğru geri döndürülemez bir evrim geçirmektedir. 2025 yılı itibarıyla, "Arama Motoru Optimizasyonu" (SEO) kavramı, yerini "Cevap Motoru Optimizasyonu" (AEO) ve "Üretken Motor Optimizasyonu" (GEO) kavramlarına bırakmaktadır.¹ Bu rapor, yalnızca web sitelerinin Googlebot gibi geleneksel tarayıcılar tarafından nasıl daha iyi anlaşılacağını değil, aynı zamanda site verilerinin yapay zeka modelleri için nasıl yüksek kaliteli bir "eğitim veri seti" (training dataset) haline getirileceğini detaylandıran kapsamlı bir teknik başvuru kaynağıdır.

Geleneksel SEO, anahtar kelimelerin frekansına ve geri bağlantı (backlink) sayısına odaklanırken; modern teknik SEO, veri bütünlüğü, semantik yapılandırma, sunucu yanıt performansı ve taranabilirlik bütçesi (crawl budget) optimizasyonu üzerine kuruludur. SearchGPT, Perplexity ve Google AI Overviews (SGE) gibi sistemler, web sitelerini sadece birer URL listesi olarak değil, doğrulanmış bilgi parçacıklarının (facts) kaynağı olarak görmektedir. Bu bağlamda, teknik altyapıdaki her hata (örneğin; yanlış canonical kullanımı veya eksik hiyerarşik yapı), sitenizin sadece sıralama kaybetmesine değil, yapay zeka tarafından üretilen yanıtlarda tamamen görmezden gelinmesine veya yanlış temsil edilmesine (halüsinasyon) yol açmaktadır.³

Bu rapor, kullanıcı sorgusunda belirtilen tüm teknik gereksinimleri (hataların tespiti, kod düzeyinde onarımı, Semrush benzeri derinlikte analiz yeteneği ve AI eğitimi için veri hazırlığı) karşılamak üzere dokuz ana bölümde yapılandırılmıştır. Her bölüm, teorik altyapıyı pratik uygulama kodları ve stratejik öngörülerle birleştirmektedir.

2. Sunucu Altyapısı ve Erişim Protokolleri

Web sitenizin teknik sağlığı, CMS (İçerik Yönetim Sistemi) seviyesinden önce, sunucu seviyesinde başlar. Arama motoru botları ve AI ajanları sitenize ulaştığında ilk karşılaştıkları katman burasıdır. Bu katmandaki hatalar, içeriğiniz ne kadar kaliteli olursa olsun, erişimi tamamen engelleyebilir.

2.1. HTTP Durum Kodları ve İleri Seviye Yönlendirme Stratejileri

HTTP durum kodları, sunucunuzun tarayıcıyla konuştuğu dildir. Bu kodların yanlış yönetimi, tarama bütçesinin israfına ve dizinleme hatalarına neden olur.⁵

2.1.1. Kalıcı Yönlendirmeler (301) ve Zincirleme Sorunları

301 yönlendirmesi, bir kaynağın kalıcı olarak taşındığını belirtir. Ancak, sık yapılan bir hata "Yönlendirme Zincirleri" (Redirect Chains) oluşturmaktır. Örneğin; Sayfa A -> Sayfa B -> Sayfa C şeklinde bir zincir, botların taramayı yarıda kesmesine neden olur. Her yönlendirme milisaniyeler süren bir gecikme (latency) yaratır ve tarama bütçesinden yer.

Çözüm Stratejisi: Zincirdeki tüm ara adımları atlayarak, Sayfa A'yı doğrudan Sayfa C'ye yönlendirmelisiniz.

Apache (.htaccess) için Doğrudan 301 Yönlendirme: Aşağıdaki kod, eski URL yapısını yeni yapıya, query string parametrelerini koruyarak yönlendirir.⁶

Apache

```
RewriteEngine On
# Redirect Chain Önleme: Tek adımda nihai hedefe git
RewriteCond %{REQUEST_URI} ^/eski-kategori/(.*)$
RewriteRule ^(.*)$ https://www.siteniz.com/yeni-kategori/$1
```

2.1.2. WWW ve Protokol Tutarlılığı (Canonicalization at Server Level)

Sitenizin http://, https://, http://www. ve https://www. versiyonlarının hepsi erişilebilir durumdaysa, arama motorları bunları dört farklı site olarak algılar ve "Duplicate Content" (Kopya İçerik) cezası uygular. Tek bir versiyonu (tercihen https://www veya https://non-www) zorunlu kılmalısınız.⁸

Nginx Konfigürasyonu (Force HTTPS & WWW): Nginx sunucularda bu işlem server blokları ile yapılır. Bu yöntem.htaccess'e göre daha performanslıdır çünkü sunucu her istekte dosyayı yeniden okumak zorunda kalmaz.¹⁰

Nginx

```
# HTTP'den HTTPS'e yönlendirme bloğu
server {
```

```
listen 80;
server_name example.com www.example.com;
# 301 Kalıcı yönlendirme
return 301 https://www.example.com$request_uri;
}

# Non-WWW'den WWW'ye yönlendirme (HTTPS)
server {
    listen 443 ssl;
    server_name example.com;

    ssl_certificate /etc/ssl/certs/example_com.crt;
    ssl_certificate_key /etc/ssl/private/example_com.key;

    return 301 https://www.example.com$request_uri;
}
```

2.2. Karışık İçerik (Mixed Content) ve Güvenlik

SSL sertifikası kurulu olsa dahi, sayfa kaynağında http:// protokolü ile çağrılan bir görsel veya script varsa, tarayıcı "Güvenli Değil" uyarısı verir. Bu durum kullanıcı güvenliğini zedeler ve sıralama kaybına yol açar.

Otomatik Düzeltme (Content-Security-Policy): Binlerce sayfadaki http linklerini tek tek düzeltmek yerine, tarayıcıya "tüm güvensiz istekleri güvenli (https) olarak yükle" talimatı verebilirsiniz. Bu, sunucu yanıt başlığına (response header) eklenerek yapılır.¹²

Apache (.htaccess) Kodu:

Apache

```
<IfModule mod_headers.c>
    Header always set Content-Security-Policy "upgrade-insecure-requests;"
</IfModule>
```

Bu kod, sayfa içindeki http://site.com/resim.jpg isteğini tarayıcı seviyesinde https://site.com/resim.jpg olarak değiştirir.

2.3. Özel 404 Sayfa Yönetimi

Bir sayfa bulunamadığında sunucu 404 Not Found durum kodu döndürmelidir. Ancak bazı

sistemler, kullanıcıya "Sayfa Bulunamadı" yazan ama teknik olarak 200 OK kodu döndüren sayfalar sunar. Bu duruma "Soft 404" denir ve ciddi bir SEO hatasıdır; çünkü Google bu boş sayfaları dizine eklemeye çalışır.¹⁴

.htaccess ile Doğru 404 Yapılandırması:

Apache

ErrorDocument 404 /404.html

Bu komut, sunucunun kullanıcıya özel tasarlanmış /404.html dosyasını gösterirken, botlara teknik olarak 404 durum kodunu iletmesini sağlar.

3. Tarama Bütçesi ve Bot Yönetimi (Crawlability)

Web, milyarlarca sayfadan oluşur ve Googlebot'un kaynakları sınırsız değildir. Sitenize ayrılan "Tarama Bütçesi"ni (Crawl Budget) verimli kullanmak, önemli sayfalarınızın daha hızlı güncellenmesini sağlar. Ayrıca, içeriğinizi SearchGPT gibi yapay zeka motorlarına açıp açmama kararı, 2025'in en kritik stratejik kararıdır.

3.1. Robots.txt: AI Çağında Erişim Kontrolü

Robots.txt dosyası, sitenizin sınır kapısıdır. Geçmişte sadece Googlebot'u yönetmek yeterliyken, bugün verilerinizin LLM eğitimi için kullanılıp kullanılmayacağını da buradan kontrol edersiniz.¹⁶

Bot Türleri ve Stratejik Kararlar

Bot Tipi	Örnekler	Amaç	İzin Verilmeli mi?
Arama Motoru	Googlebot, Bingbot	Sıralama ve Trafik	EVET (Kritik)
AI Cevap Motoru	GPTBot, PerplexityBot	SearchGPT/Perplexity'de kaynak gösterilme	EVET (Trafik potansiyeli var)
AI Eğitim Botu	Google-Extended,	LLM model eğitimi (Trafik garantisi	STRATEJİK (Veri gizliliği vs

	ClaudeBot	yok)	Görünürlük)
Veri Kazıyıcı	CCBot, Bytespider	Genel veri setleri oluşturma	HAYIR (Genellikle sunucu yükü yaratır)

Örnek Optimize Edilmiş Robots.txt Dosyası: Bu dosya, arama motorlarına ve trafik getiren AI botlarına izin verirken, sadece veri toplayan kazıyıcıları engeller.¹⁸

Plaintext

```
User-agent: *
Allow: /
# Yönetim panellerini gizle
Disallow: /wp-admin/
Disallow: /admin/
# Parametrelili URL'leri engelle (Tarama bütçesi tasarrufu)
Disallow: /*?sort=
Disallow: /*&filter=
```

```
# SearchGPT (OpenAI) - Trafik kaynağı olabilir
User-agent: GPTBot
Allow: /
```

```
# Google AI Overviews (SGE) - Trafik kaynağı
User-agent: Google-Extended
Allow: /
```

```
# Common Crawl (Genel veri seti) - Genellikle engellenir
User-agent: CCBot
Disallow: /
```

```
# Anthropic (Claude)
User-agent: ClaudeBot
Disallow: /
```

```
Sitemap: https://www.siteniz.com/sitemap_index.xml
```

3.2. Log Dosyası Analizi ve Regex ile Bot Doğrulama

Semrush gibi araçlar dışarıdan bir simülasyon yapar. Ancak sunucu logları (access logs), sitenize *gerçekte* kimin geldiğini gösteren tek kesin veridir. Log analizi ile "sahte Googlebot" saldırılarını ve tarama bütçesi israfını tespit edebilirsiniz.²⁰

Regex (Düzenli İfadeler) ile Analiz:

Log dosyalarını analiz etmek için grep veya Python kullanabilirsiniz. Sahte botlar, User-Agent kısmına "Googlebot" yazar ancak IP adresleri Google'a ait değildir.

Sahte Bot Tespit Algoritması (Kavramsal):

1. User-Agent içinde "Googlebot" geçen satırları filtrele.
2. Bu satırlardaki IP adreslerini al.
3. Reverse DNS (rDNS) sorgusu yap. Gerçek Googlebot IP'leri her zaman googlebot.com veya google.com ile biter.
4. Eşleşmeyenleri güvenlik duvarı (firewall) ile engelle.²²

Python ile Log Analiz Scripti (Basitleştirilmiş):

Python

```
import re
```

```
log_line = '66.249.66.1 - - [24/Jan/2026:10:00:00] "GET /url HTTP/1.1" 200 500 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"'
```

```
# Googlebot tespiti için Regex
```

```
if re.search(r'Googlebot', log_line):
```

```
    # IP adresini ayıkla
```

```
    ip = log_line.split()
```

```
    # Burada rDNS kontrolü yapılmalı (socket kütüphanesi ile)
```

```
    print(f"Googlebot detected from IP: {ip}")
```

3.3. XML Site Haritası Optimizasyonu

Büyük siteler için tek bir sitemap.xml yetersizdir. 50.000 URL veya 50MB dosya boyutu sınırı vardır. Ayrıca, görsel ve video içerikler için özel etiketler (tag) kullanılmalıdır.²³

Sitemap Index (Dizin) Stratejisi:

Ana site haritası, diğer alt haritaların listesini barındırmalıdır.

- sitemap-products.xml
- sitemap-blog.xml
- sitemap-images.xml

Görsel Odaklı Site Haritası Örneği:

Bu yapı, Google Görseller ve Discover trafiği için kritiktir.

XML

```
<url>
  <loc>https://www.siteniz.com/urun-detay</loc>
  <image:image>
    <image:loc>https://www.siteniz.com/images/urun-buyuk.jpg</image:loc>
    <image:title>Ürün Adı ve Modeli</image:title>
    <image:caption>Ürünün teknik detaylarını gösteren yakın çekim.</image:caption>
  </image:image>
</url>
```

4. Sayfa İç Teknik Mimari (On-Page Architecture)

Bu bölüm, kullanıcının "canonical etiketi yoksa nasıl eklenir", "H2 etiketi yoksa nasıl düzeltilir" gibi spesifik sorularına yanıt verir. Bu öğeler, hem SEO hem de yapay zeka eğitimi için verinin yapılandırılması açısından hayati önem taşır.

4.1. Canonical Etiketleri ve Kopya İçerik Yönetimi

Canonical etiketi, arama motorlarına "Bu içeriğin orijinali budur" deme yöntemidir. Özellikle e-ticaret sitelerinde filtreler (?color=red, ?sort=price) nedeniyle oluşan binlerce URL varyasyonunu yönetmek için kullanılır.²⁵

Sorun: Canonical etiketinin eksik olması veya yanlış yapılandırılması.

Çözüm: Her sayfada, kendi kendine referans veren (self-referencing) bir canonical etiketi <head> bölümünde olmalıdır.

HTML Kodu:

HTML

```
<head>
  <link rel="canonical" href="https://www.siteniz.com/kategori/ana-urun-sayfasi" />
</head>
```

Not: Eğer sayfada ?renk=kirmizi gibi bir parametre varsa, canonical etiketi yine de parametresiz ana URL'i (.../ana-urun-sayfasi) göstermelidir.

Dinamik Olarak Canonical Oluşturma (PHP/WordPress Functions):

Eğer WordPress kullanıyorsanız ve temanızda bu özellik yoksa, header.php veya functions.php dosyasına şu mantığı ekleyebilirsiniz:

PHP

```
function add_canonical_tag() {
    // Mevcut URL'yi al (query string olmadan)
    $protocol = is_ssl()? 'https' : 'http';
    $url = $protocol. '://'. $_SERVER. parse_url($_SERVER, PHP_URL_PATH);

    echo '<link rel="canonical" href="'. esc_url($url). "' />'. "\n";
}
add_action('wp_head', 'add_canonical_tag');
```

4.2. Başlık Hiyerarşisi (Heading Vectors)

Yapay zeka modelleri (LLM), metinleri vektörlere dönüştürürken başlık yapılarını (H1, H2, H3) birer "çapa" (anchor) olarak kullanır. Başlık hiyerarşisinin bozuk olması (örneğin H1'den direkt H4'e geçilmesi), içeriğin anlamsal bütünlüğünü bozar ve modelin konuyu kavramasını zorlaştırır.²⁷

Hata: H2 etiketinin kullanılmaması.

Düzeltilme ve Uygulama:

İçeriğinizi bir kitap gibi düşünün. H1 kitap adı, H2 bölüm adları, H3 ise bölüm içindeki alt

başlıklardır.

HTML Yapısı:

HTML

```
<article>
  <h1>Ana Konu (Sayfanın en önemli başlığı)</h1>
  <p>Giriş metni...</p>

  <h2>Ana Konunun Alt Bölümü 1</h2>
  <p>Bu bölüm hakkında detaylar...</p>

  <h3>Detayın Detayı (H3)</h3>
  <p>Spesifik veri...</p>

  <h2>Ana Konunun Alt Bölümü 2</h2>
  <p>Diğer detaylar...</p>
</article>
```

Görsel Düzenleme (CSS):

Tasarımcılar bazen "H2 çok büyük görünüyor" diyerek kullanmaktan kaçınır. Bu yanlıştır; HTML semantik, CSS ise estetikdir. H2 kullanın ama CSS ile boyutunu ayarlayın.

CSS

```
h2 {
  font-size: 24px; /* Boyutu tasarımınıza göre ayarlayın */
  font-weight: 700;
  color: #333333;
  margin-top: 20px;
}
```

4.3. Eksik Alt Etiketleri (Image Alt Text)

Görseller, yapay zeka (Computer Vision) tarafından analiz edilse de, alt etiketi içeriğin

bağlamını (context) belirleyen en güçlü sinyaldir. Boş bırakılan alt etiketleri, erişilebilirlik (accessibility) puanını düşürür ve görsel SEO fırsatlarını yok eder.²⁸

Otomatik Tespit ve Düzeltme Mantiğı:

Büyük sitelerde eksik alt etiketlerini bulmak için tarayıcı konsolunda veya Python scriptleri ile tarama yapabilirsiniz.

HTML Düzeltmesi:

HTML

```

```

```

```

Dekoratif Görseller: Eğer görsel sadece süs amaçlıysa (örneğin bir ayırıcı çizgi), alt="" şeklinde boş bırakılmalıdır. Bu, ekran okuyuculara "bunu okuma" sinyali verir.³⁰

5. İleri Seviye Semantik SEO ve Schema İşaretleme

Arama motorlarının ve yapay zekanın sitenizi "anlaması" için ona yapılandırılmış veri (Structured Data) sunmalısınız. Bu, içeriğinizi makine tarafından okunabilir bir formatta (JSON-LD) sunmak demektir.

5.1. JSON-LD ile Zengin Sonuçlar (Rich Snippets)

2025'te Schema, sadece yıldızlı değerlendirmeler için değil, AI ajanlarının sitenizden veri çekebilmesi için bir API görevi görür. Özellikle FAQPage, Article ve Organization şemaları kritiktir.³¹

Soru-Cevap (FAQ) Şeması ve AI Entegrasyonu

FAQ şeması, SearchGPT gibi sistemlerin doğrudan yanıt üretebilmesi için en uygun formattır. Soruyu ve cevabı net bir şekilde etiklediğinizde, modelin halüsinasyon görme ihtimalini azaltırsınız.³³

Kod Örneği (Sayfanın <head> veya <body> sonuna eklenmeli):

HTML

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "FAQPage",
  "mainEntity":
}
</script>
```

5.2. Schema Doğrulama ve Test Araçları

Kodunuzu yazdıktan sonra mutlaka doğrulamanız gerekir. Hatalı JSON-LD kodu, sayfanın tamamen işlenmesini engelleyebilir.

- **Schema Markup Validator:** validator.schema.org (Sözdizimi hatalarını kontrol eder).³¹
- **Rich Results Test:** search.google.com/test/rich-results (Google'ın bu şemayı destekleyip desteklemediğini kontrol eder).³⁵

6. JavaScript SEO ve Performans Optimizasyonu (INP)

Google'ın Core Web Vitals metriklerinde FID (First Input Delay) yerini **INP (Interaction to Next Paint)** metriğine bırakmıştır. INP, kullanıcının bir butona tıkladığında görsel tepkiyi ne kadar sürede gördüğünü ölçer. Bu, JavaScript yoğun siteler için en büyük teknik zorluktur.³⁶

6.1. INP Sorunlarını Kod ile Çözme (yieldToMain)

Kötü bir INP skoru (200ms üzeri), genellikle tarayıcının "Ana Thread"inin (Main Thread) uzun süren JavaScript işlemleriyle meşgul olmasından kaynaklanır. Tarayıcı meşgulken kullanıcı tıklarsa, tarayıcı yanıt veremez.

Çözüm: Ana Akışa Yol Verme (Yielding to Main Thread) Uzun süren işlemleri (Long Tasks) küçük parçalara bölerek, araya kullanıcı etkileşiminin girmesine izin vermelisiniz.³⁸

Modern JavaScript Çözümü (Async/Await):

JavaScript

```
// Yardımcı fonksiyon: İşlemi milisaniyelerce duraklatarak tarayıcıya nefes aldırır
function yieldToMain() {
  return new Promise(resolve => {
    setTimeout(resolve, 0);
  });
}

async function heavyTaskProcessor(dataList) {
  for (let i = 0; i < dataList.length; i++) {
    processItem(dataList[i]); // İşlemi yap

    // Her 50 işlemde bir durakla ve ana thread'e kontrolü ver
    // Böylece kullanıcı o an tıklarsa tarayıcı cevap verebilir
    if (i % 50 === 0) {
      await yieldToMain();
    }
  }
}
```

6.2. Rendering Stratejileri (SSR vs CSR)

Yapay zeka botları JavaScript çalıştırabilir ancak bu maliyetlidir ve her zaman garanti değildir. İçeriğinizi sunmak için en güvenli yol **SSR (Server-Side Rendering)** kullanmaktır.⁴⁰

- **SSR (Önerilen):** Next.js gibi frameworkler ile HTML sunucuda oluşturulur. Botlar tam içeriği görür.
- **CSR (Client-Side Rendering):** İçerik tarayıcıda yüklenir. Botlar bazen boş sayfa görebilir.
- **Next.js Metadata API:** Next.js 13+ sürümünde SEO etiketlerini yönetmek için head yerine Metadata API kullanılmalıdır.

Next.js Metadata Örneği:

JavaScript

```
export const metadata = {
  title: 'Teknik SEO Rehberi',
  description: '2026 Teknik SEO stratejileri...',
  alternates: {
    canonical: 'https://site.com/teknik-seo',
  },
  openGraph: {
```

```
images: ['/og-image.jpg'],
},
}
```

7. Platform Bazlı Spesifik Çözümler

Her altyapının kendine has kronik teknik SEO sorunları vardır. Aşağıda en popüler platformlar için "nasıl düzeltilir" rehberi bulunmaktadır.

7.1. WordPress: Güvenlik ve Performans

WordPress, varsayılan olarak sürüm bilgisini kaynak kodda gösterir. Bu bir güvenlik açığıdır.

Çözüm (functions.php):

PHP

```
function remove_wp_version() {
    return "";
}
add_filter('the_generator', 'remove_wp_version');
```

Ayrıca, gereksiz sorgu parametrelerini (?ver=x.x) CSS/JS dosyalarından kaldırmak yükleme hızını artırır.⁴²

7.2. Shopify: URL Yapısı ve Kopya İçerik

Shopify, aynı ürünü /products/urun-adi ve /collections/kategori/products/urun-adi olarak iki farklı URL'de sunar. Bu ciddi bir kopya içerik sorunudur.⁴⁴ **Çözüm (theme.liquid):** Canonical etiketinin her zaman "kısa" ve "temiz" URL'i gösterdiğinden emin olun.

Kod snippet'i

```
<link rel="canonical" href="{{ product.url }}" />
```

Ayrıca, kategori sayfalarındaki filtre kombinasyonlarını (tag sayfaları) robots.txt üzerinden engelleyerek tarama bütçesini koruyun.

7.3. Nginx Sunucular: Gzip Sıkıştırma

Sunucu yanıt süresini (TTFB) düşürmek için metin dosyalarının sıkıştırılması gerekir.

Nginx Konfigürasyonu:

Nginx

```
gzip on;  
gzip_types text/plain text/css application/json application/javascript text/xml application/xml;  
gzip_min_length 1000;
```

Bu ayar, dosyaları sunucudan çıkmadan önce sıkıştırarak veri transfer boyutunu %70'e kadar azaltır.⁴⁵

8. Yapay Zeka Eğitimi İçin Veri Hazırlama (Data Engineering for LLMs)

Kullanıcının talebi üzerine, bu belgenin en kritik ve yenilikçi bölümü burasıdır. Web sitenizdeki verileri, bir yapay zeka modelini (GPT-4, Llama 3 vb.) eğitmek (fine-tuning) veya bir RAG (Retrieval-Augmented Generation) sistemine beslemek için nasıl hazırlamalısınız?

Geleneksel HTML, yapay zeka eğitimi için "kirli" veridir. HTML etiketleri, CSS sınıfları ve navigasyon menüleri, modelin öğrenme sürecinde gürültü (noise) yaratır. Veriyi **JSONL (JSON Lines)** formatına dönüştürmeniz gerekir.⁴⁶

8.1. HTML'den Dataset Oluşturma Süreci

Sitenizdeki "Hatalar ve Çözümleri" veritabanını bir yapay zeka asistanına dönüştürmek istediğinizi varsayalım.

Adım 1: Veri Temizleme (Sanitization)

HTML içeriğindeki <div>, , class="..." gibi yapıları temizleyin. Sadece saf metni (plain text) ve anlamsal yapıyı koruyun.

Adım 2: Chunking (Parçalama)

Uzun metinleri, konuya göre mantıksal parçalara ayırın. Her bir H2 başlığı ve altındaki paragraf, tek bir eğitim verisi olabilir.

8.2. Fine-Tuning İçin JSONL Formatı

OpenAI ve diğer modeller, genellikle "Sohbet" formatında (Chat format) veri bekler. Her satır bağımsız bir JSON objesidir.

Örnek Eğitim Verisi (JSONL Dosyası):

JSON

```
{"messages":}  
{"messages":}
```

8.3. Veri Seti Kalite Kontrolü

Eğitim verisi oluştururken dikkat edilmesi gerekenler ⁴⁸:

1. **Tutarlılık:** Cevapların formatı hep aynı olmalı (örneğin önce tanım, sonra kod örneği).
2. **Bağlam (Context):** Sistem mesajında ("system role") AI'ya kim olduğu net tanımlanmalı.
3. **Token Limiti:** Her bir örneğin modelin token limitini aşmadığından emin olun (genellikle 4096 veya 8192 token).

9. Metodoloji ve Kendi Denetim Sisteminizi Kurma

Semrush veya Ahrefs gibi araçlar olmadan da, bu raporda anlatılan yöntemlerle kendi "Site Audit" (Site Denetim) sisteminizi kurabilirsiniz.

9.1. Manuel Denetim Scripti (Python Konsepti)

Python'un requests ve BeautifulSoup kütüphanelerini kullanarak sitenizi tarayan ve hataları raporlayan basit bir bot yazabilirsiniz.

- **Status Code Kontrolü:** Her URL'ye istek atıp response.status_code 200 mü kontrol et.
- **Canonical Kontrolü:** HTML içinde link[rel='canonical'] var mı bak.
- **Hiyerarşi Kontrolü:** Sayfada h1 sayısı 1'den fazla mı veya hiç yok mu kontrol et.

Bu rapor, teknik SEO'nun sadece bir "onarım" süreci değil, aynı zamanda bir "veri mühendisliği" süreci olduğunu ortaya koymaktadır. Sitenizi bu standartlara göre optimize

ettiğinizde, hem bugünün Google sıralamalarında yükselirsiniz hem de geleceğin yapay zeka modelleri için "referans kaynak" haline gelirsiniz.

Alıntılanan çalışmalar

1. How to Do SEO for SearchGPT and Rank #1, erişim tarihi Ocak 24, 2026, <https://www.airankingskool.com/post/how-to-do-seo-for-searchgpt>
2. Tech SEO in GEO: Why Your Technical Setup Is Essential for LLM Visibility - Webrepublic, erişim tarihi Ocak 24, 2026, <https://www.webrepublic.com/en/insights/geo>
3. Block AI Bots from Crawling Websites Using Robots.txt - Originality.ai, erişim tarihi Ocak 24, 2026, <https://originality.ai/ai-bot-blocking>
4. I Tested LLM SEO for Months. Here's How to Rank in AI Generated Results : r/seogrowth, erişim tarihi Ocak 24, 2026, https://www.reddit.com/r/seogrowth/comments/1jdmct3/i_tested_llm_seo_for_months_heres_how_to_rank_in/
5. Site Audit Statistics - Semrush, erişim tarihi Ocak 24, 2026, <https://www.semrush.com/kb/544-site-audit-statistics>
6. How to Redirect WWW to non WWW WordPress [htaccess snippet] - Bootstrap Creative, erişim tarihi Ocak 24, 2026, <https://bootstrapcreative.com/how-to-redirect-www-to-non-www-wordpress/>
7. How to redirect an https site from www to non www - Webmasters Stack Exchange, erişim tarihi Ocak 24, 2026, <https://webmasters.stackexchange.com/questions/129738/how-to-redirect-an-https-site-from-www-to-non-www>
8. Generic htaccess redirect www to non-www - Stack Overflow, erişim tarihi Ocak 24, 2026, <https://stackoverflow.com/questions/234723/generic-htaccess-redirect-www-to-non-www>
9. Force www. and https in nginx.conf (SSL) - Stack Overflow, erişim tarihi Ocak 24, 2026, <https://stackoverflow.com/questions/32486739/force-www-and-https-in-nginx-conf-ssl>
10. Advanced configuration with Snippets | NGINX Documentation, erişim tarihi Ocak 24, 2026, <https://docs.nginx.com/nginx-ingress-controller/configuration/ingress-resources/advanced-configuration-with-snippets/>
11. Configuring HTTPS servers - nginx, erişim tarihi Ocak 24, 2026, https://nginx.org/en/docs/http/configuring_https_servers.html
12. Fixing Mixed content via .htaccess | KnowledgeBase - ResellerClub, erişim tarihi Ocak 24, 2026, <https://kb.resellerclub.com/article/Fixing-Mixed-content-via-htaccess>
13. Fixing mixed content | Articles - web.dev, erişim tarihi Ocak 24, 2026, <https://web.dev/articles/fixing-mixed-content>
14. How to Set a Custom 404 Page Using .htaccess | InMotion Hosting, erişim tarihi

- Ocak 24, 2026,
<https://www.inmotionhosting.com/support/website/error-numbers/set-custom-404-page/>
15. htaccess not redirecting to custom 404 page - Stack Overflow, erişim tarihi Ocak 24, 2026,
<https://stackoverflow.com/questions/38437101/htaccess-not-redirecting-to-custom-404-page>
 16. Using Robots.txt to Disallow or Allow Bot Crawlers - DataDome, erişim tarihi Ocak 24, 2026,
<https://datadome.co/bot-management-protection/blocking-with-robots-txt/>
 17. Create and Submit a robots.txt File | Google Crawling Infrastructure, erişim tarihi Ocak 24, 2026,
<https://developers.google.com/crawling/docs/robots-txt/create-robots-txt>
 18. Robots.txt - Guide for AI ranking, erişim tarihi Ocak 24, 2026,
<https://www.botrank.ai/technical-doc/robots-txt>
 19. How to prevent ChatGPT from crawling your website - Classic - RapidWeaver Forum, erişim tarihi Ocak 24, 2026,
<https://forums.realmacsoftware.com/t/how-to-prevent-chatgpt-from-crawling-our-website/42591>
 20. Advanced Log File Analysis for SEO, erişim tarihi Ocak 24, 2026,
<https://seohq.github.io/advanced-log-file-analysis>
 21. Using Log File Analysis To Improve Your SEO Performance - Matthew Edgar, erişim tarihi Ocak 24, 2026, <https://www.matthewedgar.net/log-file-analysis/>
 22. How to determine if a user agent string has proper syntax or might be a hacking attempt?, erişim tarihi Ocak 24, 2026,
<https://webmasters.stackexchange.com/questions/90074/how-to-determine-if-a-user-agent-string-has-proper-syntax-or-might-be-a-hacking>
 23. Build and Submit a Sitemap | Google Search Central | Documentation, erişim tarihi Ocak 24, 2026,
<https://developers.google.com/search/docs/crawling-indexing/sitemaps/build-site-map>
 24. Your guide to sitemaps: best practices for crawling and indexing - Search Engine Land, erişim tarihi Ocak 24, 2026, <https://searchengineland.com/guide/sitemap>
 25. Canonical Tags Explained: Why They Matter For SEO - Ahrefs, erişim tarihi Ocak 24, 2026, <https://ahrefs.com/blog/canonical-tags/>
 26. How to specify a canonical URL with rel="canonical" and other methods, erişim tarihi Ocak 24, 2026,
<https://developers.google.com/search/docs/crawling-indexing/consolidate-duplicate-urls>
 27. A Practical Guide to LLM SEO Optimization - LLMrefs, erişim tarihi Ocak 24, 2026,
<https://llmrefs.com/blog/llm-seo-optimization>
 28. Missing alternative text - Pope Tech Result Documentation, erişim tarihi Ocak 24, 2026,
<https://app.pope.tech/result-documentation/articles/missing-alternative-text>
 29. How to Find and Fix Missing Alt Text [Image Alt Text Checkers Included] - SEO

- PowerSuite, erişim tarihi Ocak 24, 2026,
<https://www.link-assistant.com/news/missing-alt-text.html>
30. Accessibility: Image Alt text best practices - Siteimprove Support, erişim tarihi Ocak 24, 2026,
<https://help.siteimprove.com/support/solutions/articles/80000863904-accessibility-image-alt-text-best-practices>
 31. How Can Schema Markup Specifically Enhance LLM Visibility | Walker Sands, erişim tarihi Ocak 24, 2026,
<https://www.walkersands.com/about/blog/how-can-schema-markup-support-llm-visibility/>
 32. Organization Schema Markup | Google Search Central | Documentation, erişim tarihi Ocak 24, 2026,
<https://developers.google.com/search/docs/appearance/structured-data/organization>
 33. FAQ Schema for AI Answers | Setup Guide & Examples - Passionfruit SEO, erişim tarihi Ocak 24, 2026,
<https://www.getpassionfruit.com/blog/faq-schema-for-ai-answers>
 34. Mark Up FAQs with Structured Data | Google Search Central | Documentation, erişim tarihi Ocak 24, 2026,
<https://developers.google.com/search/docs/appearance/structured-data/faqpage>
 35. Schema Markup Testing Tool | Google Search Central, erişim tarihi Ocak 24, 2026,
<https://developers.google.com/search/docs/appearance/structured-data>
 36. How to Optimize Interaction to Next Paint (INP) - web.dev, erişim tarihi Ocak 24, 2026,
<https://web.dev/explore/how-to-optimize-inp>
 37. INP Optimization: Complete Guide to Interaction to Next Paint | 2026 - LinkGraph, erişim tarihi Ocak 24, 2026,
<https://www.linkgraph.com/blog/interaction-to-next-paint-optimization/>
 38. Understanding and Improving Interaction to Next Paint (INP) - SpeedCurve, erişim tarihi Ocak 24, 2026,
<https://www.speedcurve.com/web-performance-guide/understanding-and-improving-interaction-to-next-paint/>
 39. How To Improve INP: Yield Patterns | Jacob 'Kurt' Groß - kurtextrem, erişim tarihi Ocak 24, 2026,
<https://kurtextrem.de/posts/improve-inp>
 40. The Complete Next.js SEO Guide for Building Fast and Crawlable Apps - Strapi, erişim tarihi Ocak 24, 2026,
<https://strapi.io/blog/nextjs-seo>
 41. Rendering Strategies - SEO - Next.js, erişim tarihi Ocak 24, 2026,
<https://nextjs.org/learn/seo/rendering-strategies>
 42. How to Quickly Hide WordPress Version Number - Rank Math, erişim tarihi Ocak 24, 2026,
<https://rankmath.com/blog/hide-wordpress-version/>
 43. The RIGHT Way to Remove WordPress Version Number - WPBeginner, erişim tarihi Ocak 24, 2026,
<https://www.wpbeginner.com/wp-tutorials/the-right-way-to-remove-wordpress-version-number/>
 44. 7 Core Technical SEO Challenges For Shopify Websites & How To Fix Them - blubolt, erişim tarihi Ocak 24, 2026,

<https://blubolt.com/insights/7-technical-seo-challenges>

45. Module ngx_http_gzip_module - nginx, erişim tarihi Ocak 24, 2026,
http://nginx.org/en/docs/http/ngx_http_gzip_module.html
46. Prepare your training datasets for fine-tuning and continued pre-training - Amazon Bedrock, erişim tarihi Ocak 24, 2026,
<https://docs.aws.amazon.com/bedrock/latest/userguide/model-customization-prepare.html>
47. Tool for formatting fine-tuning data into proper json? : r/LocalLLaMA - Reddit, erişim tarihi Ocak 24, 2026,
https://www.reddit.com/r/LocalLLaMA/comments/18rc8fp/tool_for_formatting_finetuning_data_into_proper/
48. Best Practices for JSON Formatting and Code Readability - Medium, erişim tarihi Ocak 24, 2026,
<https://medium.com/@AlexanderObregon/best-practices-for-json-formatting-and-code-readability-f5e769e3d051>