Etkili Prompt Tasarım Yöntemleri

Eğitmen
Kubilay Tuna

Ders İçeriği

1. Prompt Mühendisliği ve Prompting Nedir?
2. Etkili Prompt Tasarımı ve Önemi
3. Temel Prompt Türleri ve Kullanım Alanları
4. Teknik Bağlam ve Yapısal İpuçları
5. Prompt Kalitesini Test Etme Yöntemleri
6. Prompt Güvenliği ve Risk Yönetimi

Prompt Mühendisliği ve Prompting Nedir?

LLM’leri etkili bir şekilde yönetmenin anahtarı!

Prompting Nedir?
Modelin görevini anlayıp yerine getirmesi için verilen talimatlar.

Doğru Prompt Yazımının Önemi

Netlik: Belirsizlikten kaçınılmalı, talimat açıkça belirtilmeli.

Bağlam: Modelin daha iyi yanıt verebilmesi için ilgili bilgiler sağlanmalı.

Kısalık: Gereksiz uzunluklardan kaçınılmalı, model sadece gerekli bilgiyi almalı.

Zorluklar ve Dikkat Edilmesi Gerekenler

Anlamlı sonuçlar almak için bağlamın doğru verilmesi: Eksik veya yanlış bağlam modellerin anlamlı sonuçlar üretmesini engeller.

Yanlış Prompt sonuçları: Yanlış veya belirsiz prompt'lar modelin beklenmedik ve yanlış sonuçlar üretmesine neden olabilir.

Etkili Prompt Tasarımı ve Önemi

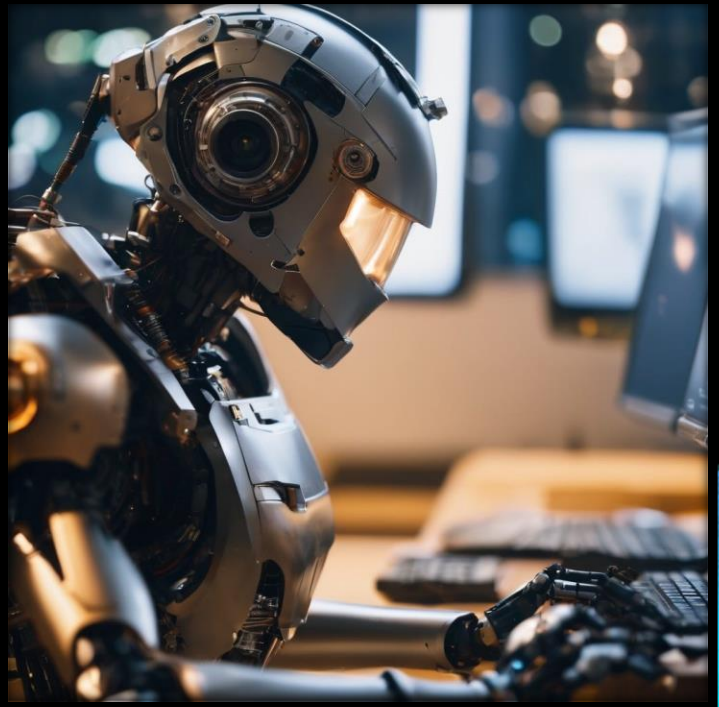
Modelin ne istediğinizi tahmin etmesi ne kadar az olursa, isteğinizi elde etme olasılığınız o kadar artar!

Açık uçlu sorular: Daha yaratıcı ve geniş yanıtlar almak için kullanılır. Modelin esnekliği artar.

Spesifik talimatlar: Belirli bir sonuç için yönlendirilmiş talimatlar verilir. Örnek: "Bir blog yazısı yaz" yerine, "300 kelimelik SEO uyumlu bir blog yazısı yaz. "

Bağlamsal bilgi sağlama: Modelin konu hakkında daha derinlemesine bir yanıt vermesi için ek bilgiler sağlanır.

Iterative prompting (Yinelenen prompt): İlk denemede istenen sonuca ulaşılmadığında, prompt'ı adım adım geliştirerek daha iyi sonuç elde etmek.



"İstem, yazılım mühendisliği değildir. Google'lamaya daha yakındır." **Saron Zhou**

Etkili Prompt Tasarımı ve Önemi

Prompt'un Yapısal Elemanları

İyi bir prompt, yapılandırılmış, hedef odaklı ve açık olmalıdır. Etkili bir prompt aşağıdaki 5 temel bileşeni içerir.

1. 🎯 Bağlam (Context)

Modelin görevi doğru anlaması için gereken ön bilgidir. Prompt'un giriş kısmında yer alır.

Önemi: Dil modelleri bir "ön bilgi" olmadan sıklıkla halüsinasyon yapabilir ya da yanlış varsayımlar üzerinden ilerleyebilir.

İyi bir örnek: Bir müşteri hizmetleri temsilcisisin. Aşağıdaki metni okuyarak müşterinin şikayetini özetle.

>> Metin: "Siparişim 3 gündür gelmedi ve kimseye ulaşamıyorum."

Kötü bir örnek: Şikayeti özetle.

2. 📝 Talimat (Instruction)

Modelin tam olarak ne yapması gerektiğini net biçimde belirtir. Görevin tanımıdır.

Önemi: Eksik ya da belirsiz talimat, modelin ne yapacağını anlamamasına ve rasgele cevaplar üretmesine neden olur.

İyi bir örnek: Lütfen aşağıdaki metni akademik bir üslupla 3 cümlelik bir özet olarak yaz.

Kötü bir örnek: Bunu düzenle.

3. 📑 Çıktı Biçimi (Output Format)

Modelden beklenen çıktının yapısal formatını belirtir. JSON, tablo, madde madde liste vb. olabilir.

Önemi: Format belirtilmediğinde modelin ürettiği içerik yapısız, tutarsız veya sistem tarafından işlenmez hale glebilir.

İyi bir örnek: Yanıtını JSON formatında ver:

```
{
  "özet": "",
  "duygu": "",
  "öncelik": ""
}
```

Kötü bir örnek: Cevapla.

Etkili Prompt Tasarımı ve Önemi

1. 👨‍🎓 Rol Atama (Role Instruction)

Modelin bir uzman, danışman, öğretmen vb. gibi davranmasını sağlamak için bağlamsal kimlik tanımı.

Önemi: Model, bir role atanmadığında cevabını hangi bilgi düzeyinde ve hangi tarzda vereceğine karar veremez.

İyi bir örnek: Sen bir psikologsun. Aşağıdaki davranış örüntülerini değerlendir ve önerilerde bulun.

Kötü bir örnek: Bunu analiz et.

2. 📏 Kısıtlar (Constraints)

Yanıt uzunluğu, stil, dil, teknik seviye gibi sınırların belirtilmesidir.

Önemi: LLM'ler bazen gereğinden fazla uzun, dağınık veya karmaşık yanıtlar verebilir. Kısıtlar bu dağınıklığı kontrol altına alır.

İyi bir örnek: Cevabın maksimum 3 cümle olsun. Teknik terimler kullanma. 12 yaşındaki bir öğrenci anlayabilmeli.

Kötü bir örnek: Kısa tut.

Örnek Prompt Şablonu:

```
[ROL] Sen bir tıp araştırma asistanısın.
[TALİMAT] Aşağıdaki makaleyi özetle:
[GİRDİ] "2024 yılında yapay zeka tanı sistemleri…"
[FORMAT] JSON:
{
  "başlık": "",
  "ana_kesit": "",
  "tarih": ""
}
[KISIT] En fazla 120 kelime; akademik dil
```

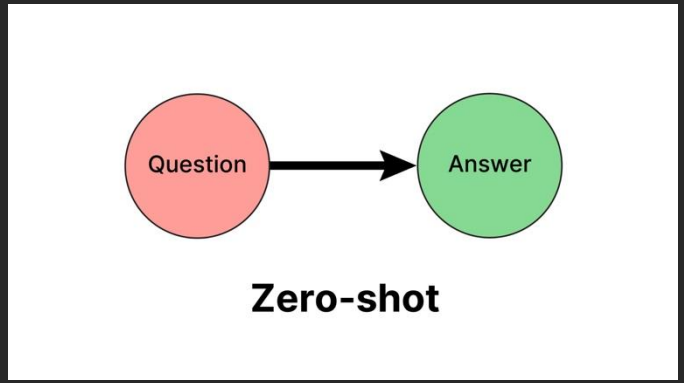
Temel Prompt Türleri ve Kullanım Alanları

Zero-shot Prompting

Modelden herhangi bir örnek vermeden doğrudan istenilen çıktıyı talep etme yöntemidir.

Ör: "Paris hakkında kısa bir bilgi ver. "

Kullanım Durumu: Hızlı bir yanıt almak istendiğinde veya modelin esnekliğinden yararlanmak istendiğinde kullanılır.

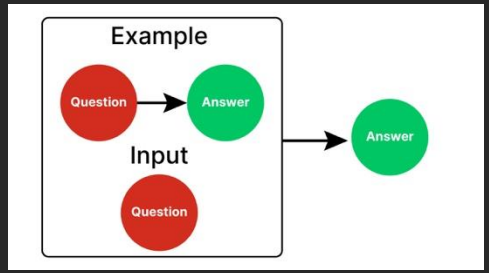


One-shot Prompting

Modelden istenen çıktının yanında bir örnek vermek suretiyle yönlendirmede bulunma tekniğidir.

Ör: "Aşağıdaki cümleyi özetleyin: 'Güneş doğarken dağların ardında parlıyor.' Örnek: 'Dağlar parlıyor.'

Kullanım Durumu: Örneğin yanıtın biçimini belirlemek istendiğinde veya spesifik bir format gerekiyorsa tercih edilir.



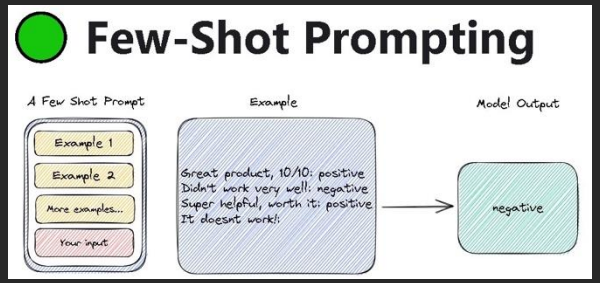
Temel Prompt Türleri ve Kullanım Alanları

Few-shot Prompting

Modelden istenen çıktının yanında birden fazla örnek vererek yönlendirme sağlama yöntemidir.

Ör: "Aşağıdaki cümleleri özetleyin: 'Güneş doğarken dağların ardında parlıyor.' -> 'Dağlar parlıyor.' 'Deniz akşamları maviye bürünüyor.' -> 'Deniz mavi.'»

Kullanım Durumu: Modelin daha spesifik bir bağlamda doğru yanıtlar vermesi gerektiğinde etkilidir.

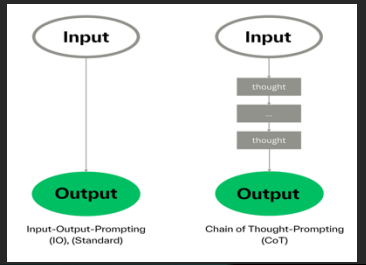


Chain-of-Thought Prompting

Modelden bir problemi veya soruyu adım adım düşünmesini istemek. Böylece daha karmaşık problemler çözülürken arka plandaki düşünme süreci model tarafından görünür hale gelir.

Ör: "Bir otobüs yolculuğunda geçen süreyi hesaplamak için önce kaç durak olduğunu ve her bir durakta geçen süreyi düşün.»

Kullanım Durumu: Mantık yürütme gerektiren sorular veya daha derin analiz gerektiren durumlar için uygundur.



Teknik Bağlam ve Yapısal İpuçları

LLM'lerin gücünü ortaya çıkarmak için genel yönlendirme prensiplerini ve ipuçları;

1. **Basitten başlayın:** Amacımız daha iyi sonuçlar elde etmek, bu nedenle bunun gerçekten yinelemeli bir süreç olduğunu, özellikle de LLM'leri kullanarak karmaşık görevleri başarmaya çalışırken aklınızda bulundurmanız önemlidir.
2. **Belirli talimatlar yazın:** Bunu birinin uzmanlığına başvurmak istiyormuşuz gibi düşünebiliriz. Ancak her şeyden önce ne tür bir uzmanlığa ihtiyacımız olduğunu anlamamız gerekir. Buna göre, uzmanlığa ihtiyaç duyduğumuz belirli alanı belirleyebilir ve o alandaki en iyisini seçebiliriz.
3. **Yapmak yerine yapılmaması gerekeni söyleyin:** İstemleri tasarlarken sıkça kullanılan bir diğer ipucu ise ne yapılması gerektiğini söylemek yerine ne yapılmaması gerektiğini söylemektir. Bu, modelin çok daha iyi sonuçlar üretmesini sağlar.

Teknik Bağlam ve Yapısal İpuçları

4. **Bir sonuca varmak için acele etmeden önce kendi çözümünüzü bulmanızı söyleyin:** Bazen LLM'leri kullanarak bir probleme yönelik çözümünüzün doğruluğunu kontrol etmek isteyebilirsiniz. Bu noktada modele, önce kendi çözümünü bularak çözümümüzün doğruluğunu kontrol etmesini söylemek iyi bir fikirdir, aksi takdirde model sizi takip edecektir.

Bunların dışında dikkat edilmesi gereken dört temel nokta vardır.

- **Token Optimizasyonu:** Az kelime ile çok bilgi aktaran yapılar tercih edilir.
- **Örnek Sayısı Seçimi:** Few-shot için ideal örnek sayısı genellikle 3-5 arasıdır. Çünkü çok fazla örnek, daha fazla token ve bağlam kaybı riski demektir.
- **Embedding vs Prompt:** Eğer büyük bağlamlar varsa embedding uzayında işlemleri gerçekleştirmek verimi artırır.

Teknik Bağlam ve Yapısal İpuçları

```
### Görev: Aşağıdaki haber metnini özetle.

### Girdi:
“Türkiye 2023 yılında enerji ihracatını artırdı…”

### Biçim:
Özet üç cümle olacak şekilde yazılmalı. JSON formatında döndür:

{
  "özet": "..."
}
```

Yandaki örneklerde görüldüğü gibi rol atama biçim belirtme, sınır belirtme ve giriş talimatı verme de promptların yapısal tasarım tekniklerindendir ve model performansını önemli ölçüde etkiler.

Sen bir jeofizik veri analisti olarak çalışıyorsun. Aşağıdaki SEG-Y dosya başlık bilgisini incele ve olası kalite sorunlarını belirt.

[SEG-Y HEADER BLOCK]

Yanıtı madde madde teknik rapor biçiminde yaz.

PROMPT KALİTESİNİ TEST ETME YÖNTEMLERİ

Temel Amaç

Prompt'un doğruluğu, tutarlılığı, güvenliği ve etkililiğini değerlendirmek

Yöntem	Hız	Objektiflik	Uygunluk	Kullanım Alanı
LLM-as-a-judge	Yüksek	Orta	Geniş	Otomatik testler
Human-in-the-loop	Orta	Yüksek	Geniş	Kullanıcı bazlı ürünler
Otomatik Metrikler	Yüksek	Düşük	Dar	Özetleme, çeviri
A/B Testi	Düşük	Yüksek	Orta	Kullanıcı tercih analizi
Test-case evaluation	Orta	Yüksek	Geniş	Fonksiyonel doğrulama
Adversarial testing	Orta	Yüksek	Geniş	Güvenlik, kırılganlık

Prompt Güvenliği ve Risk Yönetimi

Temel amacı LLM sistemlerinin kötü niyetli girdilere karşı korunması ve riskli çıktılar üretmesinin engellenmektir.

Prompt güvenliği, bir dil modelinin manipülasyona açık olmayan, güvenilir ve etik yönergeler çerçevesinde çalışmasını sağlar.

Modern LLM sistemlerinde kullanıcıdan gelen serbest metin girdileri, sistemi şu şekillerde tehlikeye atabilir:

- Sistem talimatlarını baypas etme (Prompt injection)
- Zararlı içerik üretme (toksik, yanlış, zararlı)
- Bilgi sızdırma (sistem prompt'unun ifşası)
- Role-playing saldırıları ("sen artık bir korsansın" gibi).

Prompt Güvenliği = Sistem Güvenliği demektir!

Bir LLM sisteminin çıktıları sadece doğru değil, güvenli de olmalıdır. Bu nedenle prompt güvenliği, proaktif olarak ele alınmalı ve sistemin her katmanında uygulanmalıdır.


Q&A

TEŞEKKÜRLER!

Kubilay Tuna

Senior Data Scientist

kubilaytuna26@hotmail.com