



TRAKYA ÜNİVERSİTELER BİRLİĞİ
4. ULUSLARARASI AKADEMİK FİLOLOJİ ÇALIŞMALARI KONGRESİ
BEŞERİ BİLİMLERDE DİJİTALLEŞME
11-12 Aralık 2025
Tekirdağ / Türkiye

Türkçe Dil Araştırmaları İçin Yapay Zekâ Araçlarında Kod Üretimi: Prompt Stratejileri, Örnekleri ve Python Uygulamaları

Doç. Dr. Ahmet Kayasandık

Abdullah Gül Üniversitesi, Kayseri, Türkiye
ahmet.kayasandik@agu.edu.tr

Özet

Bu bildiri, yapay zekâ (YZ) araçlarına doğru istemlerle Google Colab veya Python'da çalıştırılacak kodlar yazdırarak kodlamayı bilmeden de bu uygulamaların Türkçe dil araştırmalarında nasıl kullanılabileceğini göstermek amacıyla hazırlanmıştır. ChatGPT, Claude, Qwen, DeepSeek, Gemini, Copilot gibi büyük dil modellerine Google Colab veya Python ortamlarında çalıştırılabilir kodlar yazdırmak için promptların (istemlerin); açık, spesifik, bağlama uygun, formatı tanımlı, kütüphane tercihleri belirli, kısıtları ve şartları belli, örnekli, çıktısı tanımlı gibi nitelikleri taşıması gerekmektedir. Etkili prompt stratejileriyle yazdırılan ve aşağıdaki işleri yapan kodların hatasız çalıştığı test edilmiştir. Bu bildiride, dil araştırmalarında kullanılabilecek kodları YZ araçlarına yazdırmak için örnek promptlar (istemler) sıralanmıştır. Bu örnek istemlerle YZ araçlarına yazdırılan kodlar, kopyalanıp Google Colab'a veya Python'a yapıştırılarak kelime sıklığı analizi, kelime bulutu oluşturma, kavram ağı görselleştirme, kelime türü belirleme, kelime dağılımı, grafik oluşturma, metin ayıklama, söz varlığı tespiti, metin karşılaştırma, morfolojik analiz, okunabilirlik analizi gibi işler kolaylıkla yapılabilmektedir. Bunların nasıl yapılacağı da anlatıldı ancak bildiri sınırlarını fazlasıyla aşacağından analizi yapılacak örnek metne ve Python kodlarına yer verilmedi.

Ayrıca doğru istemler girilerek word makroları yazdırılabilir ve bu makrolarla birtakım metin düzenlemeleri word programıyla da yapılabilir. Kod gereksinimi olmadan da YZ araçlarına doğru istemler girilerek mesela tarihî metinlerin günümüz Türkçesine aktarılması, işlevsel analiz gibi işler için de daha doğru çıktılar alınabilir.

Bildiri, YZ araçlarının Türkçe dil araştırmalarında da zaman tasarrufu ve kolaylık sağlayan, doğruluğu yüksek, tekrarlanabilir, erişilebilir çözümler sunduğunu örnekleriyle ortaya koymaktadır. Dil araştırmacılarının/uzmanlarının teknik programlama bilgisi olmadan da karmaşık metin analizleri yapabilmesine imkân veren bu yaklaşım, dijital filolojinin geleceği için yeni bakış açıları ortaya koymaktadır.

Anahtar Sözcükler: Yapay zekâ destekli filoloji, istem stratejileri, istem örnekleri, Python kodları üretme ve çalıştırma

1. Giriş

"Teknolojik dönüşüm karşısında ne onu tümünden reddetmek ne de onu sorgulamadan teslim olmak doğru bir tutumdur. Önemli olan, bu dönüşümle bilinçli ve dengeli biçimde başa çıkabilmek; birey ve toplum olarak teknolojiyi kendimize hizmet edecek şekilde şekillendirebilmektir. Yapay zekâ



TRAKYA ÜNİVERSİTELER BİRLİĞİ
4. ULUSLARARASI AKADEMİK FİLOLOJİ ÇALIŞMALARI KONGRESİ
BEŞERİ BİLİMLERDE DİJİTALLEŞME
11-12 Aralık 2025
Tekirdağ / Türkiye

artık uzak bir geleceğin kurgusu değil; günümüz dünyasının yönünü belirleyen somut ve etkili bir güçtür. Bu güç kaçınılmaz olduğu kadar yönlendirilebilir niteliktedir" (Akdoğan, 2025, s.23-24).

Dil araştırmalarında yararlanılan yöntemler, tarihî süreçte çeşitli aşamalardan geçmiş, zamanla değişik yaklaşımlar ve teknikler ortaya çıkmıştır. Son zamanlarda yaşanan dijital ve teknolojik gelişmeler ve yapay zekânın (YZ) her geçen gün artan yetenekleri dil araştırmalarında da kolaylıkları mümkün kılmıştır. Dijital çözümler üreten çeşitli yazılımlarla, YZ araçlarıyla ve YZ araçlarına yazdırılabilecek Python kodlarıyla çeşitli dil araştırmaları/analizleri daha kısa sürede ve yüksek doğruluk oranıyla yapılabilmektedir.

Konunun uzmanları, üretken yapay zekâ araçlarının artık yalnızca sanatçıya ve yazara yardımcı olan araçlar olmaktan çıktığını; bilgi üretiminde, düşünce geliştirmede ve araştırma süreçlerinde aktif bir iş ortağı hâline geldiklerini belirtmektedir. Bu araçlar yalnızca metin yazmakla kalmamakta, düzeltme, genişletme, kısaltma ve ifade biçimlerini dönüştürme gibi işlevlerle yazma sürecini kolaylaştırmaktadır. Ayrıca, diller arası çeviri yetenekleri sayesinde metinleri farklı dillere çevirebilmekte; bu da hem kültürel erişimi hem de literatür takibini büyük ölçüde kolaylaştırmaktadır (Tosun, 2025, s. 28).

Yapay zekâ veri işleme ve problem çözme yetenekleriyle geleceği şekillendiren bir güç hâline gelmiş, sınırsız potansiyel barındırdığını göstermiştir. Bu potansiyeli ortaya çıkarmaksa etkili promptlar (istemler) oluşturmayı gerektirir. Prompt yazımının inceliklerini bilmek yapay zekâ modellerinin performanslarının üst düzeye çıkarılmasını sağlar. Doğru tasarlanmış promptlar yapay zekânın istenen çıktıları tutarlı ve verimli üretmesini kolaylaştırır. Yapay zekâ modeline verilen talimat veya girdi anlamına gelen prompt, modelin üreteceği çıktıyı doğrudan şekillendirir. Prompt yalnızca modelin ne yapması gerektiğini belirlemekle kalmaz, aynı zamanda ne kadar doğru ve anlamlı bir sonuç elde edileceğini de etkiler. Doğru tasarlanmış bir prompt, modelin istenen sonucu üretmesini kolaylaştırır ve çıktının tutarlılığını artırır (Erdör, 2025).



TRAKYA ÜNİVERSİTELER BİRLİĞİ
4. ULUSLARARASI AKADEMİK FİLOLOJİ ÇALIŞMALARI KONGRESİ
BEŞERİ BİLİMLERDE DİJİTALLEŞME
11-12 Aralık 2025
Tekirdağ / Türkiye

YZ'nın karmaşıklığı, belirsizliği ve çeşitliliği gerekçesiyle verimlilik ve maliyet tasarrufu sağlama amacıyla prompt mühendisliğine ihtiyaç duyulmuştur. Prompt mühendisleri, YZ modellerinden en üst düzeyde yararlanmaya, kullanıcıların YZ araçlarıyla doğru iletişim kurmasına, istemlerin dikkatli ve bilinçli tasarlamasına odaklanırlar.

2. Kuramsal Çerçeve

YZ araçları, dil araştırmalarında da yeni imkânlar ve kolaylıklar sunarak araştırma süreçlerini dönüştürmektedir. Ancak bu araçlardan üst düzeyde verim almak, etkili prompt (istem) yazma becerisine bağlıdır. Bu bildiri, dil araştırmalarında YZ araçlarını verimli kullanabilmek için istem yazarken dikkat edilmesi gereken temel ilkeleri, örnek istemleri ve çıktıların nasıl kullanılacağını göstermek amacıyla hazırlanmıştır. İstem ilkeleri, tercih edilecek YZ araçları, istem örnekleri ve ürettirilen kodların nasıl çalıştırılacağı bu bildirinin kuramsal çerçevesini oluşturmaktadır.

3. Yöntem

Bu çalışmada nitel yöntemi kullanılmıştır. YZ araçlarından dil araştırmalarında yararlanmak için istemlerin nasıl yazılması gerektiği belirtildikten sonra amaca uygun istem örnekleri sıralanmıştır. Bu istemlerle ürettirilen kodların sorunsuz çalıştığı teyit edilmiştir. Bu kodların Python'da nasıl çalıştırılacağı da anlatılmıştır.

4. Bulgular

4.1. Etkili istem yazmanın temel ilkeleri

"Üretken YZ aracına girilen bilgiler, cümleler veya sorular (istemler), alınan çıktıların kalitesi üzerinde büyük bir etkiye sahiptir. Bir istem girdikten sonra, YZ modeli girdiyi analiz eder ve eğitim yoluyla öğrendiği kalıplara dayanarak bir yanıt üretir. Daha açıklayıcı istemler, çıktıların kalitesini artırabilir" (HUIT, 2023). "İyi bir prompt, yapay zekâ sistemlerinin istenilen sonuçları etkili bir şekilde üretmesini sağlayan kritik bir unsurdur" (Sevil, 2025, s. 25). Prompt (istem, komut), YZ modüllerine verilen talimatlardır.



TRAKYA ÜNİVERSİTELER BİRLİĞİ
4. ULUSLARARASI AKADEMİK FİLOLOJİ ÇALIŞMALARI KONGRESİ
BEŞERÎ BİLİMLERDE DİJİTALLEŞME
11-12 Aralık 2025
Tekirdağ / Türkiye

YZ modeli girilen istemin analizini yapar ve öğrendiği kalıplara göre cevap üretir. Rolü tanımlı, açık, anlaşılır, sınırları belli, çıktı formatı belirlenmiş net istemler, çıktıların doğruluğunu ve kalitesini arttırabilir. İstemler her ne kadar usulüne uygun girilmiş olsa bile YZ araçlarının ürettiği içerik uydurma, yanlış veya yanıltıcı olabileceği için çıktılar incelenmeli ve teyit edilmelidir.

"YZ araçları sizin söylediklerinizi anlamaya çalışır ama düşünmez. Bu nedenle verdiğiniz bilginin netliği, detay seviyesi ve hedefi sonucu doğrudan etkiler. Doğru yazılmış bir prompt; zaman kazandırır, daha profesyonel ve tutarlı sonuçlar üretir, deneme-yanılma süresini azaltır, yaratıcı fikirleri destekler. Yanlış yazılmış bir prompt ise çoğu zaman yüzeysel, alakasız veya tekrar eden sonuçlar doğurur" (Dijibiz). İstemler İngilizce yazıldığında YZ uygulamaları daha az token harcamakta, daha hızlı ve tutarlı çıktılar üretmektedir. Bu sebeple Türkçe istemlerin İngilizceye çevrilmiş şekilleri de tercih edilebilir.

YZ araçlarından daha doğru, kapsamlı ve amaca uygun çıktılar alabilmek için istemler girilirken aşağıda sıralananlar gözetilmelidir:

1. Temel yapısal unsurlar

a) Rol (veya kişilik) atama: YZ modeline belli bir rol atandığında çıktının içeriği ve bakış açısı şekillendirilerek uzmanlık isteyen konularda daha tutarlı ve ayrıntılı çıktılar alınabilir. Örnekler: Sen bir Türkçe profesörün/uzmanısın. Sen Ömer Seyfettin tarzında öyküler yazan bir yazarsın.

b) Görevi tanımlama (Task): "Yapay zekâya görev atarken açık ve öz olmak da önemlidir" (Valliani, 2024). YZ modelinden yapılması istenenin net biçimde ifade edildiği emir kipinde çekimlenmiş bir fiille görev tanımlanmalıdır. Örnekler: Yükleğim belgenin özetini çıkar. Eklediğim belgedeki tarihî metni günümüz Türkçesine aktar.

c) Bağlam sağlama (Context): İstemde arka plan bilgisi, bağlamsal veriler, hedef kitle ve ayrıntılar bulunduğunda daha tutarlı çıktılar alınır. Örnek: Önce yüklediğim belgeleri oku. Sonra sadece bu



TRAKYA ÜNİVERSİTELER BİRLİĞİ
4. ULUSLARARASI AKADEMİK FİLOLOJİ ÇALIŞMALARI KONGRESİ
BEŞERÎ BİLİMLERDE DİJİTALLEŞME
11-12 Aralık 2025
Tekirdağ / Türkiye

belgelerdeki bilgilerden yararlanarak lisans düzeyindeki edebiyat bölümü öğrencileri için Eski Anadolu Türkçesinin ses özelliklerini özetle.

ç) Biçimi belirleme (Format): Çıktı biçiminin açıkça belirtilmesi, yetersiz bilgi veya gereksiz ayrıntıyı önler, vakit kaybını engeller. Çıktı hangi dosya formatında nasıl kaydedilecek veya nerede görünecek? Örnekler: Sonuçları indirilebilecek word (.docx) belgesine sütunlar hâlinde yazdır. Yükleğim belgedeki verilerin çıktısını pasta grafik olarak .png biçiminde yazdır.

2. İçerik ve detaylandırma

Temel yapıya ek olarak aşağıdaki hususları da dikkate almak çıktı doğruluğunu ve kalitesini yükseltecektir:

a) Açıklık ve spesifiklik: Genel ifadeler yerine amaç net olarak belirtilmeli ve spesifik yönergeler yazılmalıdır. "Bir öykü yaz" gibi genel bir istem yerine "10-15 yaş aralığındaki çocuklar için, büyüklere saygı konusunun işlendiği, 400-500 kelime arasında, eğlenceli bir öykü yaz" gibi istem tercih edilmelidir. Prompt örnekleri genel ağ sayfasında YZ araçlarının ne istediğimizi bizim kadar sezemediği, istek zihninizde çok net olsa bile, bu açıkça ifade edilmediğinde YZ modelinin sorunu doğru anlamlandıramayıp farklı bir yöne gidebileceği ve ilgisiz bir cevap verebileceği uyarısı yapılmaktadır.

b) Üslubu ayarlama: Çıktının hangi üslupta (resmî, akademik, içten vb.) istendiği YZ modülünde seçilmelidir.

c) Kısıtları ve şartları belirtme: Çıktının sınırlarını ve ölçütlerini (mesela en fazla 300 kelime kullan, büyük harflerle yazılanları dikkate alma vb. gibi) belirlemek gereksiz içerik üretimini önler. Bunun yanında kısıtların çok fazla olması, yapay çıktılara neden olabilir.

ç) "Yap" ve "yapma" kullanımı: İstemde "şunu/şunları yap", "şunu/şunları yapma" şeklinde istenenleri veya istenmeyenleri belirtme çıktıda gereksiz bilgileri önler.



TRAKYA ÜNİVERSİTELER BİRLİĞİ
4. ULUSLARARASI AKADEMİK FİLOLOJİ ÇALIŞMALARI KONGRESİ
BEŞERÎ BİLİMLERDE DİJİTALLEŞME
11-12 Aralık 2025
Tekirdağ / Türkiye

d) Örnek verme: YZ modülüne çıktının nasıl olması gerektiğine dair örnek(ler) verilmesi, daha tutarlı sonuçlar üretir. Örnekler: Eklediğim metindeki cümlelerin başındaki sıra numaralarını "25-Aç ne yemez, tok ne demez. Aç ne yemez, tok ne demez (25)." örneğindeki gibi cümlenin sonunda noktadan önce parantez içinde yaz. Paragrafları aşağıdaki paragrafın üslubuna benzer şekilde düzenle.

e) Doğal dili kullanma: İstemler, muhatap insanmış gibi doğal dille, günlük konuşmaları taklit eder nitelikte, tam cümleler şeklinde yazılmalı/girilmeli ve yazım kurallarına uyulmalıdır. Nazik bir dil, YZ araçlarının yapıcı ve olumlu çıktıları üretmesine yardımcı olabilir.

f) Özlü ifade: İstemlerde gereksiz ayrıntılardan kaçınılmalıdır. Komutlar öz biçimde, kolayca anlaşılabilir nitelikte olmalıdır.

"Bazen tek bir prompt ile istenen mükemmel çıktıyı almak zor olabilir. Yinelemeli iyileştirmeye ilk alınan cevap değerlendirilip ardışık sorularla iyileştirilebilir. Örneğin 'Bu metni daha resmî bir tonda tekrar yazar mısın?' veya 'Şu bölüm hakkında daha detaylı bilgi ekler misin?' gibi ek promptlarla sohbet oturumunu sürdürülerek cevap adım adım geliştirilebilir" (İsimsiz). Karmaşık görevler için istem yazmak gerektiğinde sürecin aşamalarını belirtmek, adım adım yönlendirmek tutarlı çıktılar elde etmeyi sağlar. İstemler için şablonlar da kullanılabileceği gibi istenenler sıralanarak YZ modülüne de promptlar yazdırılabilir. Akademik ve kurumsal çalışmalarda veri gizliliğine, telif haklarına, yerel politikalara dikkat edilmeli, onaylı araçlar kullanılmalı ve çıktılar teyit edilmelidir.

İstemlerde bağlam eksikliği, belirsiz genel ifadeler, format belirtmeme, gerçekçi olmayan beklentiler gibi hatalar yapılırsa çıktıların yanlış, eksik, tutarsız olma ihtimali çok yüksek olduğu gibi yazdırılan kodlar da çalışmayabilir.

Dil araştırmalarında çoğunlukla metin tabanlı istemler kullanılmaktadır. Yukarıda sıralanan genel ilkeler göz önünde bulundurulmak kaydıyla hedef dil ve lehçe (Türkiye Türkçesi, Türkmen



TRAKYA ÜNİVERSİTELER BİRLİĞİ
4. ULUSLARARASI AKADEMİK FİLOLOJİ ÇALIŞMALARI KONGRESİ
BEŞERİ BİLİMLERDE DİJİTALLEŞME
11-12 Aralık 2025
Tekirdağ / Türkiye

Türkçesi vb gibi), metnin türü (akademik, konuşma dili), dilbilimsel odak (semantik, morfolojik analiz vb.) ve çıktı uzunluğu da belirtilmelidir. Google Colab veya Python'da çalıştırılacak kodlar istenirken de çalışma ortamı (Google Colab, Python), girdi formatı (.txt, .docx, .pdf), kullanılacak kütüphaneler (python-docx, TurkishBERT, zemberek, stanza vb.) ve çıktı formatı da (.txt, .docx, .xcl, .csv, istatistiksel analiz, dosya çıktısı görselleştirme vb.) unutulmamalıdır.

4.2. Hangi YZ aracı tercih edilmeli?

Her geçen gün YZ araçlarına yenileri eklendiği gibi mevcut araçlara da yeni yetenekler kazandırılmaktadır. Bu sebeple burada yapılan değerlendirmelerin bugün itibariyle geçerli olduğu unutulmamalıdır. Bugün girilen isteme göre kod üretemeyen bir YZ aracı yarın bunu yapabileceği gibi, şimdi çalışan bir kod üreten YZ modeli daha sonra aynı çıktıyı veremeyebilir. ChatGPT, Claude, DeepSeek, Perplexity, Gemini, Qwen, Grok ve Copilot çok bilinen YZ araçlarıdır. Tarihi metinlerin günümüz Türkçesine aktarılmasında, metin üretmede, çeviride ve bağlama uygunlukta Claude'un diğerlerine göre daha başarılı olduğu tarafımızdan deneyimlenmiştir. Akbulut (2025) ise yaptığı araştırmayla OpenAI (ChatGPT, Codex), Google Gemini, Copilot (GitHub) ve Microsoft Copilot uygulamalarından ChatGPT'nin karmaşık görevlerde diğer uygulamalara göre daha iyi performans gösterdiğini, GitHub Copilot'ın günlük yazılım geliştirme süreçlerinde ise hızlı ve pratik çözümler sunduğunu, eğitim ve temel projeler için basit ve doğrudan çözümler sunan Google Gemini'in uygun olduğunu, büyük ölçekli kurumsal projelerde entegrasyon ve güvenlik uyumlu çözümler sunmasıyla Microsoft Copilot tercih edilebileceği sonucuna ulaşmıştır.

4.3. Dil araştırmaları/çalışmaları için istem (prompt) örnekleri

YZ modüllerine word makroları yazdırma, doğrudan YZ araçlarına metin yazdırma, Colab'da veya Python'da çalıştırılmak üzere kodlar yazdırma için girilebilecek istemler aşağıda örneklenmiştir. Bu örnek istemlerden elde edilen çıktılar adı geçen uygulamalarda denenerek doğrulukları teyit edilmiştir. Kodları üretmede kullanılan YZ araçları parantez içinde belirtilmiştir. Yazdırılan bir



TRAKYA ÜNİVERSİTELER BİRLİĞİ
4. ULUSLARARASI AKADEMİK FİLOLOJİ ÇALIŞMALARI KONGRESİ
BEŞERİ BİLİMLERDE DİJİTALLEŞME
11-12 Aralık 2025
Tekirdağ / Türkiye

kod hata verdiğiğinde uygulamadaki kod hücrelerinin altında bu hatanın ne olduğu görülür. Bu kısım kopyalanıp kod hangi YZ modülüne yazdırılmışsa onun sohbet kutusuna "Şu hatayı verdi. Bunu düzeltip kodu güncelle:" yazılarak yapıştırılır. Bu işlem doğru çalışan kod ürettirilene kadar tekrar edilebilir. Aşağıdaki örneklerde değişkenler köşeli parantez içinde, kullanılan YZ modülü ise parantez içinde gösterilmiştir.

İdeal bir prompt (istem) örneği (Gemini): Sen Türkçe dil uzmanı bir akademisyensin. Google Colab'da çalışacak şekilde bir Python kodu yaz. Kullanıcıdan bir .docx dosyası seçmesini iste. Seçilen belgedeki tüm kelimeleri al, noktalama işaretlerini temizle, küçük harfe çevir. Tekrar eden kelimelerden sadece birini göster, kaç kez geçtiğini yanına yaz. Sonuçları frekans sıklığı sırasına göre sütunlar hâlinde yeni bir word belgesine yazdır. Bu word belgesi için indir seçeneği ekle. Gerekli kütüphaneleri otomatik kur (python-docx). Dosya seçilmezse işlemi iptal et ve uygun mesaj ver. Başarılı olduğunda dosya yolunu ekrana yazdır. Kodu tek hücrede çalıştırılabilir şekilde yaz.

1. Örnek: Word makrosu yazdırma (Gemini, Claude): Word için VBA makrosu yaz. Makro, aktif belgedeki metin üzerinde şu adımları eksiksiz uygulasin: Satır sonlarını ve tablaları boşluğa çevirsin. Tüm harfleri Türkçe kurallara göre küçük harfe dönüştürsün (İ → i, I → ı; Ç, Ğ, Ö, Ş, Ü doğru küçülsün). RegExp kullanmadan, karakter karakter kontrol ile yalnızca Türkçe harfleri (a-z ve ç, ğ, ı, i, ö, ş, ü) ve boşlukları korusun; diğer tüm karakterleri boşluğa çevirsin. Çoklu boşlukları tek boşluğa indirsin, baştaki/sondaki boşlukları kırpın. Metni kelimelere ayır ve Türkçe alfabeye göre (a, b, c, ç, d, e, f, g, ğ, h, ı, i, j, k, l, m, n, o, ö, p, r, s, ş, t, u, ü, v, y, z) özel karşılaştırma fonksiyonu ile sıralasın (vbTextCompare kullanmasın). Sonucu belgeye tek sütun hâlinde (her kelime yeni satır) yazsın. Boş satırları silsin. Kod tek modülde olsun ve hiçbir dış kütüphane/RegExp nesnesi kullanmasın.



TRAKYA ÜNİVERSİTELER BİRLİĞİ
4. ULUSLARARASI AKADEMİK FİLOLOJİ ÇALIŞMALARI KONGRESİ
BEŞERÎ BİLİMLERDE DİJİTALLEŞME
11-12 Aralık 2025
Tekirdağ / Türkiye

Bu istem "Word için VBA makrosu yaz. Makro, aktif belgedeki metin üzerinde şu adımları eksiksiz uygulasın: Satır sonlarını ve tablaları boşluğa çevirsin. Tüm harfleri Türkçe kurallara göre küçük harfe dönüştürsün. Metni kelimelere ayırsın ve Türkçe alfabeye göre özel karşılaştırma fonksiyonu ile sıralasın. Sonucu belgeye tek sütun hâlinde (her kelime yeni satır) yazsın. Kod tek modülde olsun ve hiçbir dış kütüphane/RegExp nesnesi kullanmasın. " şeklinde kısaltıldığında da Gemini, Perplexity ve ChatGPT doğru çalışan bir makro üretmektedir.

(Not: Uzun metinlerde word bunu yapamayabilir. Uzun metinler için Google Colab veya Python'da çalıştırılacak kodlar yazdırılmalıdır.)

2. Örnek: Paragrafı yeniden düzenleme (Claude, ChatGPT, Perplexity): Sen Türkçe dil uzmanısın. Aşağıdaki paragrafı daha ilgi çekici ve profesyonel hâle getirmek için akademik ton kullanarak yeniden düzenler misin? Okunabilirliği artırmak için çeşitli cümle yapıları ve daha açıklayıcı bir dil kullan. [Paragrafı ekleyin/yapıştırın.]

3. Örnek: Tarihî metinleri günümüz Türkçesine aktarma (Claude): Eklediğim belgede Eski Anadolu Türkçesi dönemine ait bir metin var. Bu metni, bağlama dikkat ederek günümüz Türkçesine aktarır mısın? (Metnin sözlüğü varsa önce bu yüklenir. YZ aracından önce bunu okuması sonra bu sözlükteki anlamlara bağlı kalarak aktarması istenir.)

4. Örnek: Çeviri yaptırma (Claude): Eklediğim Türkçe belgeyi, bir çeviri bilim uzmanı olarak akademik bir tonda [İngilizceye] çevirir misin?

5. Örnek: Bildiri metni yazdırma (Gemini, Claude): Sen dil uzmanı bir akademisyensin. Senden akademik bir bildiri metni yazmanı istiyorum. Eklediğim belgede [içinde de- fiilinin geçtiği Türk atasözleri] var. Önce bu belgeyi oku. Sonra bu atasözlerinin morfolojik ve işlevsel analizini yap. Hangi atasözleri hangi alt başlıklar için örnek olursa hepsini kullan. [250-300] kelime aralığında bildiri özetini ve [4-5] anahtar kelimeyi bildirinin başına yaz. Ayrıca bu bildiri için başlık önerilerinde bulun. Bildiri resmî bir tonda olsun.



TRAKYA ÜNİVERSİTELER BİRLİĞİ
4. ULUSLARARASI AKADEMİK FİLOLOJİ ÇALIŞMALARI KONGRESİ
BEŞERİ BİLİMLERDE DİJİTALLEŞME
11-12 Aralık 2025
Tekirdağ / Türkiye

6. Örnek: Makale yazdırma (Gemini, Claude, Perplexity): Sen dil uzmanı bir akademisyensin. [Yapay zekâ araçlarının dil bilgisi eğitiminde kullanılması] konusunda bilimsel bir makale yazmanı istiyorum. Bunun için izleyeceğin yol ve kısıtlar şunlardır: Önce, eklediğim belgeleri oku ve kaynak olarak sadece bunları kullan. Başta [150-200] kelime aralığında Türkçe özet ve [4-5] anahtar kelime olsun. Özeti ve anahtar kelimeleri [İngilizceye] çevir. Bu çeviriyi Türkçe anahtar kelimelerle giriş bölümünün arasına yerleştir. Makale metni [2500-3000] kelime aralığında olsun. Makalenin giriş, amaç, yöntem, sonuç ve kaynakça bölümlerini bu alt başlıkları kullanarak yaz ve alıntı (veya atıf) yapılan kaynakları "kaynakça" başlığı altına yaz. Bilgilendirici bir ton kullan.

7. Örnek: Kaynak araştırma (Perplexity, Qwen, ChatGPT): Sen dil uzmanı bir akademisyensin. [Osmanlı Türkçesinden Türkiye Türkçesine geçiş sürecinde ses değişimlerini inceleyen 200] kelimelik bir literatür taraması yazar mısın? Harf inkılabının etkisini ele al, en az 8 akademik kaynak öner (gerçek kaynaklar olsun). Ses denklikleri için örnekler ver (örnek: $\text{ğ} > \text{k}$, $\text{ğ} > \text{g}$). APA 7 atıf stilini kullan.

8. Örnek: Sosyodilbilim araştırması (Gemini): Sen dilbilimcisin. [Türkiye'de yaş gruplarına göre teknoloji terimlerinin kullanımını] araştıran bir anket soruları seti hazırlamanı istiyorum. Bu sette [3] farklı yaş grubu (7-10, 11-15, 16-18) için uygun sorular yaz. Sorulara açık uçlu ve kapalı uçlu sorular ekle. Kod değiştirme (code-switching) davranışlarını ölçebilecek soruları dâhil et ve etik kurul onayı için gerekli bilgilendirme metnini ekle.

9. Örnek: Psikodilbilim deneyi tasarımı (Gemini): Sen dilbilim uzmanı bir akademisyensin. İki dilli [Türkçe-İngilizce] konuşucularda kod değiştirme (code-switching) davranışlarını inceleyen bir deney protokolü tasarlamayı istiyorum. Bunu yaparken şunları kullan: [Katılımcı seçim kriterleri ($n=30$), uyaran malzemeleri (20 karma cümle örneği), veri toplama yöntemi (kayıt, anket, gözlem), analiz parametreleri (frekans, bağlam, gramatikal kategori), beklenen bulgular ve olası yorumlar, madde işaretleri ve alt başlıklar kullan.]



TRAKYA ÜNİVERSİTELER BİRLİĞİ
4. ULUSLARARASI AKADEMİK FİLOLOJİ ÇALIŞMALARI KONGRESİ
BEŞERİ BİLİMLERDE DİJİTALLEŞME
11-12 Aralık 2025
Tekirdağ / Türkiye

10. Örnek: Morfolojik analiz-1 (Perplexity, Claude, Gemini): Sen Türkçe dil bilgisi uzmanısın. Görevin [çocukluğumuzdan] kelimesinin morfolojik analizini yapmaktır. Kök kelimeyi belirle. Her eki tanımla (çoğul eki, iyelik eki, ayrılma hâli eki). Ek sıralamasının Türkçe dil bilgisi kurallarına uygunluğunu açıkla. Benzer yapıda [3] örnek daha ver. Bulgularını tablo formatında sun. Açıklamaları Türkçe yaz.

Morfolojik analiz-2 (ChatGPT, Claude): Sen Türkçe dil bilgisi uzmanısın. Görevin word belgesindeki bütün kelimelerin morfolojik analizini yapacak Python kodunu yazmaktır. Kurallar: Kod [MacOS] işletim sisteminde kurulu VSCode'da çalıştırılacak. Gerekli kütüphaneler otomatik kurulsun. Kütüphane bulunamazsa kullanıcıdan (zemberek-full.jar veya turkishBERT) manuel olarak yüklemesi istensin. Kullanıcıdan analizi yapılacak dosyayı yüklemesi istensin. Sonuçlar indirilebilir word belgesine yazdırılsın. Ayrıca bu kodun nasıl çalıştırılacağını da yazar mısın?

11. Örnek: Metinden, tanımlı cümle bulma (Gemini, Claude): Word belgesindeki metinden içinde [büyük] kelimesi geçen cümleleri ayrı bir word belgesine yazdırmak istiyorum. Bunun için Google Colab'da çalıştırılacak Python kodunu yazar mısın? Kurallar: Kod tek hücrede olsun. Gerekli kütüphaneler otomatik olarak kurulsun. Kullanıcıdan "dosya seç" düğmesiyle dosyayı yüklemesi istensin. (Bu işlem YZ araçlarına yaptırılabilir ama sonuçlar eksik çıkıyor.)

12. Örnek: Metin ayıklama (ChatGPT): Senden bir word belgesinden sıra numaraları tanımlı cümleleri seçip yeni bir word belgesine yazdıracak bir Python kodu yazmanı istiyorum. Kurallar: [MacOS işletim sistemi kurulu Macbook Air] bilgisayar kullanıyorum. Üreteceğin kod bu bilgisayarda kurulu VS Code'da yerel olarak çalıştırmaya uygun olmalıdır. Kod çalıştırıldığında kullanıcıdan 2 dosya seçerek yüklemesini istemelidir. 1. dosyada [162- Ağaç kökünden yıkılır.] örneğindeki gibi sıra numaralı atasözleri var. 2. dosyada ise 1. dosyadan ayıklanacak [atasözlerinin sıra numaraları] var. Kullanıcıdan önce 1. dosyayı sonra 2. dosyayı yüklemesi istenmeli. 2.



TRAKYA ÜNİVERSİTELER BİRLİĞİ
4. ULUSLARARASI AKADEMİK FİLOLOJİ ÇALIŞMALARI KONGRESİ
BEŞERİ BİLİMLERDE DİJİTALLEŞME
11-12 Aralık 2025
Tekirdağ / Türkiye

dosyadaki sıra numaralarını taşıyan cümleler, 1. dosyadan ayıklanan (seçilen) cümleler yeni word belgesine alt alta yazdırılmalı. Oluşturulan yeni word belgesi [indirilenler] klasörüne indirilmeli.

13. Örnek: Söz varlığı karşılaştırma (Gemini, ChatGPT, Claude): Görev: .docx formatındaki 2 ayrı dosyadaki metinleri karşılaştırarak ortak kelimeleri bulmak için Python kodunu yazmak. Kurallar: Kod [MacOS] işletim sisteminde kurulu VSCode'da çalıştırılacak. Kod tek hücrede tek başına çalışabilir olsun. Gerekli kütüphaneler otomatik olarak kurulsun. Kütüphaneler bulunamazsa kullanıcıdan (zemberek-full.jar veya turkishBERT) manuel olarak yüklemesi istensin. Kullanıcıdan 1. ve 2. dosyaları yüklemesi "dosya seç" düğmesiyle istensin. Bu iki dosyadaki ortak kelimelerin kaç kez geçtiği yanlarına yazılsın ve frekans sıklığına göre iki sütun hâlinde sıralansın. (Alfabetik sıralama istenirse "sıralama Türk alfabesine (a, b, c, ç, d, e, f, g, ğ, h, ı, i, j, k, l, m, n, o, ö, p, r, s, ş, t, u, ü, v, y, z) göre olsun" şeklinde değiştirilebilir.) Sonuçlar word belgesine yazdırılsın ve bu dosyayı kaydetme (indir) penceresiyle indirme imkânı versin.

14. Örnek: Korpus analizi (Gemini, ChatGPT): Görev: Google Colab'da tek hücrede çalışacak bir Python kodu yaz. Kurallar: Kod tek hücre içinde olacak. Kullanıcıdan analiz edilecek .docx dosyasını yüklemesini iste. Gerekli kütüphaneler otomatik algılsın, eksik olanlar pip install ile kurulsun. Metni paragraf bazlı oku, kelimeleri küçük harfe çevir, tokenize et. Stopwords listesi benim vereceğim tam listedir ve kod tarafından doğrudan kullanılmalıdır: [acaba, ah, ama, ancak, aslında, ayrıca, bazı, belki, biçimde, bile, bir, birer, birisi, birkaç, bu, bütün, çoğu, çünkü, da, daha, de, defa, değil, değilse, diğer, diye, fakat, gibi, hem, hey, hiç, ile, ise, için, kadar, kadar, kez, ki, kim, kime, kimi, kimse, kimden, lakin, madem, madem ki, mi, mu, ne, of, oysa, öbür, öyleyse, pek, sanki, şey, şu, ve, veya, ya, yalnız, yalnızca, yani, yine] Stopwords çıkarıldıktan sonra kelime frekanslarını hesapla ve en çok tekrar eden [10] kelimeyi göster. Kelime frekanslarından görsel bir word cloud oluştur ve göster. Co-occurrence (birlikte görülme) mantığıyla kelime ilişkisi ağı çıkar ve network graph çiz. Basit bir zorluk analizi yap: Ortalama kelime uzunluğu, cümle başına kelime sayısı, Type-Token Ratio, Heuristik olarak metni kolay-orta-zor şeklinde sınıflandır. Kodun



TRAKYA ÜNİVERSİTELER BİRLİĞİ
4. ULUSLARARASI AKADEMİK FİLOLOJİ ÇALIŞMALARI KONGRESİ
BEŞERİ BİLİMLERDE DİJİTALLEŞME
11-12 Aralık 2025
Tekirdağ / Türkiye

tamamen sorunsuz çalışması için nltk punkt_tab eksikliği mutlaka giderilmeli →
nltk.download('punkt_tab', quiet=True) ekle. Sonuç olarak: bana tek hücre çalışır bir kod ver.

15. Örnek: Analiz grafiği oluşturma (ChatGPT): Sen bir korpus dilbilimcisisin. Görev: MacOS işletim sistemi kurulu bilgisayarda VSCode'da çalıştırılacak tek hücrelik bir Python kodu yaz. Kurallar: Kod tek parça (tek hücre) olsun. Gerekli tüm kütüphaneler kod içinden otomatik yüklensin. Kullanıcıdan .docx dosyası seçmesini iste. Dosyadaki kelimeleri temizle, şu stopwords kelimelerini çıkar: "acaba, ah, ama, ancak, aslında, ayrıca, bazı, belki, biçimde, bile, bir, birer, birisi, birkaç, bu, bütün, çoğu, çünkü, da, daha, de, defa, değil, değilse, diğer, diye, fakat, gibi, hem, hey, hiç, ile, ise, için, kadar, kez, ki, kim, kime, kimi, kimse, kimden, lakin, madem, madem ki, mi, mu, ne, of, oysa, öbür, öyleyse, pek, sanki, şey, şu, ve, veya, ya, yalnız, yalnızca, yani, yine" Analiz: Kelimelerin POS (kelime türü) analizini yap. POS türlerinin yüzdelik oran dağılımını pasta grafik (.png) olarak üret. POS frekanslarını sütun grafik (.png) olarak üret. Kelime uzunluklarının dağılımını sütun grafik (.png) olarak üret. Çıktı: Tüm grafikler masaüstüne cikti_gorselleri/ klasörüne kaydedilsin. Kod grafikleri aynı anda ekranda gösterip görüntülenebilir olsun. Sonunda çıktı dosyalarının yollarını yazdır. Format örneği: pos_pasta.png, pos_sutun.png, uzunluk_sutun.png. Beklenen çıktı: Yukarıdaki tüm işleri yapan eksiksiz Python kodu.

16. Örnek: Karmaşık konuyu basitleştirme (Perplexity, Claude, Gemini, ChatGPT, Qwen3-Max): Rol: Sen alanında uzman, aynı zamanda öğrenci seviyesine inebilen bir açıklayıcısın. Yüklenen belge için kurallar: Ton: Basit, yalın ve örneklerle zenginleştirilmiş bir dil kullan. Gerekirse benzetmelerle kavramları somutlaştır. Sınırlamalar: Özgün metindeki temel kavramlar korunarak [lisans 1.] sınıf düzeyine uygun kısa cümlelerle yeniden ifade et. Teknik terimlerin basit karşılıklarını ver. Amaç: Yüklenen metnin ana hatlarını bozmadan, onu daha anlaşılır bir biçimde özetlemek. Çıktılar: Metnin sadeleştirilmiş versiyonu; önemli noktaları vurgulayan, paragraf yapısını koruyan ancak gereksiz detayları atılmış bir açıklama. İş akışı: Önce kısaca konunun ne olduğunu belirt, sonra alt başlıklara göre basitleştirilmiş açıklamaları paragraflar hâlinde yaz.



TRAKYA ÜNİVERSİTELER BİRLİĞİ
4. ULUSLARARASI AKADEMİK FİLOLOJİ ÇALIŞMALARI KONGRESİ
BEŞERİ BİLİMLERDE DİJİTALLEŞME
11-12 Aralık 2025
Tekirdağ / Türkiye

Uzun ve karmaşık görevler için aynı sohbet akışı içinde istemler adım adım yazılarak daha tutarlı çıktılar almak mümkündür.

4.4. Kodlar uygulamalarda nasıl çalıştırılır?

1. Word makroları: YZ araçlarına yazdırılan word makrolarını çalıştırmak için önce metnin bulunduğu word belgesi açılır. Uygulama menüsünden Araçlar>Makro>Visual Basic Düzenleyici yolu izlenerek veya Alt+F11 kısa yol tuş kombinasyonuna basılarak Microsoft Visual Basic açılır. Insert (Ekle) menüsünden Module tıklanır. YZ aracında ürettirilen kod kopyalanarak açılan Module kutusuna yapıştırılır. Run (Çalıştır) ► düğmesine basılır. İstenen işlem gerçekleşir. (YZ araçları ürettikleri kodun nasıl çalıştırılacağını da adım adım yazar.)

2. Google Colab: Google Colab'da kodları çalıştırmanın iki yöntemi vardır: İlki YZ araçlarından Google Colab'da çalıştırılmak üzere istenen kod kopyalanır: Google hesabıyla <https://colab.research.google.com/> genel ağ sayfası açılır. "Not defteri aç" popup sayfası otomatik olarak açılır. Bunun sol alt köşesindeki "+Yeni not defteri" düğmesi tıklanır. Sonra sayfanın sağ üst köşesindeki "Bağlan" açılır menüsünden "Çalışma zamanı türünü değiştir" seçilir. Buradan Python 3 ve CPU işaretlenir. Sayfanın solundaki "Kodlamaya başlayın veya yapay zeka ile kod oluşturun." yazan kutucuğa tıklanır. Kopyalanan kod buraya yapıştırılır ve ► (hücreyi çalıştır) düğmesine basılır. Program çalışır. Kullanıcıdan "dosya seç" düğmesine tıklayıp dosya yüklemesi istenir. Çıktılar görüntülenir/kaydedilir. Uygulamanın "Dosya" menüsündeki "Kaydet" seçilerek bu not defteri Google Drive'a kaydedilir. Aynı işlem için farklı metinlerde bu not defterine tıklamak yeterli olur.

İkinci yöntemde de aynı yol izlenir. Ancak YZ araçlarına ürettirilen kod kullanılmaz. Açılan sayfanın altındaki Gemini sohbet kutucuğuna istem yapıştırılır. İşlem adımları "Kabul et ve çalıştır" düğmesine tıklanarak adım adım gerçekleştirilir.



TRAKYA ÜNİVERSİTELER BİRLİĞİ
4. ULUSLARARASI AKADEMİK FİLOLOJİ ÇALIŞMALARI KONGRESİ
BEŞERİ BİLİMLERDE DİJİTALLEŞME
11-12 Aralık 2025
Tekirdağ / Türkiye

3. Visual Studio Code: (Kullanılan işletim sistemine göre <https://www.python.org/downloads/> genel ağ sayfasından yazılım indirilir ve kurulur. YouTube 'da bu işlemin nasıl yapılacağını görsellerle anlatan çok sayıda kısa video var. https://www.youtube.com/results?search_query=python+nasıl+kurulur) Program çalıştırılır. Sol üst köşedeki kod satırına YZ araçlarında yazdırılan kod kopyalanarak yapıştırılır. Sağ köşedeki "Çalıştır" ► düğmesine tıklanır. İstenen dosya(lar) yüklenir. Çıktılar ekranda görüntülenir/kaydedilir. (Kurulamayan gerekli kütüphaneler olursa program kullanıcıdan bunları yüklemesini ister. Benzer şekilde YZ araçları Python'da çalıştırılmak üzere ürettikleri kodların nasıl çalıştırılacağını da adım adım yazar.)

Yukarıda sıralan işlemlerden bazıları için kod gerektirmeyen <http://tools.nlp.itu.edu.tr/index.jsp> ağ sayfası da kullanılabilir. Kopyalanan metin, metin kutusuna yapıştırılır veya alttaki "dosya seç" düğmesine tıklanarak metin belgesi yüklenir. Bu ağ sayfasında "cümle ayırıcı, kelime ayırıcı, metni normale dönüştürme, yazım ve noktalama düzeltme, kelimelerin eksik ünlülerini tamamlama, yanlış yazımları düzeltme, kelime yazımlarını kontrol edip doğru veya yanlış olduğunu saptama, morfolojik analiz, morfolojik anlam ayrıştırma, tanımlanmış adları tanıma (NER), bağımlılık ayrıştırma, bütün işleri topluca yapma gibi işler yapılabilir. Hangi işlem isteniyorsa metin kutusunun solundaki menüye tıklanır.

Derlemi oluşturulmak isten metinler için <https://voyant-tools.org/> ağ sayfasındaki kutucuğa metin yapıştırılır veya dosya yüklenir. Tek tıklamayla yüklenen metnin derlemi görüntülenebilir.

5. Sonuç

Dijital ve teknolojik gelişmeler ve YZ araçlarının her geçen gün artan yetenekleri dil araştırmalarında da kullanılabilecek bazı kolaylıklar sağlamıştır. YZ araştırmacının yerine geçmez ancak araştırma süreçlerini hızlandıran, fikir üreten ve tekrarlayan işleri otomatikleştiren güçlü bir yardımcıdır. Doğru istemlerle YZ, dil araştırmalarında verimli bir iş ortağı olabilir. YZ



TRAKYA ÜNİVERSİTELER BİRLİĞİ
4. ULUSLARARASI AKADEMİK FİLOLOJİ ÇALIŞMALARI KONGRESİ
BEŞERİ BİLİMLERDE DİJİTALLEŞME
11-12 Aralık 2025
Tekirdağ / Türkiye

modüllerinin bağlama uygun ve tutarlı çıktılar üretebilmesi doğru tasarlanmış istemlere bağlıdır. Zira istemler, hem modelin ne yapması gerektiğini belirler hem de anlamlı ve doğru çıktılara ulaşmayı sağlar. YZ araçları, dil araştırmacılarına bazı kolaylıklar ve imkânlar sunmaktadır ancak bunlardan yararlanabilmek doğru ve etkili istemler yazabilmeyi gerektirir. YZ araçlarından verimli çıktılar alma, nasıl sorulduğuna bağlıdır. İyi bir istem, YZ'yi yönlendirir, kötü bir istem ise onu boşa çalıştırır.

Prompt mühendisliğinin önerilerinden yararlanmak istemleri geliştirmeye yardımcı olsa da YZ tüm sorunları çözen sihirli bir araç değildir. Doğal dili anlamada konusunda ilerlemeye devam etmesi sayesinde kod yazmadan, istekler iletebilmektedir. Ancak bunların açık ve ayrıntılı biçimde yazılması kullanıcının sorumluluğundadır. YZ araçlarından beklenen çıktılar elde edilemezse istemler gözden geçirilmelidir. Daha açık ve kuralları belirlenmiş istemlerle modeli adım adım yönlendirerek istenen cevaplara ulaşılabilir.

İstemler yazılırken temel yapısal unsurları göz önünde bulundurulmalı, içeriğe ve detaylandırmaya da dikkat edilmelidir. İyi bir istem yazılırken rol (veya kişilik) atanmalı, görevi iyi tanımlanmamalı, bağlam sağlanmalı ve format belirtilmelidir. Ne istendiğinin açıkça belirtilmesi, tonun ayarlanması, kısıtların ve şartların belirtilmesi, yap ve yapma komutlarının kullanılması, örnek verilmesi ve doğal dille özlü ifadelerin tercih edilmesi çıktılarının doğruluğunu ve verimliliğini arttıracaktır.

İstem yazmayı geliştirme, pratik yapma, değişik YZ modüllerinin güçlü/zayıf yönlerini tanıma, üretilen içeriği akademik ölçütlere göre doğrulama, etik kurallara uyma, istem kütüphanesi oluşturup işe yarayan kullanışlı istemleri kaydetme bu konuda sıralanabilecek önerilerdendir.

Kaynakça

Akbulut, M. (2025). Yapay zeka uygulamalarının Python kod üretimindeki performans karşılaştırması. *Sinop Ü Fen Bilimleri Dergisi*, 10(1), 259-288.
<https://doi.org/10.33484/sinopfbd.1635155>



TRAKYA ÜNİVERSİTELER BİRLİĞİ
4. ULUSLARARASI AKADEMİK FİLOLOJİ ÇALIŞMALARI KONGRESİ
BEŞERÎ BİLİMLERDE DİJİTALLEŞME
11-12 Aralık 2025
Tekirdağ / Türkiye

- Akdoğan, E. (2025). Yapay zekânın dönüştürücü etkisi: İnsan, toplum ve etik. *.tr*, Ankara: Yunus Emre Enstitüsü, s.4-24.
- Dijibiz (2025). Dijibiz.com. https://www.dijibiz.com/tr/blog/dogru-prompt-yazma-rehberi-yapay-zekadan-en-iyi-sonucu-alin?srsId=AfmBOor69SNxmyJRCiBm-WsijdznKOOu1SdvT3u7_VnU6CQV6rIqt4zb
- Erdör, M. (2025). Yapay zekâ araçlarında etkili prompt (istem) yazımı rehberi. <https://muraterdor.com/yapay-zeka-araclarinda-etkili-prompt-istem-yazimi-rehberi/>
- HUIT (Harvard University information technology) (2023). Getting started with prompts for text-based generative AI tools, <https://www.huit.harvard.edu/news/ai-prompts>
- İsimsiz (2025). Sık Sorulan Sorular (FAQ) - Prompt Örnekleri. <https://tinyurl.com/25l6h3no>
- Kayasandık, A. (2025). Dil araştırmalarında yapay zekâ ve Python: Geleneksel yöntemlerden dijital çözümlere. 17. Uluslararası Dünya Dili Türkçe Sempozyumu, 23-25 Kasım 2005, Bartın.
- Sevil, Ş. (2025). Yapay zekâ ve eğitim: Öğretmenler için uygulamalı prompt mühendisliği ve üretken araçlarla yenilikçi öğrenme stratejileri, Ankara: MEB.
- Tosun, N. (2025). Sanat, edebiyat ve yapay zekâ. *.tr*, Ankara: Yunus Emre Enstitüsü, s.25-48.
- Valliani, J. (2024). The ultimate guide to writing effective AI prompts. <https://www.atlassian.com/blog/artificial-intelligence/ultimate-guide-writing-ai-prompts>