

NLP- Sectoral ReAct Agent Design Task

Student: Erdem Kandilci

Student_id: 2021556041

In this assignment, we were tasked with transforming an LLM model into a ReAct Agent capable of reasoning, decision-making, and problem-solving using external resources. A base model notebook was provided as an example, and we were asked to select a domain and develop an AI assistant specialized in that field. For this project, I developed a ReAct Agent System for Turkish Customs and Foreign Trade Consultancy. This domain was well-suited to the assignment requirements as it encompasses both technical knowledge and operational procedures, while also offering practical value to individuals engaged in customs and trade operations.

I used a RAG system as the project methodology, as my documents consisted of laws and regulations published in the Official Gazette. This approach proved more effective for this type of structured legal content. However, when I initially used the PDF documents as-is, I encountered chunking errors and hallucinations. The legal PDFs contained numerous repetitions and cross-references to previous articles, which disrupted the system's performance. To address these issues, I restructured the PDFs to include only customs-related sections and created a JSON database containing GTiP codes along with general customs duty and VAT rates for various products. Despite these improvements, I continued to receive incorrect answers due to chunk fragmentation issues. This led me to further refine the PDF documents by reorganizing them into an article-based format (Madde 1, Madde 2, etc.) that specifically covered customs and tax-related sections. This article-based structure enabled the system to retrieve information directly from the relevant article in the PDF when addressing technical queries. Additionally, I optimized the system prompt to properly interpret the GTiP database and tested the system with questions of varying difficulty levels—ranging from simple to intermediate and complex scenarios. This iterative testing and refinement process ensured robust performance across different query types.

Example of Trace Outputs:

1- QUESTION: 50 akilli telefon ithal etmek istiyorum. Birim fiyat 800 USD, kur 34 TL, toplam ağırlık 10 kg, hava kargo. Toplam maliyet ne kadar?

Thought: Öncelikle akilli telefonun GTiP kodunu bulalım

Action: search_gtip_database: akilli telefon

-- running search_gtip_database akilli telefon

[GTiP SEARCH] Aranan: 'akilli telefon'

Observation: 'akilli telefon' için bulunan ürünler:

1. Akıllı Telefon GTiP: 8517.12.00 Gümrük Vergisi: %0 KDV: %20

Thought: GTiP kodu ve vergi oranları bulundu. Şimdi CIF değerini hesaplayalım: $50 * 800 * 34$

Action: calculator: $50 * 800 * 34$

-- running calculator $50 * 800 * 34$

[CALC] Islem: $50 * 800 * 34$

Observation: 1,360,000.00

Thought: CIF değeri hesaplandı. Şimdi gümrük vergisi ve KDV'yi hesaplayalım: 1,360,000, 0, 20

Action: calculate_customs_duty: 1360000, 0, 20

-- running calculate_customs_duty 1360000, 0, 20

[TAX CALC] CIF: 1360000.0 TL, Gumruk: %0.0, KDV: %20.0

Observation: CIF: 1,360,000.00 TL, Gumruk: 0.00 TL, KDV: 272,000.00 TL, TOPLAM: 1,632,000.00 TL

Thought: Gümrük vergisi ve KDV hesaplandı. Şimdi hava kargo için navlun maliyetini hesaplayalım: 10 kg, hava kargo

Action: calculate_shipping_cost: 10, hava

-- running calculate_shipping_cost 10, hava

[SHIPPING] 10.0 kg, Yontem: hava

Observation: HAVA: 450.00 TL

Thought: Navlun maliyeti hesaplandı. Şimdi toplam maliyeti hesaplayalım: 1,632,000 + 450

Action: calculator: $1632000 + 450$

-- running calculator $1632000 + 450$

[CALC] Islem: $1632000 + 450$

Observation: 1,632,450.00

Thought: Toplam maliyet hesaplandı.

Answer: Toplam maliyet 1,632,450.00 TL'dir.

FINAL ANSWER:

Toplam maliyet 1,632,450.00 TL'dir.

2 QUESTION: 20000 TL CIF degeri olan bir urun icin gumruk vergisi %5, KDV %20 ise toplam ne kadar vergi ederim?

Thought: First, calculate customs duty

Action: calculator: $20000 * 0.05$

-- running calculator $20000 * 0.05$

[CALC] Islem: $20000 * 0.05$

Observation: 1,000.00

Thought: Now, calculate VAT (KDV) on the total value including customs duty

Action: calculator: $(20000 + 1000) * 0.20$

-- running calculator $(20000 + 1000) * 0.20$

[CALC] İşlem: $(20000 + 1000) * 0.20$

Observation: 4,200.00

Thought: Now, calculate the total amount including CIF value, customs duty, and VAT

Action: calculator: $20000 + 1000 + 4200$

-- running calculator $20000 + 1000 + 4200$

[CALC] İşlem: $20000 + 1000 + 4200$

Observation: 25,200.00

Thought: I have all the necessary information to calculate the total amount

Action: None needed, I can provide the answer directly

Answer: Toplam vergi ve CIF dahil 25,200.00 TL olur.

FINAL ANSWER:

Toplam vergi ve CIF dahil 25,200.00 TL olur.

I have 2 traces outputs examples here, and 12 examples in the notebook file. Here, 1-observation 2-thought 3-action and loop respectively, if there is an answer, it goes to the answer section.

Challenges

The agent found the answer but failed to output ,instead writing "Action:None" and stopping without providing a response. I prefer to enhance the system prompt with explicit rules for solution .

KRİTİK KURALLAR:

1. Her Observation'dan sonra MUTLAKA şunlardan BİRİNİ yap:
 - a) Yeterli bilgi varsa "Cevap: [cevabın]" yaz
 - b) Daha fazla bilgiye ihtiyacın varsa "Action: araç_adı: girdi" yaz
2. ASLA sadece "Thought:" ile bitirme - MUTLAKA "Cevap:" veya "Action:" ekle
3. ASLA "Action: None" yazma - bu GEÇERSİZ!
4. KISMEN bilgi bulsan bile "Cevap:" yaz ve bulduğunu söyle
5. Mükemmel bilgi bekleme - eldeki verilerle cevapla
6. "Cevap:" konuşmayı bitirmenin TEK geçerli yolu

Second challenges i configured is that rag chunking and information loss. Because the primary sources for this project were laws, there were often repetitive clauses, and because it was divided into sections, problems were occurring in the chunk partitioning. To solve this problem, I made adjustments to my PDF sources and opted to split the chunk under "Madde-x" instead of using individual words. After correcting both my chunk algorithm and my PDF, I was able to get correct answers.

Third challenges is Token Rate Limit. During test execution , the API hit dailt token limit (100k tokens/day) causing the test to fail mid-execution . Even though i tried changing it to the llama-3.1-8b model , it's too weak for ReAct tasks, struggles with tool selection and long-context tracking . So that i just wait to earn my token wait.

Fourth challenges is utf-8 and Turkish response. In the JSON file object names are turkish so that when agent try to matching words it can't find true object . I adjusted it with add turkish normalization part of the main code .

Fifth challenges is the calculator multiple expression error. Agent attempted multiple calculations simultaneously but received errors. I adjust this to trained the agent through system prompt.