UFG

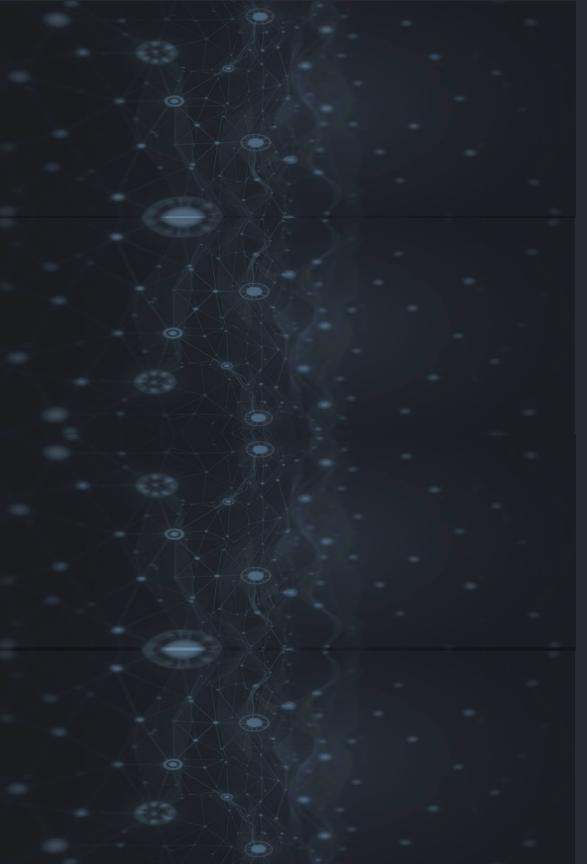
Pós-Graduação em Sistemas e Agentes Inteligentes

Disciplina

Engenharia de Prompt

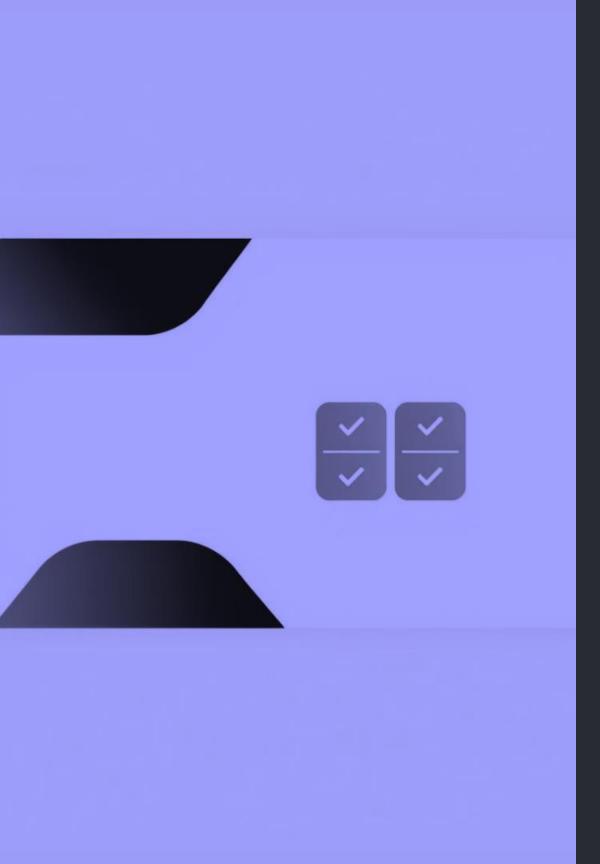
Professor Sanderson Macedo

- Marcu Loreto
- Leandro Lima
- Ricardo Kerr



Assistente Virtual de Suporte Técnico com OpenAl e Guardrails

Esta apresentação demonstra como integrar um modelo de linguagem (LLM) da OpenAl a um assistente de suporte técnico, utilizando guardrails para valididação.



Resumo e Objetivo

Objetivo

Implementar um assistente virtual de suporte técnico que utiliza a OpenAl para fornecer respostas consistentes e personalizadas aos usuários.

Funcionalidade

O sistema recebe perguntas do usuário, pesquisa informações relevantes e fornece respostas que incluem o slogan "Resetou ta Novo".

Guardrails

O código utiliza guardrails para garantir que as respostas da LLM estejam de acordo com as diretrizes e incluam o slogan definido.

Benefícios

O sistema oferece suporte técnico eficiente, automatizado e consistente, melhorando a experiência do usuário.

Configuração do Ambiente

Carregamento de Variáveis

O código utiliza a biblioteca dotenv para carregar variáveis de ambiente, incluindo a chave da API da OpenAI (OPENAI_API_KEY) armazenada em um arquivo .env.

Validação da Chave de API

Antes de iniciar o sistema, o código verifica se a chave foi encontrada, levantando um erro caso contrário. Isso garante que a comunicação com a API da OpenAI seja estabelecida corretamente.

Instanciação do Cliente OpenAl

Após validar a chave, um cliente OpenAl (objeto client) é criado, permitindo enviar requisições ao modelo de linguagem.

		The second secon			CACHA INGRES	war and a second second
Resal	9	Tell Rea	Person, Calapan'T	Wess Teas		
Retail	3	Teg Male	Fanny Consport	Thelier 19195	AM TARI)	/
Paset	3	Tog Fittes	Petty, Cringert	Weht178855	Hagas: [14]	AREY)
Retal	9	Tag Note	Petey, Caoquert	Hehe: 16965	Fanty Tard]	Meliosten, com, Resitten Estabolis, and
Resail	6	Tog Parks	Rasily, 200sgart	Hep: 120095	Ysyr tittd)	MetGS()
Secal	0	Teg Foirles	Petoy, Caergert	Rek 1728085	night fael)	Meaning Retalts A. CDR, and (eraherte)
Called	0	Tag Phymenge	Postby, Chiggert	Walter 16835	(Convia. 5)	GRSW, (onitcommpment to maste)
* Saest	9	Tog Peorlases	Petty, Catigart	Heric: 18955	recomitee[]	
Const	8	Tag Palle	Petrky, Cooggant	Well 1220014	recomts:5)	OH7L
Canet	0	Teg Sestars	Roldy, Steigert	Wells: 16535	rccomiae()	ARCEL
Daite	0	Tas Alle	Poony, Casigors	Wan: 172035	(Painker)	
Gett 5	0	Tog Victo	Potey, Cangert	Hell 1220955	notert (18)	Mayte Regsertion, sils, comoytayi)
	0	Tog Europes	Rotsy, Consgert	Wegn: 72065	Fredite (1)	
Geali 1	3	Tog Papes	Resey, Chagert	Web 1278003	Heathroy 1AEST	2145)

Lógica de Negócio: Busca de Documentos

Função search_documents(query)

Esta função simula a consulta a uma base de dados local, recebendo uma string (consulta do usuário) e retornando informações relevantes com base em um conjunto de dados pré-definido.

Exemplos de Chaves

As chaves de pesquisa correspondem a problemas comuns, como "máquina não liga" ou "máquina travando", retornando soluções correspondentes.

Benefícios

Esse mecanismo permite contextualizar as respostas do assistente, fornecendo informações específicas para problemas comuns, sem necessidade de reescrever as soluções no código principal.



Prompt Inicial e Regras de Atendimento



Contextualização

O prompt inicial define o assistente como um especialista em suporte técnico, definindo claramente seu papel e responsabilidades.



Instruções Detalhadas

O prompt lista objetivos específicos (identificar problemas, coletar dados do cliente, incluir slogan), fornecendo diretrizes claras.



Tom de Voz

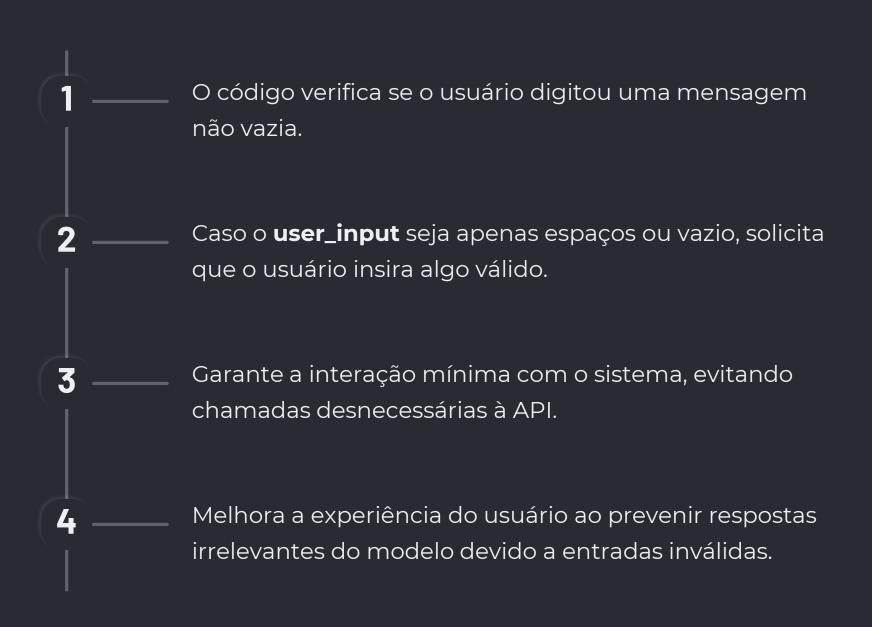
O prompt define que o atendimento deve ser claro, educado e objetivo, garantindo respostas consistentes e alinhadas aos valores da assistência.

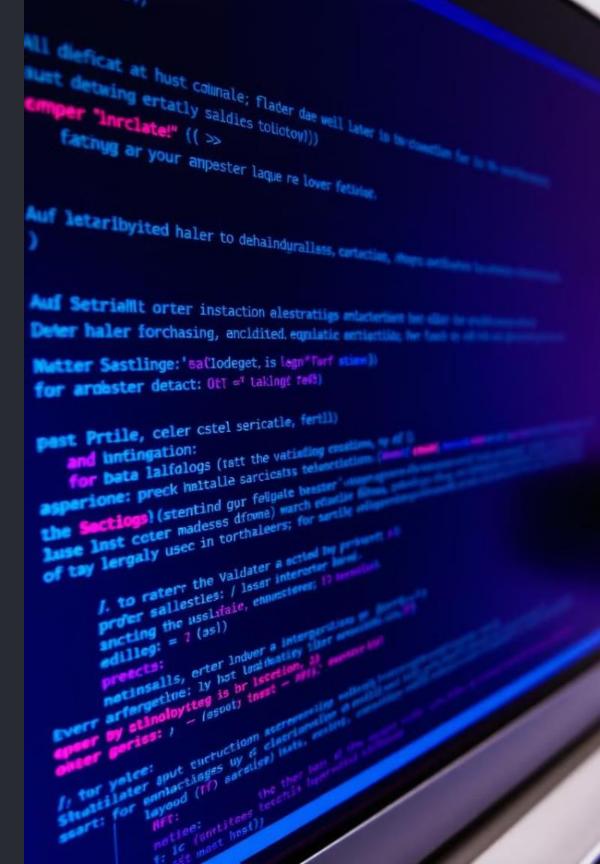


Memória Contextual

A exigência de incluir o slogan "Resetou ta Novo" e o endereço da assistência técnica é uma técnica de prompting que garante a fidelidade às diretrizes em cada resposta.

Validação da Entrada do Usuário





```
Adulit Crarcaguet Vised Cho
        17 forusstalcal (
                  Vartialls: {
                  Variieveelt fealog: ws[
                   Vaalicational (f feet Commonwation -7 tool Cacal regulation)
appli Connecivat Indne- Tacedion: - (Mayler Gazillot, et al.)
                         tatil —alatenment thee relaine to refer mode
                        tatil requests: 'logumel; '- Eaplocion, mroflest tatit -alceniccattion, facalisopetion: 'bdp ween
          17
                           Veal ioppe/erapors faxti incertinginisto
          13
                              facuismont (lattls, 1)
  12
          14
                           prceest())
                          datal = trasxputime, latre that feature):
                           putal = lasti. joncecation, network (lestertion) spilt by the special
         17
                           asscencestl) - falugs:)
                           acceetn(1) = <ndtwume |meduarcs:
                          api intsessinal (eata);
                          data teneuastied APPI (Mfert))
                          data rocates, for (iggmeeil) Tata for formular, salars to continue to the
                          data sractinsecilf (igguealstion, whils receive til, on week to week
                          the fostpute intee rescuets to cration:
                      on intresstealsnegter, ass, in inproperties the lifedsarcons, lest. Chaistary List concitons or raftreef cusel, the lescars!)
                          coplecalsnentiocy; vescaptement
```

Interação com a LLM e Guardrails

Chamadas à API OpenAI

O código utiliza client.chat.completions.create para gerar respostas com base no histórico de mensagens.

Validação com Guardrails

2 O Guard() com o validador ContainsString(substring="Resetou ta Novo") garante que a resposta da LLM inclua o slogan definido.

Benefício da Validação

Isso garante conformidade no conteúdo, evitando que a LLM esqueça o slogan ou traga respostas em desacordo com o prompt inicial.

Demonstração do Fluxo de Execução

1

Entrada do Usuário

O usuário digita uma mensagem descrevendo seu problema.

2

Validação

O código valida se a entrada é válida (não vazia).

3

Busca de Informações

Busca informações relevantes na base simulada, adicionando-as ao contexto se necessário. 4

Resposta da LLM

O sistema envia o histórico para a LLM via API, gera uma resposta e a valida com Guardrails.



Conclusão

O código demonstra como integrar um modelo de linguagem (LLM) em um cenário de atendimento técnico.

As técnicas de prompting e o uso de validações com Guardrails asseguram a qualidade, relevância e consistência das respostas.

A abordagem garante que informações essenciais (como slogan) sejam sempre incluídas, mantendo a identidade e profissionalismo do suporte.