




MultiDoc- QA

Monique Monteiro –
moniquelouise@gmail.com



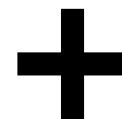


Conceitos importantes

-
- Capacidade few-shot de LLMs pode reduzir os custos para resolução de tarefas da *question answering* (QA)
 - Implementação de sistemas de QA para diferentes domínios sem a necessidade de datasets específicos anotados
 - Melhor desempenho quando modelos são induzidos a mostrar evidências
 - *CoT (Chain-of-Thought)*
- 

Técnicas para garantir corretude

- Inspeção visual:
 - Da composição das perguntas
 - Das evidências geradas



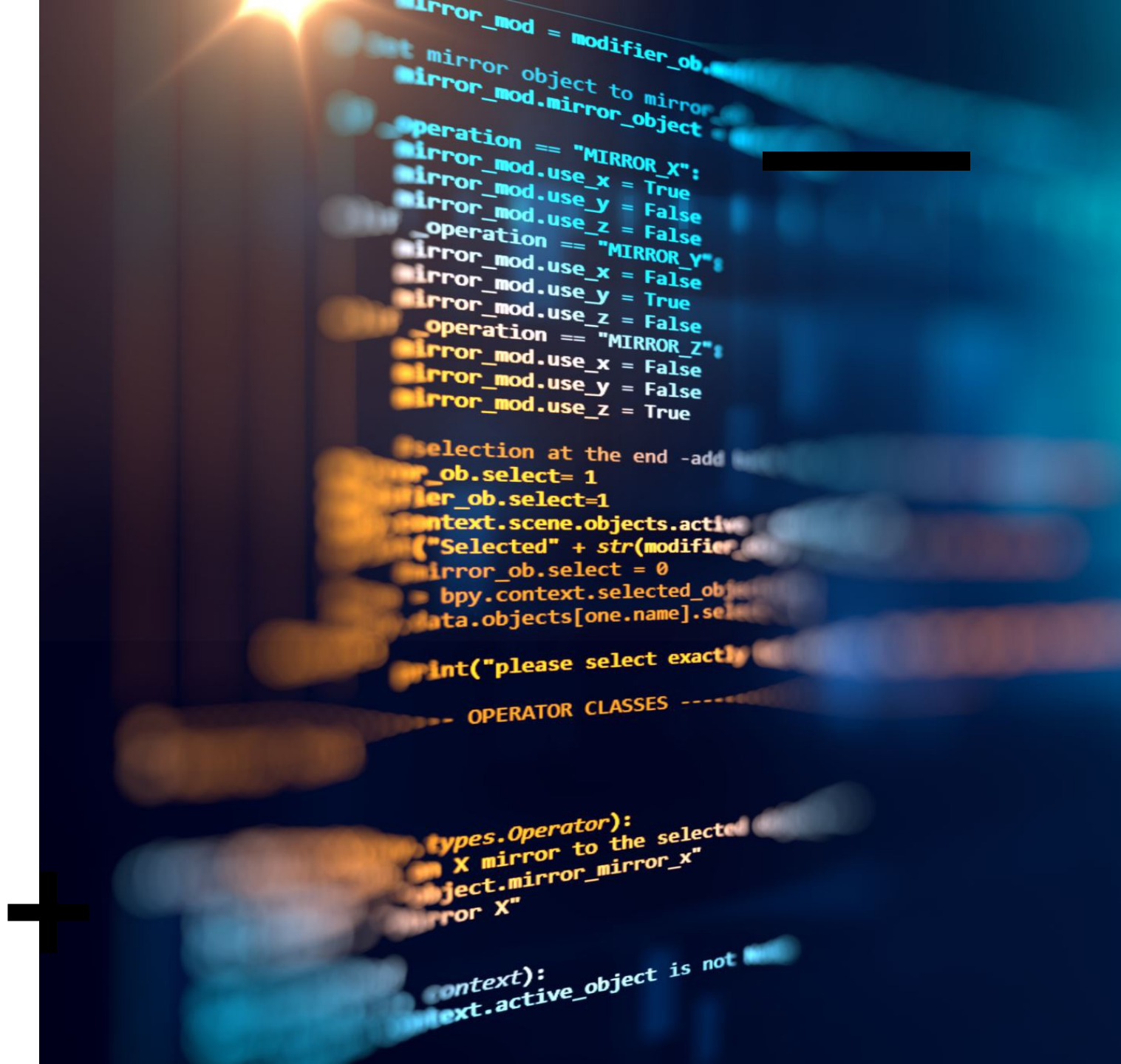
Truques de código que funcionaram

- monoT5 e bibliotecas do pygagle possuem incompatibilidade com demais bibliotecas
- Solução:
 - Colocar em notebook separado



Implementação

- Inspirada pelo [repositório do Viscode](#)
 - Parte do código de pré-processamento, indexação e reranqueamento
 - Código de coleta de métricas
- Diferenciais
 - Chamadas à API da OpenAI (GPT 3.5 turbo)
 - Modelada como diálogo
 - Tratamento para mensagens longas
 - Hiperparâmetros
 - Remoção de tags HTML



Pipeline

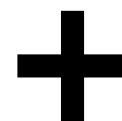
```
mirror_mod = modifier_ob.  
# Set mirror object to mirror  
mirror_mod.mirror_object =  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True  
  
# selection at the end - add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob.  
mirror_ob.select = 0  
bpy.context.selected_object  
data.objects[one.name].select  
  
print("please select exactly  
  
-- OPERATOR CLASSES ----  
  
types.Operator):  
X mirror to the selected  
object.mirror_mirror_x"  
mirror X"  
  
context):  
context.active_object is not
```

1. Indexação
 - Conjunto de teste
 - Texto principal + artigos associados aos links
 - Salvamento do índice com janelas de context
2. Rerankeamento
 - Decomposição das perguntas (máx. 50) (GPT 3.5 turbo com prompt estático)
 - Rerankeamento com monoT5
3. Inferência/Avaliação
 - Uso de dataset de evidências pré-disponibilizado para construção de prompts dinâmicos com K exemplos mais próximos (K=4)
 - Geração de respostas + evidências com GPT 3.5 turbo

Resultados (inesperados)

- Melhor F1 para documentos com tags HTML (?)
- Hipótese:
 - Alguma aleatoriedade no pré-processamento, indexação ou chamadas ao GPT

Presença de tags HTML	F1	EM
Sim	0,4529	0,34
Não	0,4319	0,36



Dúvida básica

Não seria a geração de evidências *zero-shot learning*?

