Code Presentation

IA368DD_2023S1: Deep Learning aplicado a Sistemas de Buscas

Student: Marcus Vinícius Borela de Castro

Multi-doc-qa

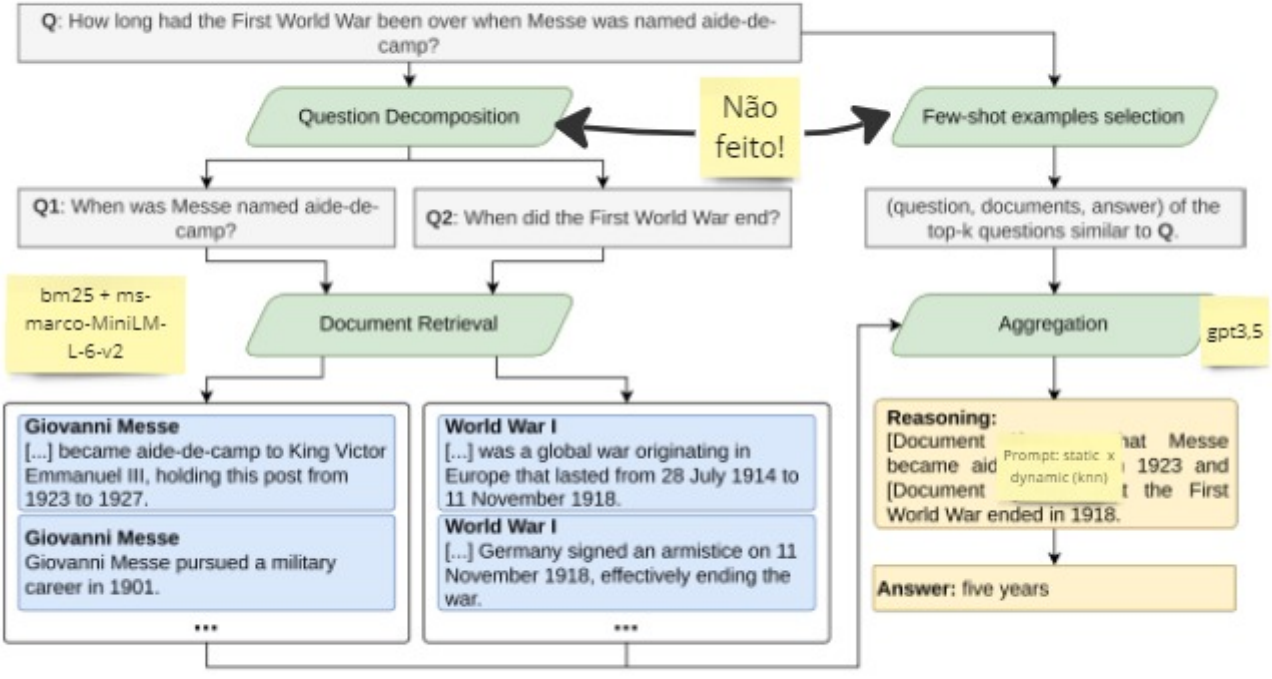


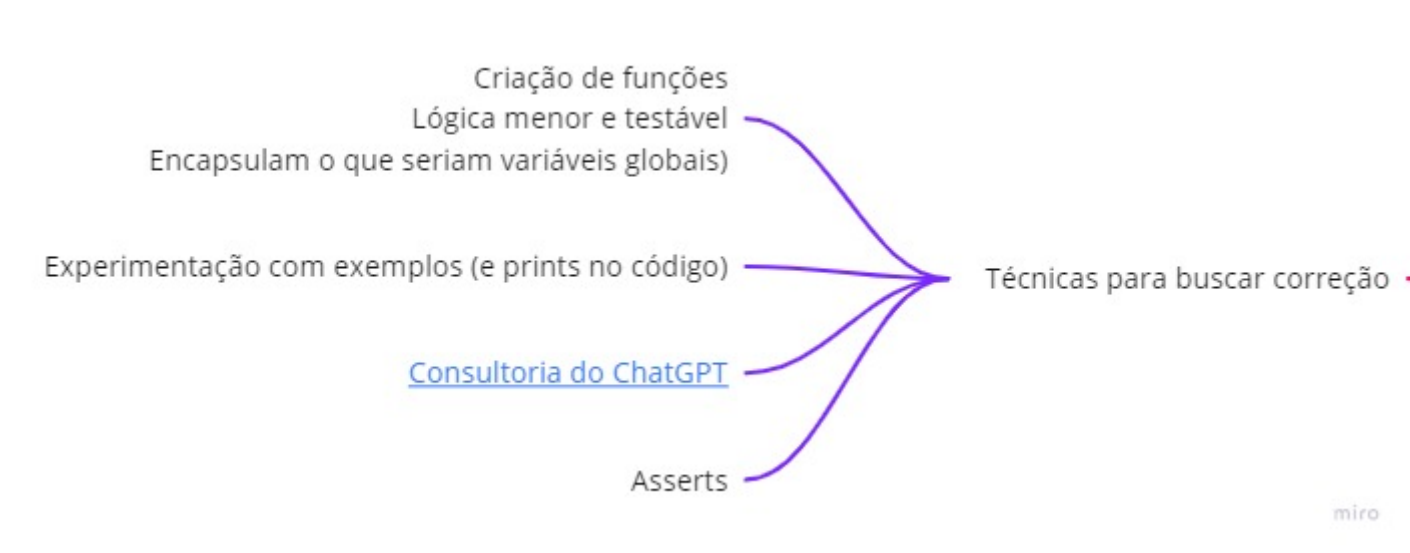
Fig. 1: Visconde QA flow.

Modelo usado: text-davinci-002 (gpt 3.5)

Similar capabilities to text-davinci-003 but trained with supervised fine tuning instead of reinforcement learning

model text-davinci-003 (gpt 3.5)

Can do any language task with better quality, longer output, and consistent instruction-following than the curie, babbage, or ada models. Also supports inserting completions within text.


Criação de funções
Lógica menor e testável
Encapsulam o que seriam variáveis globais)
Experimentação com exemplos (e prints no código)
Consultoria do ChatGPT
Asserts
Técnicas para buscar correção
miro

Critéro para seleção de amostra (20, 10 de cada tipo: binary/value)
ind_complexidade = num_context + num_context_links + round(doc_len_text/500)

Describe():
count: 43.0 mean:6.3 std:1.5 min: 4 25%:5 50%: 6 75%: 7 max: 15

Truques de código

Experimentado e avaliado, além do critério **Retrieval: Gold CTX with CoT**
Comecei por ele para validação do pipeline

Problemas encontrados e soluções

Percebi nos meus resultados que o critério usado no código original (Visconde) para identificar respostas :

```
pattern = "(?<=Answer:)(.*)$"
matches = re.findall(pattern, res)
```

Não identificava todas as respostas. Como o caso abaixo:

```
"(...)_IXs.\n\nAnswer:\n\nYes."
```

Pois, aparentemente (.*) não casa com '\n' para findall.

Mudei para: `item['results'].split("Answer:")[1].strip()`

Mas acredito que se acrescentar flags=re.MULTILINE também resolva.

Percebi no test_set[24], uma pequena incorreção em sua segunda pergunta:
A chave 'context' aponta para apenas 2 contextos: um main e outro de um link "Jerry Reinsdorf". Mas na chave "question_links' aparece o link 'Chicago White Sox' que não consta da lista de contextos para responder à pergunta, embora essa substring apareça no texto da passagem indicada.
(sem impacto nos resultados)]

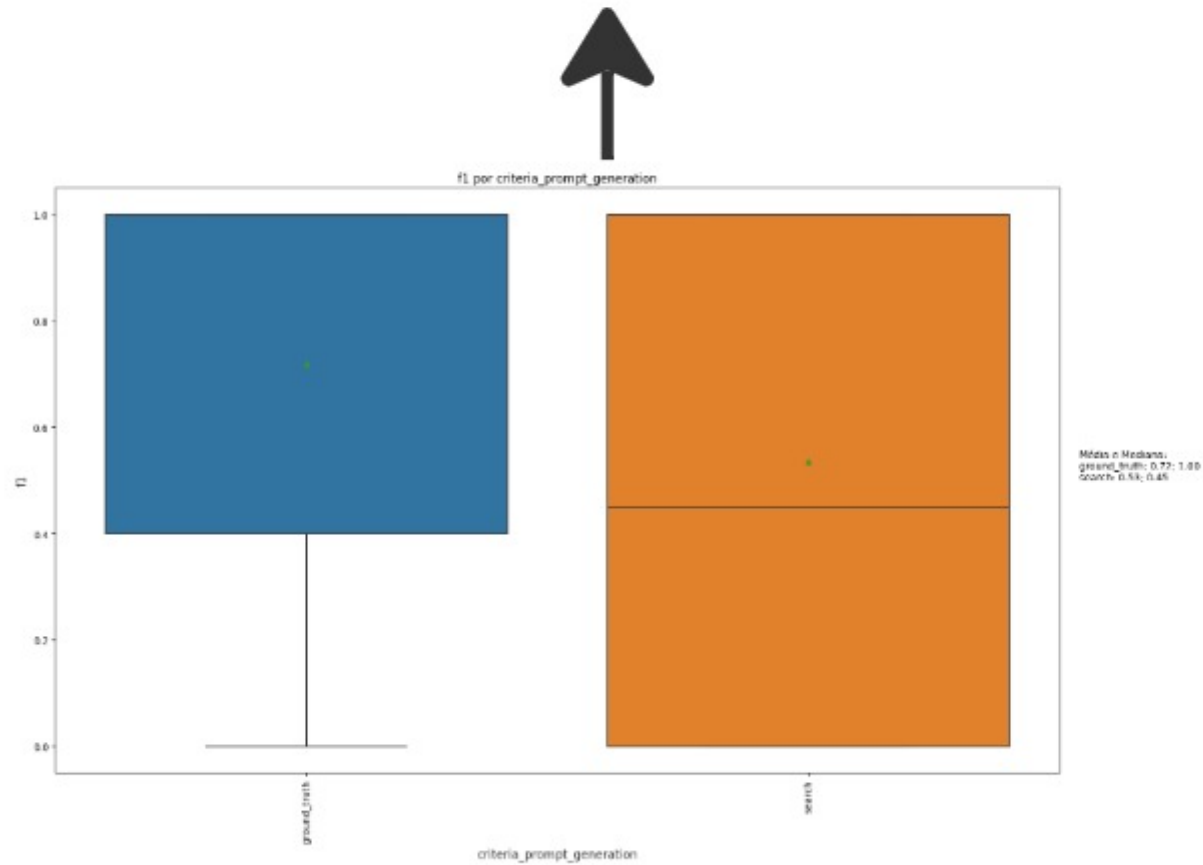
Médias das métricas (por critério)
ground_truth EM: 65% F1:71,85%
search EM: 45% F1:53,50%

Médias das métricas (por tipo de questão)
binary EM: 70% F1:70%
value EM: 40% F1:55,35%

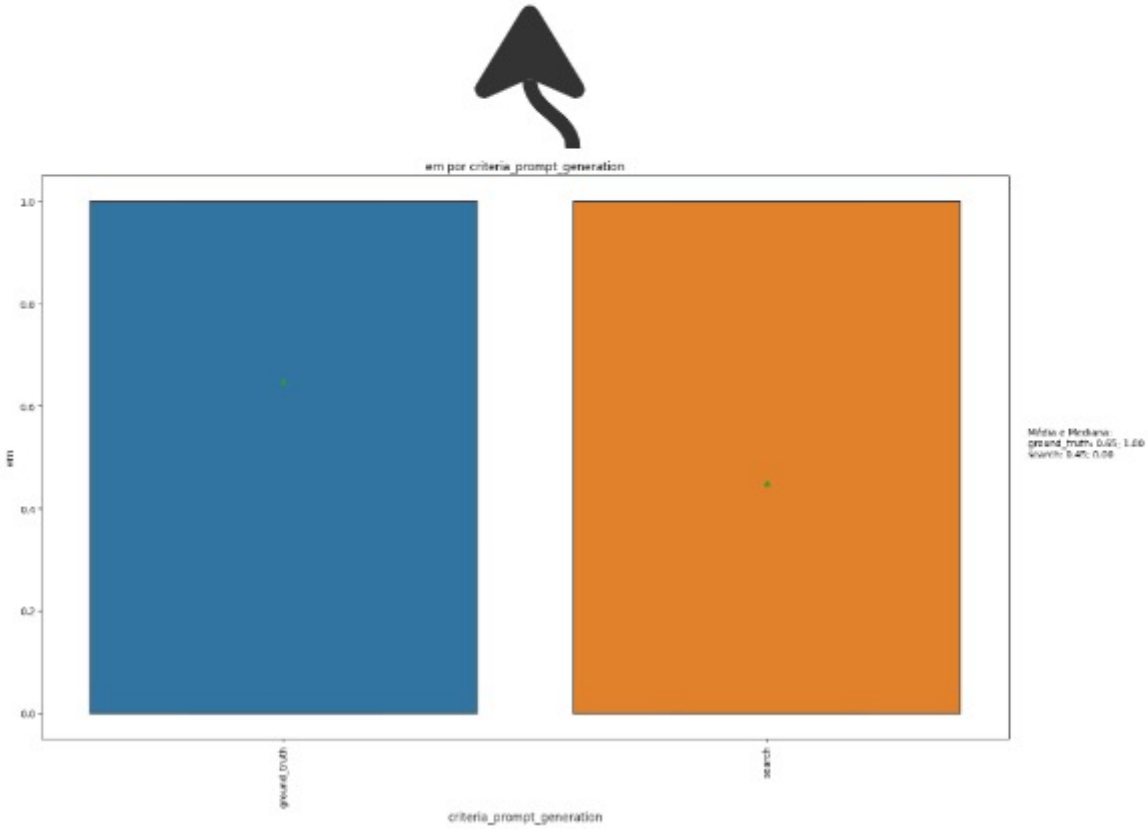
- Resultados interessantes

		em								f1							
		count	mean	std	min	25%	50%	75%	max	count	mean	std	min	25%	50%	75%	max
criteria_prompt_generation	type_answer																
ground_truth	binary	10.0	0.7	0.4830459	0.0	0.25	1.0	1.0	1.0	10.0	0.7000000	0.4830459	0.0	0.2500000	1.0	1.000	1.0
	value	10.0	0.6	0.5163978	0.0	0.00	1.0	1.0	1.0	10.0	0.7371429	0.3670808	0.0	0.4428571	1.0	1.000	1.0
search	binary	10.0	0.7	0.4830459	0.0	0.25	1.0	1.0	1.0	10.0	0.7000000	0.4830459	0.0	0.2500000	1.0	1.000	1.0
	value	10.0	0.2	0.4216370	0.0	0.00	0.0	0.0	1.0	10.0	0.3700000	0.3888730	0.0	0.0000000	0.4	0.475	1.0

F1 por critério prompt



EM por tipo de questão



Dúvidas básicas
(projeto final)

Qual monot5 ranker é melhor?

1. mt5-3B-mmarco-en-pt (model.bin:15gb)
2. mt5-base-mmarco-v2 (2.17gb)
3. ptt5-base-pt-msmarco-100k-v2 (850 mb)

Compensa usar o de 15gb?

Qual a diferença entre um mt5 e ptt5? — Seria o modelo base ser um multilingual ou um treinado em português?

Esses monot5 podem ser treinados como Causal Language Modeling? — Precisaria tirar a última camada que trata a relevância?

Tópicos avançados

Métricas poderiam ser repensadas.
Para tratar uma "injustiça lógica"
Injusto:
Alto F1: 35 years x 40 years
EM zero: 35 x 35 years

Para se adequar à prolixidade dos LLM: EM@Contains