Código após Aula 3

IA368DD_2023S1: Deep Learning aplicado a Sistemas de Buscas

Marcus Vinícius Borela de Castro

Resolvendo Tarefas com LLM (Large Language Model) de Maneira Zero e Few-shot

Extractive Q&A

Performance Comparison between Learning Methods Context and Transfer Learning

Leonardo Augusto da Silva Pacheco
Marcus Vinícius Borela de Castro

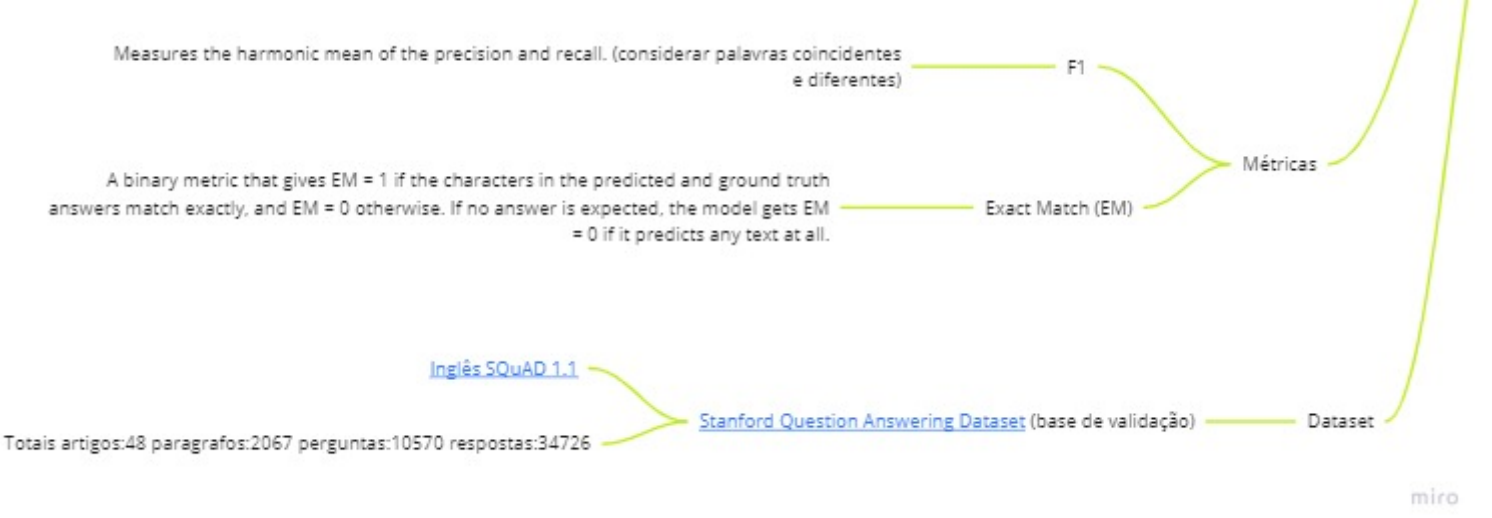
Final Project at Discipline IA025 Introdução ao Aprendizado Profundo (2022/01)

Link para github do projeto

Ver projeto de dixciplina anterior

Conceitos

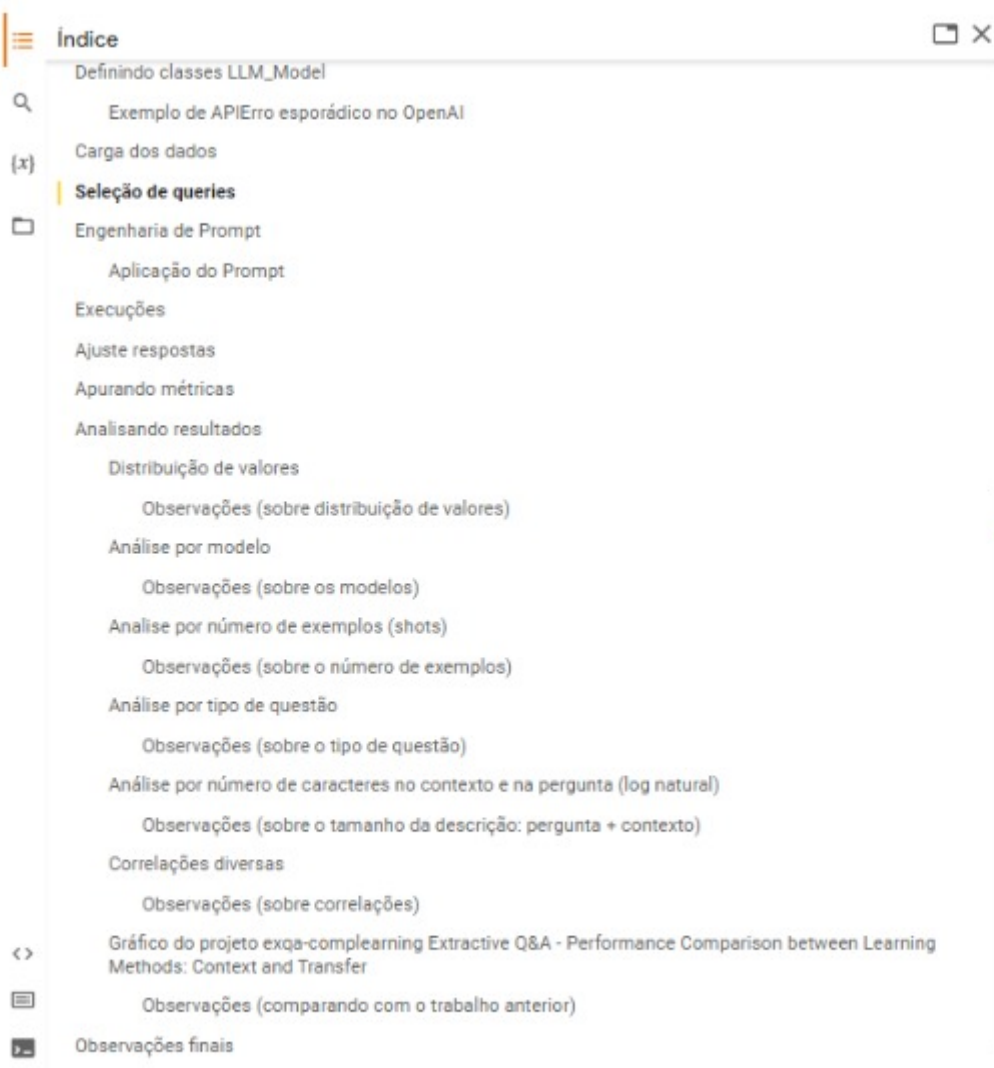
https://github.com/leonardo3108/extractive-qa/blob/main/README.md


Measures the harmonic mean of the precision and recall. (considerar palavras coincidentes e diferentes)
F1
A binary metric that gives EM = 1 if the characters in the predicted and ground truth answers match exactly, and EM = 0 otherwise. If no answer is expected, the model gets EM = 0 if it predicts any text at all.
Exact Match (EM)
Métricas
Inglês SQuAD 1.1
Stanford Question Answering Dataset (base de validação)
Totais artigos:48 paragrafos:2067 perguntas:10570 respostas:34726
Dataset
miro

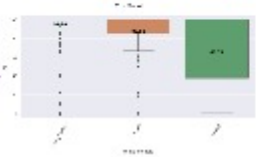
Criar classes para abstrair características específicas dos LLM
Exemplo:

```
class LLM_OpenAi_Model_QA:
def __init__(self, name:str, max_tokens:int, temperature:int=0, top_p:int=1):

def run_one_question(self, parm_prompt:str, parm_max_len_output:bool)->tuple:

def answer_one_question(self, parm_prompt:str, parm_max_len_output:bool=None)->str:
    answer, _ = self.run_one_question(parm_prompt, parm_max_len_output)
    return answer

def assert_limite_respeitado(self, parm_text:str, tamanho_max_resposta):

def conta_token(self, parm_text:str):
```

Truques de código

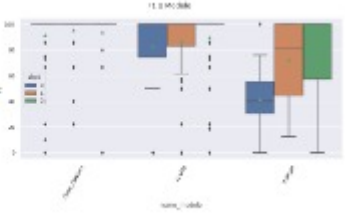


Estruturar usando o índice (imagem parcial)

- Resultados interessantes
  - Ver gráficos no caderno
  - Análise modelos
    - . O F1 do Code_DaVinci ficou bem superior aos demais (93,78%), mas acredito que o ChatGPT, por ser prolixo, foi prejudicado, visto que sua precisão (parte da fórmula do F1) é prejudicada.
    - . O EM do ChatGPT foi prejudicado também por esse motivo.
    - . Interessante ter agregado a métrica EM_Substr que considera como 100% se o a resposta dada contiver a resposta esperada. Dessa forma foi feito justiça ao "prolixo" ChatGPT. Claro que uma engenharia de prompt ou fine-tunning poderiam ser experimentados para retirar essas sobras de palavras.
    - . Nessa nova métrica proposta (EM_Substr), todos os modelos se saíram bem. Com uma diferença de 8 pontos para os modelos da OpenAi.
    - 

Análise "shot"

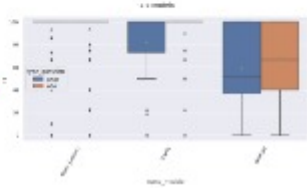
. Em média, os modelos foram melhor com 2-shot em todas as métricas apuradas.
. ChatGPT foi o que melhor respondeu ao aumento do número de exemplos (de 1 para 2).
. o Code_DaVinci foi o que pior respondeu em relação ao uso de exemplos contra 0-shot (houve até decréscimo no EM_Substr).



Análise por tipo de questão

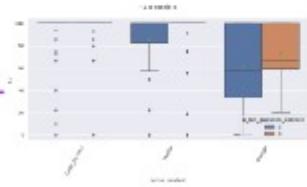
. Em média, os modelos foram melhores nas perguntas do tipo "who" do que nas do tipo "what". A métrica mais sensível foi a EM (10 pontos), indicando que o "what" pode a uma maior prolixidade.

. o Code_DaVinci foi o menos sensível ao tipo da pergunta (houve até decréscimo no EM_Substr).



Análise sobre o tamanho da descrição: pergunta + contexto

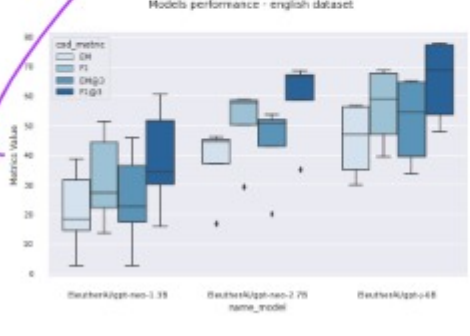
. Apenas o Code_DaVinci foi pior nas respostas para descrições maiores nas métricas. Os outros dois acertaram mais com descrições maiores. Esse é um bom ponto a ser investigado.



Análise sobre correlações

. Aparentemente a métrica proposta EM_Substr tem uma correlação com o tamanho da resposta e co o tamanho da pergunta. Em relação ao tamanho da resposta, justifica o aumento pois aumenta a chance de englobar a resposta certa.
. As métricas possuem uma certa correlação entre si. A métrica EM_Substr tem maior correlação com a F1 do que com a EM. É compreensível pois a F1 traduz melhor textos semelhantes nas duas respostas.

Comparando com projeto anterior



. Percebe-se um sensível aumento nos valores das métricas para os novos modelos, dada sua maior capacidade (treinamento realizado e número de parâmetros).
. A métrica F1 aumentou até 43 pontos percentuais: passou de 58 (gpt-j-6B) para 93,78 (Code_DaVinci) e 86,51 (LLaMa).
. A EM aumentou até 38 pontos percentuais: passou de 48 (gpt-j-6B) para 86,11 (Code_DaVinci) e 72,78 (LLaMa).
. O impacto maior na F1 é justificado pois os modelos podem responder mais palavras do que o esperado. Por isso, a métrica proposta EM_Substr se mostra uma boa alternativa nessas comparações. Mas não havia esses valores no projeto anterior, até porque os modelos foram testados no arcabouço pipeline"text-generation" da library transformer.
. Fica para tabalho futuro uma análise mais detalhada: restrita às perguntas selecionadas neste trabalho e trazendo também para a comparação resultados do modelo distilbert-base-cased-distilled-squad.

Dúvidas

Para o Code_DaVinci precisei colocar um sleep (20 segundos) devido a erro de estouro do limite de chamadas por minuto. Fiquei na dúvida: esse limite é porque ele é gratuito? Será que existe também para versões pagas como o ChatGPT (para ele usei 5 segundos).
Mas não testei limites.

Propusemos métrica EM_Substr em detrimento à métrica EM (Exact_Match) para tratar possibilidade da resposta dada conter a resposta ground thruth, já que o chatgpt acrescenta informações na sua resposta (prolixo!).
Já há alguma métrica para esse caso?
Parece interessante se pensar também em F1_Substr tratando melhor o erro de precisão por palavras adicionais.