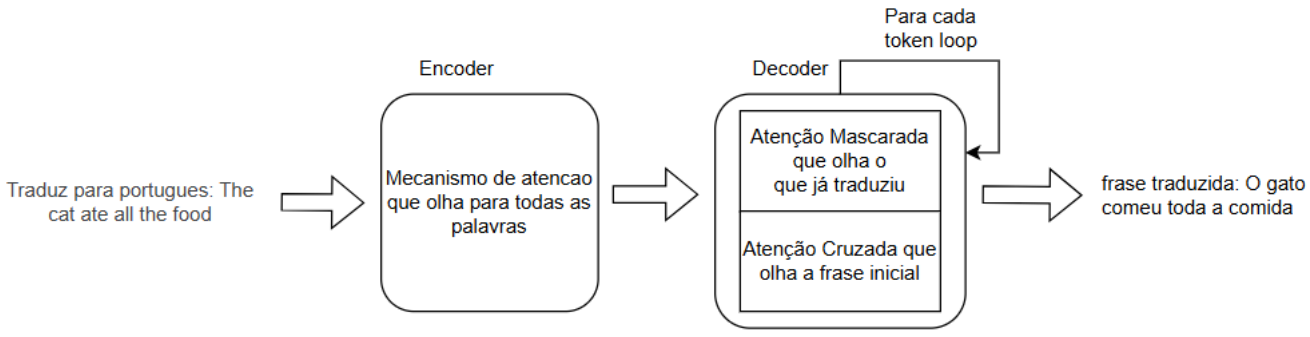
Linguagem Natural e LLMs [25E2-25E2]

Aluna: Rachel Reuters

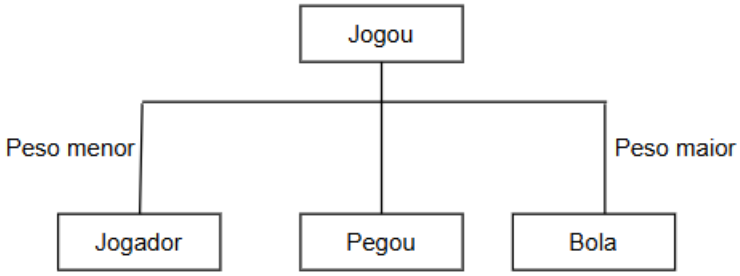
Fundamentos das LLMs

1) Explique os seguintes conceitos fundamentais dos LLMs, fornecendo exemplos práticos e diagramas onde for relevante:

- Pre-training: Primeira etapa em que o modelo aprende sobre a linguagem humana de forma não supervisionada (sem classificação previa). O objetivo ‘e ter uma compreensão básica de gramática, semântica e fatos. Exemplo: O modelo recebe a frase "O céu é..." e sua tarefa é prever a próxima palavra mais provável, que seria "azul". Ao fazer isso bilhões de vezes, ele aprende as relações estatísticas entre as palavras.
- Transfer Learning: estratégia de transferir o conhecimento obtido na fase de pre-treinamento para uma nova tarefa mais específica. Em vez de treinar um modelo do zero para cada problema, aproveita-se a base de conhecimento geral já existente. Exemplo: diagnosticar se uma tomografia do útero representa sinais de câncer ou não. A primeira etapa seria o treinamento de tomografia de úteros, sabendo identificar se representa ou não um útero. Na etapa de transfer learning seria utilizar esse modelo e aplicar o conhecimento de câncer através da imagem de tomografia.
- Embeddings: são representações numéricas de palavras, frases ou textos inteiros em um espaço vetorial. Quanto maior semelhança mais próximos serão os vetores nesse espaço. Exemplo: representações vetoriais entre casa e palácio como pode ser visto no diagrama Embedding projector - visualization of high-dimensional data.
- Transformers: Arquitetura projetada para processar dados sequenciais, como texto, e entender o contexto e as relações entre todas as partes dessa sequência. Ele possui um mecanismo de auto atenção que olha para todas as palavras da frase em paralelo e determina as mais importantes para entender o significado de cada palavra individualmente. Internamente ele funciona com encoders e decoders. Exemplo:



- Attention: A técnica que permite que o modelo pese a importância de diferentes palavras na sequência de entrada ao processar uma palavra específica. Ele aprende a prestar atenção nas palavras mais relevantes para entender o contexto, não importando o quão distantes elas estejam na frase. Exemplo: Identificar o que o “a” representa na frase O jogador pegou a bola e a jogou.



- Fine-Tunning: Segunda fase do Transfer Learning. Após o pré-treinamento, o modelo de conhecimento geral é adaptado para uma tarefa específica. Isso é feito treinando-o um pouco mais com um conjunto de dados menor e rotulado, específico para o problema que se quer resolver.

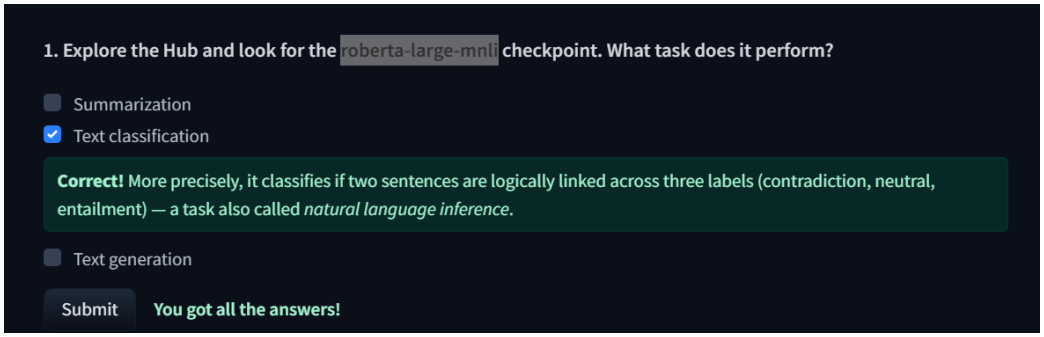
Quizzes do Curso de NLP da Hugging Face

2) Acesse os quizzes dos capítulos 1, 2 e 3 do curso de NLP da Hugging Face através do link: Curso de NLP.

Resolva os quizzes e capture screenshots dos resultados.

Anexe as screenshots a esta avaliação e explique brevemente os conceitos abordados em cada quiz.

Parte 1) O foco principal dessa primeira parte foi apresentar o Hugging face, explicar a diferença entre LLM e NLP, explicar como funciona a biblioteca Transformers, e também apresentar as arquiteturas possíveis da mesma (encoder, decoder e misto).



3. What should replace ... in this code sample?

```
from transformers import pipeline

filler = pipeline("fill-mask", model="bert-base-cased")
result = filler("...")
```

- [ ] This <mask> has been waiting for you.
- [x] This [MASK] has been waiting for you.

Correct! This model's mask token is [MASK].

- [ ] This man has been waiting for you.

Submit **You got all the answers!**

4. Why will this code fail?

```
from transformers import pipeline

classifier = pipeline("zero-shot-classification")
result = classifier("This is a course about the Transformers library")
```

- [x] This pipeline requires that labels be given to classify this text.

Correct! Right — the correct code needs to include `candidate_labels=[...]`.

- [ ] This pipeline requires several sentences, not just one.
- [ ] The 🤗 Transformers library is broken, as usual.
- [ ] This pipeline requires longer inputs; this one is too short.

Submit **You got all the answers!**

5. What does “transfer learning” mean?

- [ ] Transferring the knowledge of a pretrained model to a new model by training it on the same dataset.
- [x] Transferring the knowledge of a pretrained model to a new model by initializing the second model with the first model's weights.

Correct! When the second model is trained on a new task, it *transfers* the knowledge of the first model.

- [ ] Transferring the knowledge of a pretrained model to a new model by building the second model with the same architecture as the first model.

Submit **You got all the answers!**

6. True or false? A language model usually does not need labels for its pretraining.

- [x] True

Correct! The pretraining is usually *self-supervised*, which means the labels are created automatically from the inputs (like predicting the next word or filling in some masked words).

- [ ] False

Submit **You got all the answers!**

7. Select the sentence that best describes the terms “model”, “architecture”, and “weights”.

- [ ] If a model is a building, its architecture is the blueprint and the weights are the people living inside.
- [ ] An architecture is a map to build a model and its weights are the cities represented on the map.
- [x] An architecture is a succession of mathematical functions to build a model and its weights are those functions parameters.

Correct! The same set of mathematical functions (architecture) can be used to build different models by using different parameters (weights).

Submit **You got all the answers!**

8. Which of these types of models would you use for completing prompts with generated text?

- [ ] An encoder model
- [x] A decoder model

Correct! Decoder models are perfectly suited for text generation from a prompt.

- [ ] A sequence-to-sequence model

Submit **You got all the answers!**

9. Which of those types of models would you use for summarizing texts?

- [ ] An encoder model
- [ ] A decoder model
- [x] A sequence-to-sequence model

Correct! Sequence-to-sequence models are perfectly suited for a summarization task.

Submit **You got all the answers!**

10. Which of these types of models would you use for classifying text inputs according to certain labels?

- [x] An encoder model

Correct! An encoder model generates a representation of the whole sentence which is perfectly suited for a task like classification.

- [ ] A decoder model
- [ ] A sequence-to-sequence model

Submit **You got all the answers!**

11. What possible source can the bias observed in a model have?

- [x] The model is a fine-tuned version of a pretrained model and it picked up its bias from it.

Correct! When applying Transfer Learning, the bias in the pretrained model used persists in the fine-tuned model.

- [x] The data the model was trained on is biased.

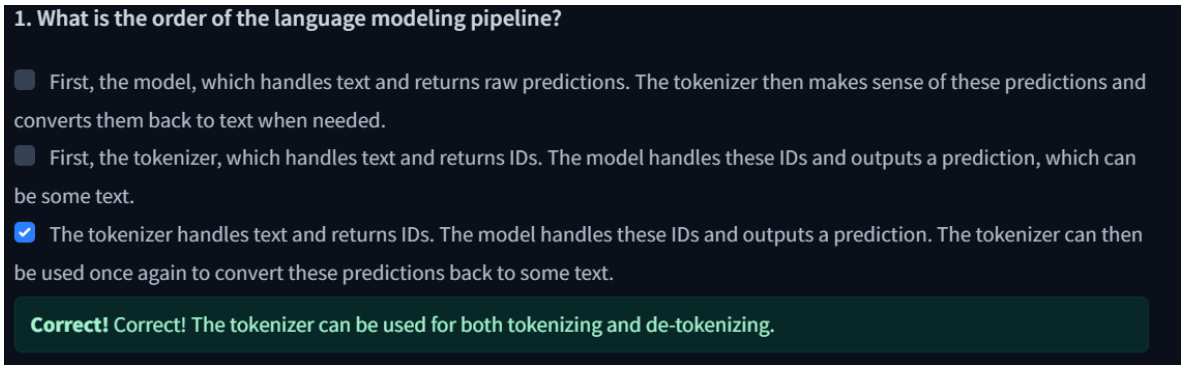
Correct! This is the most obvious source of bias, but not the only one.

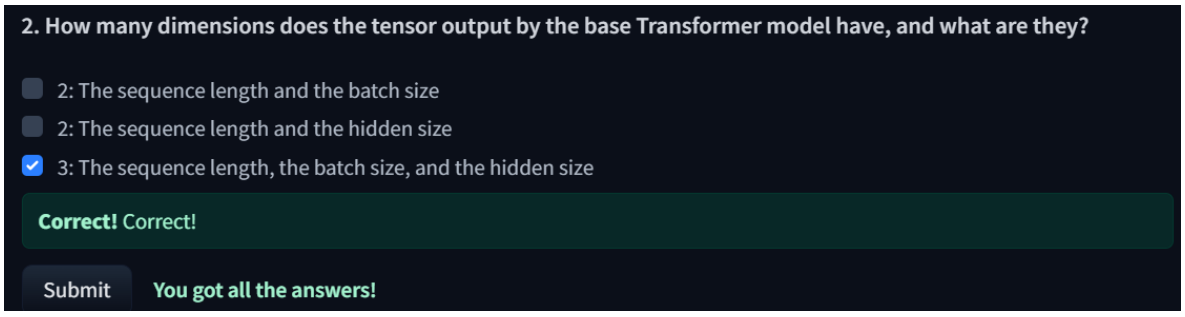
- [x] The metric the model was optimizing for is biased.

Correct! A less obvious source of bias is the way the model is trained. Your model will blindly optimize for whatever metric you chose, without any second thoughts.

Submit **You got all the answers!**

Parte 2) Na segunda parte já entra mais no detalhe de implementação da biblioteca Transformers, explica a diferença entre usar com Pytorch e Tensorflow, mostra alguns casos de uso, explica sobre os Automodels, o processo de tokenização que pode ser aplicado utilizando o modulo AutoTokenizer, explica algumas técnicas para evitar que o modelo quebre devido a textos muito longos.





3. Which of the following is an example of subword tokenization?

- [x] WordPiece

Correct! Yes, that's one example of subword tokenization!

- [ ] Character-based tokenization
- [ ] Splitting on whitespace and punctuation
- [x] BPE

Correct! Yes, that's one example of subword tokenization!

- [x] Unigram

Correct! Yes, that's one example of subword tokenization!

- [ ] None of the above

Submit **You got all the answers!**

4. What is a model head?

- [ ] A component of the base Transformer network that redirects tensors to their correct layers
- [ ] Also known as the self-attention mechanism, it adapts the representation of a token according to the other tokens of the sequence
- [x] An additional component, usually made up of one or a few layers, to convert the transformer predictions to a task-specific output

Correct! That's right. Adaptation heads, also known simply as heads, come up in different forms: language modeling heads, question answering heads, sequence classification heads...

Submit **You got all the answers!**

5. What is an AutoModel?

- [ ] A model that automatically trains on your data
- [x] An object that returns the correct architecture based on the checkpoint

Correct! Exactly: the `AutoModel` only needs to know the checkpoint from which to initialize to return the correct architecture.

- [ ] A model that automatically detects the language used for its inputs to load the correct weights

Submit **You got all the answers!**

6. What are the techniques to be aware of when batching sequences of different lengths together?

- [x] Truncating

Correct! Yes, truncation is a correct way of evening out sequences so that they fit in a rectangular shape. Is it the only one, though?

- [ ] Returning tensors
- [x] Padding

Correct! Yes, padding is a correct way of evening out sequences so that they fit in a rectangular shape. Is it the only one, though?

- [x] Attention masking

Correct! Absolutely! Attention masks are of prime importance when handling sequences of different lengths. That's not the only technique to be aware of, however.

Submit **You got all the answers!**

7. What is the point of applying a SoftMax function to the logits output by a sequence classification model?

- [ ] It softens the logits so that they're more reliable.
- [x] It applies a lower and upper bound so that they're understandable.

Correct! Correct! The resulting values are bound between 0 and 1. That's not the only reason we use a SoftMax function, though.

- [x] The total sum of the output is then 1, resulting in a possible probabilistic interpretation.

Correct! Correct! That's not the only reason we use a SoftMax function, though.

Submit **You got all the answers!**

8. What method is most of the tokenizer API centered around?

- [ ] `encode`, as it can encode text into IDs and IDs into predictions
- [x] Calling the tokenizer object directly.

Correct! Exactly! The `__call__` method of the tokenizer is a very powerful method which can handle pretty much anything. It is also the method used to retrieve predictions from a model.

- [ ] `pad`
- [ ] `tokenize`

Submit **You got all the answers!**

9. What does the result variable contain in this code sample?

```
from transformers import AutoTokenizer

tokenizer = AutoTokenizer.from_pretrained("bert-base-cased")
result = tokenizer.tokenize("Hello!")
```

- [x] A list of strings, each string being a token

Correct! Absolutely! Convert this to IDs, and send them to a model!

- [ ] A list of IDs
- [ ] A string containing all of the tokens

Submit **You got all the answers!**

10. Is there something wrong with the following code?

```
from transformers import AutoTokenizer, AutoModel

tokenizer = AutoTokenizer.from_pretrained("bert-base-cased")
model = AutoModel.from_pretrained("gpt2")

encoded = tokenizer("Hey!", return_tensors="pt")
result = model(**encoded)
```

- [ ] No, it seems correct.
- [x] The tokenizer and model should always be from the same checkpoint.

Correct! Right!

- [ ] It's good practice to pad and truncate with the tokenizer as every input is a batch.

Submit **You got all the answers!**

Parte 3) Esse modulo foi mais focado em explicar como fazer o fine-tuning, que pega um modelo pre-treinado e especializa para a necessidade desejada. Ensina como utilizar o modulo Trainer e a biblioteca Accelerate.

1. The emotion dataset contains Twitter messages labeled with emotions. Search for it in the Hub , and read the dataset card. Which of these is not one of its basic emotions?

- [ ] Joy
- [ ] Love
- [x] Confusion

Correct! Correct! Confusion is not one of the six basic emotions.

- [ ] Surprise

Submit **You got all the answers!**

2. Search for the ar_sarcasm dataset in the Hub . Which task does it support?

- [x] Sentiment classification

Correct! That's right! You can tell thanks to the tags.

- [ ] Machine translation
- [ ] Named entity recognition
- [ ] Question answering

Submit **You got all the answers!**

3. How does the BERT model expect a pair of sentences to be processed?

- [ ] Tokens_of_sentence_1 [SEP] Tokens_of_sentence_2
- [ ] [CLS] Tokens_of_sentence_1 Tokens_of_sentence_2
- [x] [CLS] Tokens_of_sentence_1 [SEP] Tokens_of_sentence_2 [SEP]

Correct! That's correct!

- [ ] [CLS] Tokens_of_sentence_1 [SEP] Tokens_of_sentence_2

Submit **You got all the answers!**

4. What are the benefits of the Dataset.map() method?

- [x] The results of the function are cached, so it won't take any time if we re-execute the code.

Correct! That is indeed one of the neat benefits of this method! It's not the only one, though...

- [x] It can apply multiprocessing to go faster than applying the function on each element of the dataset.

Correct! This is a neat feature of this method, but it's not the only one!

- [x] It does not load the whole dataset into memory, saving the results as soon as one element is processed.

Correct! That's one advantage of this method. There are others, though!

Submit **You got all the answers!**

5. What does dynamic padding mean?

- [ ] It's when you pad the inputs for each batch to the maximum length in the whole dataset.
- [x] It's when you pad your inputs when the batch is created, to the maximum length of the sentences inside that batch.

Correct! That's correct! The "dynamic" part comes from the fact that the size of each batch is determined at the time of creation, and all your batches might have different shapes as a result.

- [ ] It's when you pad your inputs so that each sentence has the same number of tokens as the previous one in the dataset.

Submit **You got all the answers!**

6. What is the purpose of a collate function?

- [ ] It ensures all the sequences in the dataset have the same length.
- [x] It puts together all the samples in a batch.

Correct! Correct! You can pass the collate function as an argument of a `DataLoader`. We used the `DataCollatorWithPadding` function, which pads all items in a batch so they have the same length.

- [ ] It preprocesses the whole dataset.
- [ ] It truncates the sequences in the dataset.

Submit **You got all the answers!**

7. What happens when you instantiate one of the AutoModelForXxx classes with a pretrained language model (such as bert-base-uncased) that corresponds to a different task than the one for which it was trained?

- [ ] Nothing, but you get a warning.
- [x] The head of the pretrained model is discarded and a new head suitable for the task is inserted instead.

Correct! Correct. For example, when we used `AutoModelForSequenceClassification` with `bert-base-uncased`, we got warnings when instantiating the model. The pretrained head is not used for the sequence classification task, so it's discarded and a new head is instantiated with random weights.

- [ ] The head of the pretrained model is discarded.
- [ ] Nothing, since the model can still be fine-tuned for the different task.

Submit **You got all the answers!**

8. What's the purpose of TrainingArguments ?

- [x] It contains all the hyperparameters used for training and evaluation with the `Trainer`.

Correct! Correct!

- [ ] It specifies the size of the model.
- [ ] It just contains the hyperparameters used for evaluation.
- [ ] It just contains the hyperparameters used for training.

Submit **You got all the answers!**

9. Why should you use the 🤗 Accelerate library?

- [ ] It provides access to faster models.
- [ ] It provides a high-level API so I don't have to implement my own training loop.
- [x] It makes our training loops work on distributed strategies.

Correct! Correct! With 🤗 Accelerate, your training loops will work for multiple GPUs and TPUs.

- [ ] It provides more optimization functions.

Submit **You got all the answers!**

Análise de Dados com NER

3) Baixe o conjunto de dados de notícias disponível em: Folha UOL News Dataset..

Utilize o modelo https://huggingface.co/monilouise/ner_news_portuguese para identificar e extrair entidades mencionadas nas notícias.Crie um ranking das organizações que mais apareceram na seção "Mercado" no primeiro trimestre de 2015.Apresente os resultados em um relatório detalhado, incluindo a metodologia utilizada e visualizações para apoiar a análise.

Link código: posGraduacaoIA/LLM/projeto_pd_ner.ipynb at main · rachelreuters/posGraduacaoIA · GitHub

Tive algumas dificuldades com os resultados, pois a partir da tokenização, apareceram muitos resultados com caractere ### que pode indicar que a estratégia de agregação do modelo em questão não foi capaz de entender como uma palavra somente. Tentei utilizar outros métodos de aggregation_strategy como max, average e simple, em todas elas continuei obtendo resultados com ##, como pode ser visto abaixo.

Porém nota-se que Brad e ###esco era para ser uma única palavra.

Algumas outras ficam muito difíceis de entender como ##er, ##u , ##i, S, O.

Outro tratamento pós processamento foi de ignorar organização Folha, pois é o jornal que está emitindo a mensagem, então acaba que o aparecimento fica bem obvio e não gera novos insights pra a análise.

Inferências :

Setor Financeiro e Bancário (mais presente):

- Brad : Provavelmente se refere ao Bradesco.
- Itaú, Ita : Itaú Unibanco.
- BTG, Pactual : BTG Pactual.
- Unibanco : Banco que se fundiu com o Itaú.
- Santa : Provavelmente se refere ao banco Santander.
- Moody's : Uma das maiores agências de classificação de risco de crédito.
- Investimentos : A própria palavra "investimentos".
- Brasil : Banco do Brasil

Mercado de Ações e Empresas Brasileiras:

- FBovespa : se refere ao Ibovespa, o principal índice da bolsa de valores brasileira.
- BM & : Provavelmente BM&F (Bolsa de Mercadorias e Futuros), que faz parte da B3, a bolsa do Brasil.
- Vale : Uma das maiores empresas da bolsa brasileira.
- Galvão : possivelmente a Galvão Engenharia, uma construtora.

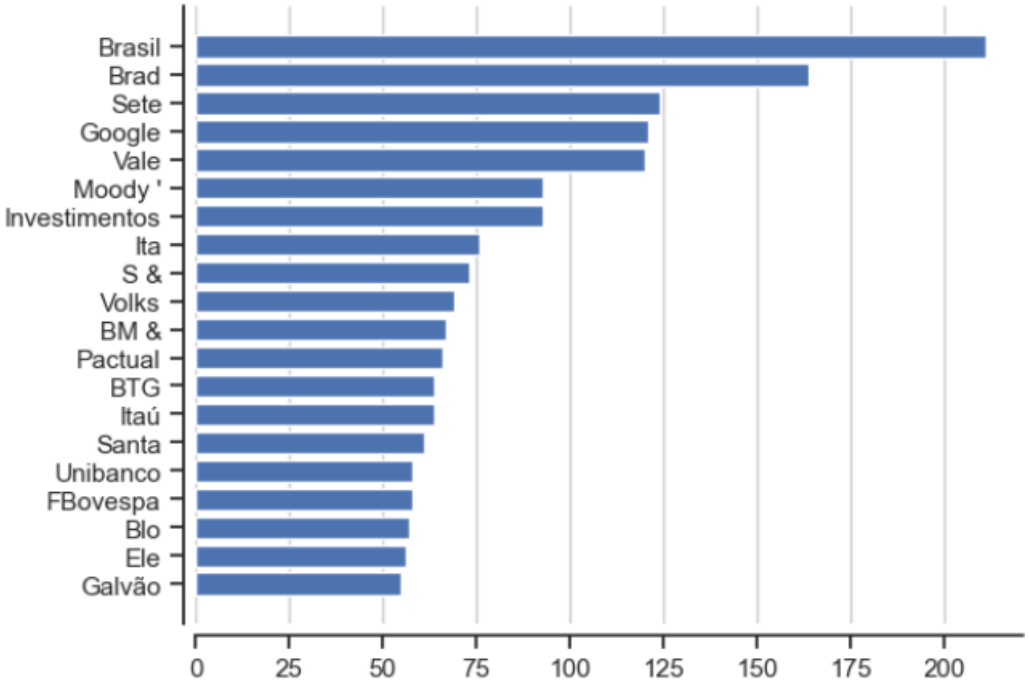
Setor Corporativo Geral:

- Google : Gigante da tecnologia.
- Volks : Volkswagen.

O texto provavelmente fala sobre o desempenho do mercado de ações brasileiro (Ibovespa), mencionando a performance das principais empresas listadas, como Vale, e o papel dos grandes bancos (Bradesco, Itaú, BTG Pactual). Também pode ser um relatório de uma corretora avaliando o cenário de investimentos no Brasil, recomendando ações e discutindo o risco-país (mencionado pela presença da Moody's).

	word	entity_group	score	start	end	count
2157	Folha	ORG	0.649513	16	21	607
3006	O	ORG	0.955168	988	989	372
1513	Brasil	ORG	0.884909	1039	1045	211
4081	s	ORG	0.903501	91	92	185
517	##i	ORG	0.622377	989	990	168
1505	Brad	ORG	0.955614	994	998	164
429	##esco	ORG	0.859656	998	1002	160
1354	B	ORG	0.708539	757	758	152
180	##S	ORG	0.471622	131	132	129
3472	Sete	ORG	0.984839	535	539	124
31	##BC	ORG	0.802392	132	134	123
2267	Google	ORG	0.975814	1060	1066	121
3837	Vale	ORG	0.934411	1821	1825	120
2409	Ib	ORG	0.625421	1238	1240	119
3079	P	ORG	0.943618	993	994	113
90	##G	ORG	0.735299	759	760	111
3649	T	ORG	0.966086	916	917	109
2311	H	ORG	0.954237	1204	1205	101
2888	Moody '	ORG	0.912413	718	724	93
2514	Investimentos	ORG	0.970368	1849	1862	93

Realizando um filtro para remover esses itens que não são facilmente compreensíveis:



Engenharia de Prompts

4) Analise os seguintes prompts e identifique por que eles poderiam gerar respostas insatisfatórias ou irrelevantes. Reformule cada prompt utilizando técnicas de engenharia de prompts para torná-los mais específicos e direcionados. Explique as melhorias feitas em cada caso e os motivos por trás das reformulações:

- Exemplo 1: "Escreva sobre cachorros."

  Esse prompt é extremamente genérico. Não se sabe exatamente o que o usuário deseja, que tipo de informação. A resposta disso seria algo bem genérico igualmente. Utilizando um método de engenharia de prompt “Self-Ask” : "Eu quero que você escreva sobre cachorros. Tentando responder algumas perguntas:
  Quais espécies são as mais queridas pelos humanos?
  Quais espécies são as mais raras de se encontrar?
  Porque cachorros são considerados os melhores amigos do homem?
  Depois de responder a essas perguntas, use as respostas para gerar um artigo bem estruturado sobre cachorros."
  Decompõe em perguntas, delimitando a resposta gerada, além disso guia o “raciocínio” do algoritmo.

- Exemplo 2: "Explique física."

  Assim como o anterior, é genérico. Física é um conceito muito amplo que pode abordar uma infinidade de coisas. Utilizando um método de engenharia de prompt “Zero-shot” : “Crie uma tabela comparativa que explique as principais diferenças entre a Física Clássica e a Física Quântica. A tabela deve ter três colunas: 'Conceito', 'Física Clássica' e 'Física Quântica'. Compare pelo menos três pontos fundamentais, como 'Escala de Aplicação' (macroscópico vs. subatômico), 'Natureza do Resultado' (determinístico vs. probabilístico) e 'Visão sobre a Energia'.” Com isso, foi definido o formato, criou um tipo de limite de conteúdo e forneceu critérios.

5) O prompt "Descreva a história da internet." foi mal formulado. Aplique técnicas de engenharia de prompts para melhorá-lo. Reformule o prompt para melhorar a especificidade e a qualidade da resposta.

Justifique as mudanças feitas e explique como elas contribuem para obter uma resposta mais eficaz e relevante.

Utilizando a técnica "Least-To-Most" dividindo o problema em uma série de problemas mais simples e guiar o modelo para resolv Descreva a história da internet em etapas, do conceito mais simples ao mais complexo. Por favor, responda cada pergunta em ordem.

1. A Origem Militar: foque no seu propósito original durante a Guerra Fria e por que sua arquitetura era tão inovadora.

2. A Transição para o Meio Acadêmico: descreva como essa tecnologia evoluiu. Explique como foi adaptada a rede para conectar universidades e centros de pesquisa.

3. A Explosão para o Público: cite duas inovações que tornaram a internet acessível ao público geral nos anos 90, explicando qual foi o ponto de virada.

4. A Conclusão: Resuma como a base militar, a expansão acadêmica e a revolução de usabilidade da WWW e dos navegadores se combinaram para formar a internet que conhecemos hoje.er em sequência.

Essa técnica dividiu a história da internet em fases lógicas, tornando as perguntas mais gerenciáveis. Cada etapa construindo um contexto progressivo. Isso melhora a qualidade e profundidade da resposta, evitando explicações superficiais e permitindo uma narrativa estruturada. Também fica mais fácil controlar os tópicos mais relevantes de acordo com a necessidade final.

6) Aplique a técnica de Chain of Thought (CoT) para melhorar o prompt "Explique como funciona a energia solar.", detalhando o raciocínio necessário para que o modelo forneça uma resposta completa e coerente. Explique como a aplicação da técnica CoT melhora a resposta do modelo.

"Explique como funciona a energia solar, seguindo esta cadeia de pensamento:

O Princípio Físico: explique como funciona a corrente elétrica, os elétrons e fótons.

Explique como são feitos os painéis solares

Aplicando o princípio físico nos painéis solares: como os painéis solares conseguem interagir com os fótons e como que funciona a liberação de energia nesses painéis.

Por fim, como que a energia é transmitida para o consumidor final depois de passar dos painéis solares."

Utilizando a CoT, é possível garantir que todos os estágios do processo sejam abordados, desde a física fundamental até a aplicação prática em uma residência, resultando em uma resposta muito mais completa.

Projeto Prático com Streamlit, LLM e LangChain

7) Escolha uma aplicação para desenvolver utilizando Streamlit, LLM e LangChain. Crie um aplicativo interativo que demonstre o uso de LLMs para resolver um problema específico.

Descreva a aplicação escolhida e os objetivos principais do projeto.

Explique a arquitetura do aplicativo, incluindo como o Streamlit, LLM e LangChain são utilizados.

Implemente o aplicativo e forneça o código-fonte, junto com instruções para execução.

Apresente evidências e exemplos de uso do aplicativo e discuta os resultados obtidos.

Objetivo:

Essa aplicação tem como objetivo gerar dicas de perfumes para um usuário comprar que sejam similares ao perfume original que ele optou. Esse aplicativo pode ser útil para algum comercio que deseja ofertar perfumes de acordo com o gosto do cliente.

Como utilizar:

Primeiramente o usuário tem que escrever no campo de texto o nome do perfume que ele deseja usar como referência. Em seguida selecionar o botão “Encontrar Perfumes similares”.

Como funciona:

O prompt irá receber um nome de perfume através do Streamlit. Em seguida a Langchain vai ser acionada com o primeiro passo do AgentExecutor, que irá executar todas as tarefas em cadeia até chegar a resposta final que irá retornar um json internamente e o Streamlit se encarrega de exibir os dados corretamente como uma grid.

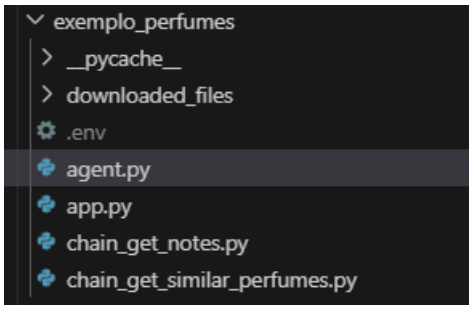
O AgentExecutor possui 3 tools:

- GetNotesTemplate: LLMChain cujo objetivo é retornar o tipo e as notas do perfume que o usuário colocou como input. Seu output é uma lista de strings com tipo e notas separadas por virgula.
- GetSimilarPerfumesTemplate: LLMChain cujo objetivo é descobrir 5 perfumes similares ao que o usuário colocou como input a partir das suas notas e seu tipo. O output dessa tool é uma lista de nomes de perfumes com o nome da marca.
- find_ml_prices : Essa é uma tool simples que irá fazer o Web Scraping no mercado livre para indicar os produtos desejados, junto ao link, preço e também a foto do produto. O output dessa etapa e uma lista de dicionários contendo link, preço, imagem e nome do produto.

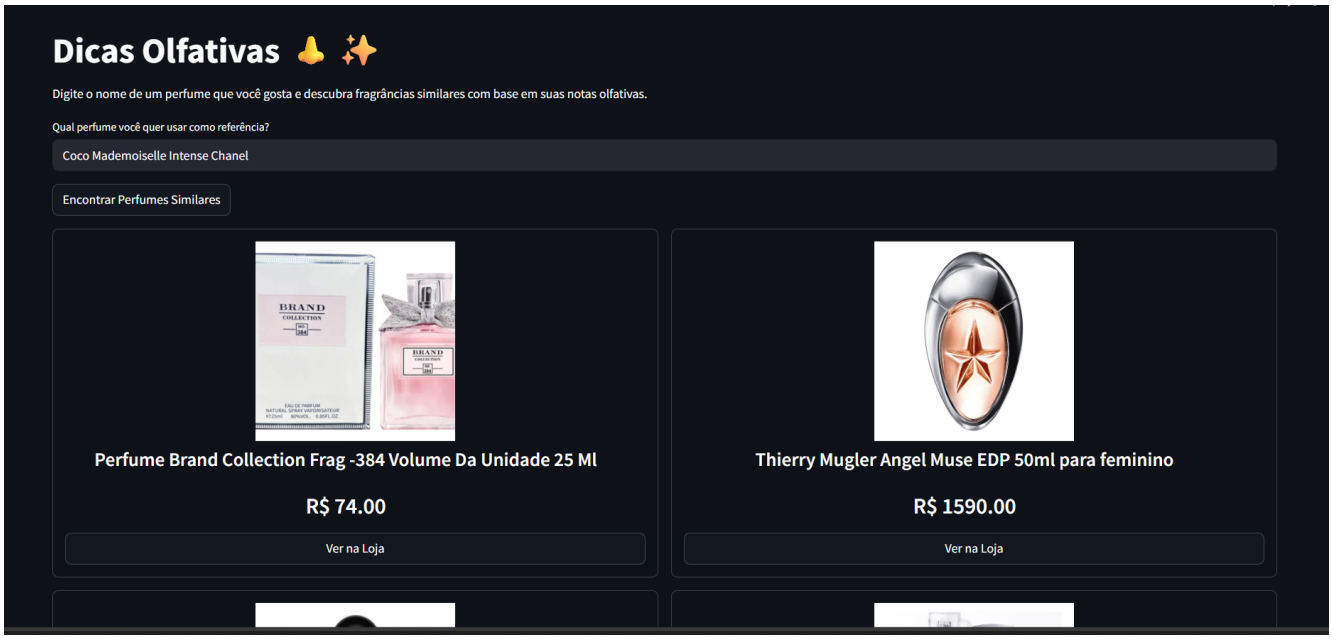
Como executar:

```
streamlit run app.py
```

Arquivos:



Streamlit:



Link do app no Github:

posGraduacaoIA/LLM/exemplo_perfumes at main · rachelreuters/posGraduacaoIA · GitHub