

Projeto Final IA generativa para linguagem

Aqui temos o readme do projeto e o link este mesmo readme no github: https://github.com/ana-delfino/infnet_gen_ai

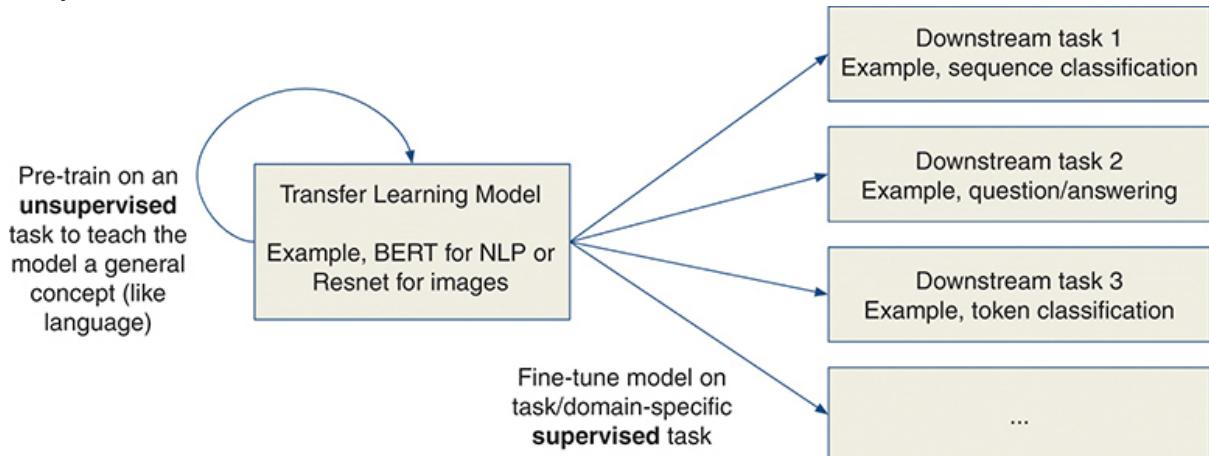
Índice

- [Parte 1 - Fundamentos das LLMs](#)
- [Parte 2 - Quizzes do Curso de NLP da Hugging Face](#)
- [Parte 3 - Análise de Dados com NER](#)
- [Parte 4 - Engenharia de Prompts](#)
- [Parte 5 - Projeto Prático com Streamlit, LLM e LangChain](#)

Parte 1 - Fundamentos das LLMs

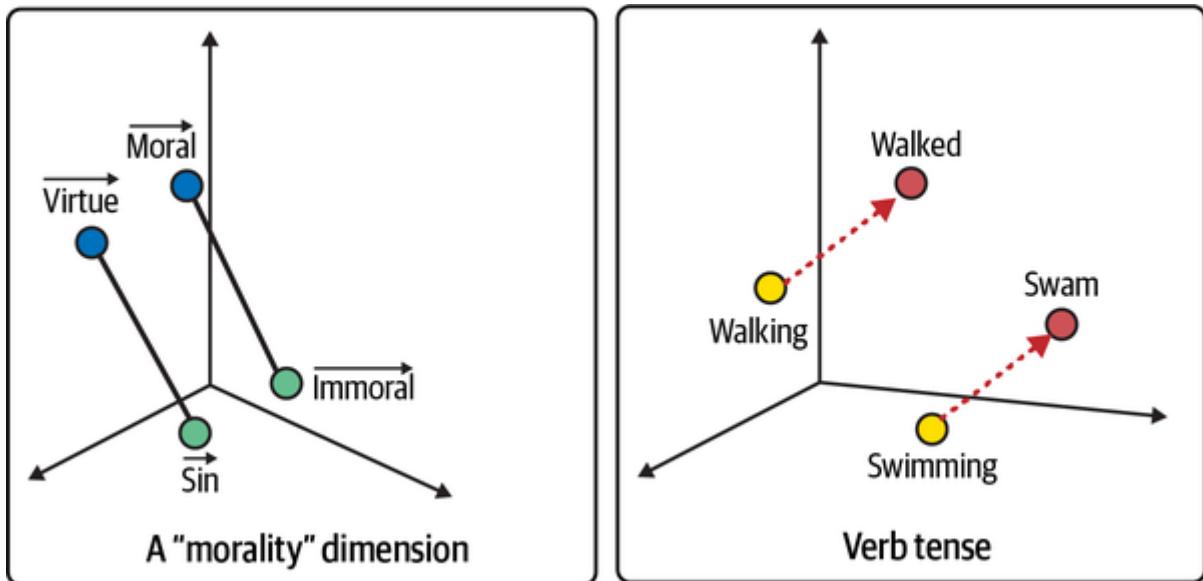
- Questão 1) Explique os seguintes conceitos fundamentais dos LLMs, fornecendo exemplos práticos e diagramas onde for relevante:

- **Pre-training:** Todos os modelos de LLM são treinados em grandes volumes de dados. Durante o pré-treino os LLMs tentam aprender e generalizar as relações entre as palavras do corpus, o desempenho dos LLMs nas diversas tarefas dependem desse dataset usado no pré-treino.
- **Transfer Learning:** Transfer learning é uma técnica usada para aproveitar o conhecimento de um modelo pré-treinado, ganho em uma tarefa, para melhorar a performance em outra tarefa similar. O transfer learning permite que a rede use o conhecimento adquirido na tarefa original para realizar outra tarefa usando muito menos dados do que seria necessário para treinar um modelo específico do zero e também permite utilizar o conhecimento já adquirido para gerar embeddings. Em termos de arquitetura, isso envolve separar o modelo em body e head, onde a head é a tarefa específica da rede. Durante o treinamento, os pesos do body aprendem características gerais do domínio de origem, e esses pesos são usados para inicializar um novo modelo para a nova tarefa ou para . A figura abaixo mostra representar essa relação:



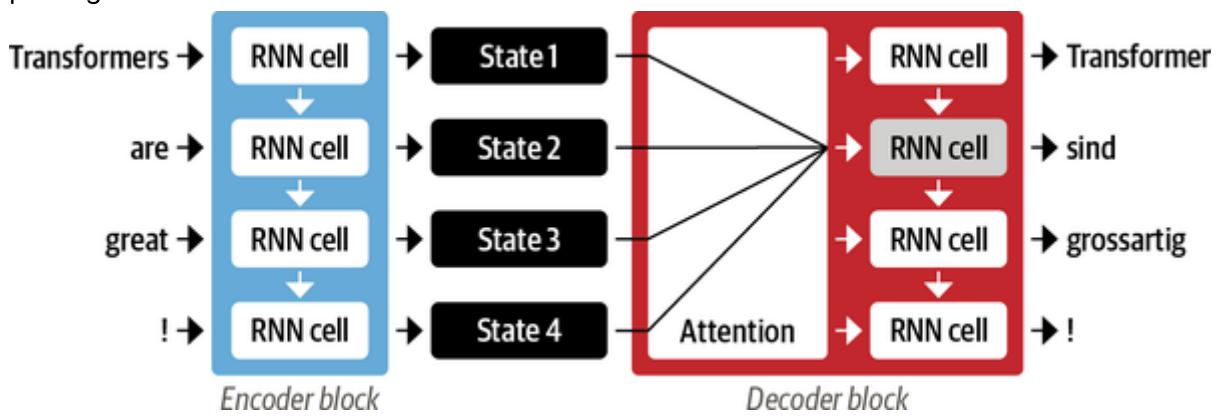
- **Embeddings:** Antes de explicar o que são embedding primeiro é necessário entender o que é um token. O token é a representação de um sentença, palavra ou subpalavra. O processo de

tokenização envolve a geração dos tokens. A representação numérica dos tokens são os vetores. Esses vetores são multidimensional (matrizes) de números que capturam a relações sintéticas (função gramatical dentro da linguagem) e semânticas (significado dentro da linguagem) entre as palavras do corpus. Esse vetores de palavras são chamados de embeddings. Durante o treinamento os modelos são desenhados para identificar e aprender esses padrões, assegurando que as palavras com significados similares sejam mapeadas próximas no espaço de alta dimensão. Além disso, há vários tipos de embeddings são possíveis, incluindo embeddings de posição, que codificam a posição de um token em uma frase, e embeddings de token, que codificam o significado semântico de um token. Exemplos na imagem abaixo:



- **Transformers**: É uma arquitetura de rede neural profunda. Foi criado para resolver o problema que as redes neurais até então existentes não conseguiram, já os modelos de Redes Neurais Recorrentes e suas variantes processam as palavras sequencialmente e tem dificuldade de capturar as relações de longa distância (contexto). Os modelos com arquitetura tipo transformers são capazes de aprender contexto e tem 2 componentes principais:
 - *Encoder (codificador)*: tem a tarefa de receber texto bruto, dividi-lo em seus componentes principais, convertendo esses componentes em vetores e usar o mecanismo de atenção para entender o contexto do texto.
 - *Decoder (decodificador)*: modifica a atenção para gerar a sequência de saída do modelo considerando o saída do codificador. Um dos diferenciais do modelo tranformers é uso um tipo especial de attention chamado self attention para permitir que cada palavra em uma sequência “dê atenção” (procure o contexto) todas as outras palavras na sequência, permitindo capturar dependências de longo alcance e relacionamentos contextuais entre palavras.
- **Attention**: Attention é um dos componentes principais da arquitetura das redes neurais do tipo transformer. A ideia por trás da attention é que ao invés de produzir uma único hidden state para um sequência de input o encoder fornece um hidden state para cada step que o decoder pode acessar. Ao usar todos os steps ao mesmo tempo a attention permite criar um input grande para o decoder, então o mecanismo de attention é usado para priorizar quais states serão usados. Isso acontece pois a attention faz com que o decoder receba diferentes pesos ou atenção para cada encoder state em cada step do decoding. O processo é ilustrado

pela figura abaixo



- **Fine-Tuning:** Uma vez que o LLM foi treinado, podemos realizar o fine tuning que consiste em treinar o LLM em um dataset menor que o dataset usado no treino do LLM para o o modelo faça uma atividade específica baseada nesse novos dados. Isso permite que o LLM utilize o conhecimento do pre-treino para melhorar a performance na tarefa específica

Fontes: Lewis Tunstall, Leandro von Werra, Thomas Wolf - Natural Language Processing with Transformers James Phoenix, Mike Taylor - Prompt Engineering for Generative AI Sinan Ozdemir - Quick Start Guide to Large Language Models: Strategies and Best Practices for Using ChatGPT and Other LLMs

Parte 2 - Quizzes do Curso de NLP da Hugging Face

- Questão 2) Acesse os quizzes dos capítulos 1, 2 e 3 do curso de NLP da Hugging Face através do link: [Curso de NLP](#).
 - 2.1) Resolva os quizzes e capture screenshots dos resultados.
 - 2.2) Anexe as screenshots a esta avaliação e explique brevemente os conceitos abordados em cada quiz.

Resposta: O **capítulo 1** fala sobre transformers NLP e LLM, explicando a diferença entre os dois tipos de modelos, as principais tarefas realizadas pelos modelos transformers, como eles funcionam, as arquiteturas, encode, decode e seq2seq. prints do capítulo estão abaixo:

capítulo_1/Captura-de-Tela-2025-06-05-às-18.45.45.png

The screenshot shows a dark-themed user interface for the Hugging Face LLM Course. On the left, there's a sidebar with navigation links for chapters 0 through 6. Chapter 0 is titled '0. SETUP'. Chapter 1 is titled '1. TRANSFORMER MODELS' and contains sections like 'Introduction', 'Natural Language Processing and Large Language Models', 'Transformers, what can they do?', 'How do Transformers work?', 'Solving Tasks with Transformers', and 'Transformer Architectures'. Chapter 2 is titled '2. USING 🤗 TRANSFORMERS'. Chapter 3 is titled '3. FINE-TUNING A PRETRAINED MODEL'. Chapter 4 is titled '4. SHARING MODELS AND TOKENIZERS'. Chapter 5 is titled '5. THE 🤗 DATASETS LIBRARY'. Chapter 6 is titled '6. THE 🤗 TOKENIZERS LIBRARY'. A 'Quick quiz' section is highlighted with a dark background.

1. Explore the Hub and look for the roberta-large-mnli checkpoint. What task does it perform?

Summarization
 Text classification

Correct! More precisely, it classifies if two sentences are logically linked across three labels (contradiction, neutral, entailment) — a task also called *natural language inference*.

Text generation

Submit You got all the answers!

2. What will the following code return?

```
from transformers import pipeline
ner = pipeline("ner", grouped_entities=True)
ner("My name is Sylvain and I work at Hugging Face in Brooklyn.")
```

It will return classification scores for this sentence, with labels "positive" or "negative".
 It will return a generated text completing this sentence.
 It will return the words representing persons, organizations or locations.

Correct! Furthermore, with `grouped_entities=True`, it will group together the words belonging to the same entity, like "Hugging Face".

Submit You got all the answers!

capítulo_1/Captura-de-Tela-2025-06-05-às-18.50.21.png

The screenshot shows a user interface for an LLM Course. On the left, there's a sidebar with navigation links for chapters 0 through 8. Chapter 3 is currently selected, indicated by a blue background. The main content area contains two sections: '3. What should replace ... in this code sample?' and '4. Why will this code fail?'. Each section has a code snippet, a list of multiple-choice questions, and a feedback message.

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.
 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")
```

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!

capítulo_1/Captura-de-Tela-2025-06-05-às-18.51.45.png

The screenshot shows a dark-themed user interface for an LLM Course. On the left, a sidebar lists sections: 0. SETUP, 1. TRANSFORMER MODELS (Introduction, Natural Language Processing and Large Language Models, Transformers, what can they do?, How do Transformers work?, Solving Tasks with Transformers, Transformer Architectures), 2. USING 🤗 TRANSFORMERS, 3. FINE-TUNING A PRETRAINED MODEL, and 4. SHARING MODELS AND TOKENIZERS. A "Quick quiz" button is highlighted. The main content area displays a question: "5. What does ‘transfer learning’ mean?". It includes two options: a radio button followed by the text "Transferring the knowledge of a pretrained model to a new model by training it on the same dataset." and a checked checkbox followed by the text "Transferring the knowledge of a pretrained model to a new model by initializing the second model with the first model's weights.". Below the question is a green box containing the text "Correct! When the second model is trained on a new task, it *transfers* the knowledge of the first model.". Another question follows: "6. True or false? A language model usually does not need labels for its pretraining." It has two options: a checked checkbox followed by "True" and a radio button followed by "False". A green box below the question states "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.)". At the bottom right of the main content area, there is a "Submit" button and the text "You got all the answers!".

capítulo_1/Captura-de-Tela-2025-06-05-às-19.02.52.png

The screenshot shows a dark-themed user interface for an LLM Course. On the left, a sidebar lists sections: 0. SETUP, 1. TRANSFORMER MODELS (Introduction, Natural Language Processing and Large Language Models, Transformers, what can they do?, How do Transformers work?, Solving Tasks with Transformers, Transformer Architectures), 2. USING 🤗 TRANSFORMERS, 3. FINE-TUNING A PRETRAINED MODEL, and 4. SHARING MODELS AND TOKENIZERS. A "Quick quiz" button is highlighted. The main content area displays a question: "7. Select the sentence that best describes the terms ‘model’, ‘architecture’, and ‘weights’.". It includes three options: a radio button followed by "If a model is a building, its architecture is the blueprint and the weights are the people living inside.", a radio button followed by "An architecture is a map to build a model and its weights are the cities represented on the map.", and a checked checkbox followed by "An architecture is a succession of mathematical functions to build a model and its weights are those functions parameters.". Below the question is a green box containing the text "Correct! The same set of mathematical functions (architecture) can be used to build different models by using different parameters (weights.)". Another question follows: "8. Which of these types of models would you use for completing prompts with generated text?". It includes three options: a radio button followed by "An encoder model", a checked checkbox followed by "A decoder model", and a radio button followed by "A sequence-to-sequence model". Below the question is a green box containing the text "Correct! Decoder models are perfectly suited for text generation from a prompt.". At the bottom right of the main content area, there is a "Submit" button and the text "You got all the answers!". To the right of the main content area, there is a sidebar titled "Ungraded quiz" with numbered questions 1 through 10, each with a brief description.

capítulo_1/Captura-de-Tela-2025-06-05-às-19.06.53.png

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

- An encoder model
- A decoder model
- 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?**

- 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?**

- 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.

- The data the model was trained on is biased.

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

- 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!

O **capítulo 2** é mais prático e explica como usar os modelos transformers destacando que os modelos não precessam textos diretamente, então o primeiro passo é converter os textos em números utilizando o tokenizar - cujo papel é separar os inputs em palavras, caracteres, pontuação e subpalavras gerando os tokens, cada token é mapeado para um inteiro. prints do capítulo estão abaixo:

capitulo_2/Captura-de-Tela-2025-06-05-às-20.43.23.png

1. What is the order of the language modeling pipeline?

- First, the model, which handles text and returns raw predictions. The tokenizer then makes sense of these predictions and converts them back to text when needed.
- First, the tokenizer, which handles text and returns IDs. The model handles these IDs and outputs a prediction, which can be some text.
- The tokenizer handles text and returns IDs. The model handles these IDs and outputs a prediction. The tokenizer can then be used once again to convert these predictions back to some text.

Correct! Correct! The tokenizer can be used for both tokenizing and de-tokenizing.

Submit

You got all the answers!

2. How many dimensions does the tensor output by the base Transformer model have, and what are they?

- 2: The sequence length and the batch size
- 2: The sequence length and the hidden size
- 3: The sequence length, the batch size, and the hidden size

Correct! Correct!

Submit

You got all the answers!

capitulo_2/Captura-de-Tela-2025-06-05-às-20.43.56.png

5. What is an AutoModel?

- A model that automatically trains on your data
- 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?

- 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
- 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?

- 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!

capítulo_2/Captura-de-Tela-2025-06-05-às-20.44.58.png

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.
- 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.

- 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
- 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
```

capítulo_2/Captura-de-Tela-2025-06-05-às-20.45.17.png

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!")
```

- 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.
- 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.

O **capítulo 3** tem foco no fine tuning de modelos pré-treinados, no capítulo há explicações de como usar seu dados para fazer o fine tuning de modelos. prints do capítulo estão abaixo:

capítulo_3/Captura-de-Tela-2025-06-05-às-21.00.38.png

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
- 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?

- 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
- [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!

capítulo_3/Captura-de-Tela-2025-06-05-às-21.00.57.png**4. What are the benefits of the Dataset.map() method?**

- 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...

- 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!

- 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.
- 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.
- 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!**

capitulo_3/Captura-de-Tela-2025-06-05-às-21.01.13.png

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.
- 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 ?

- 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.
- 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.

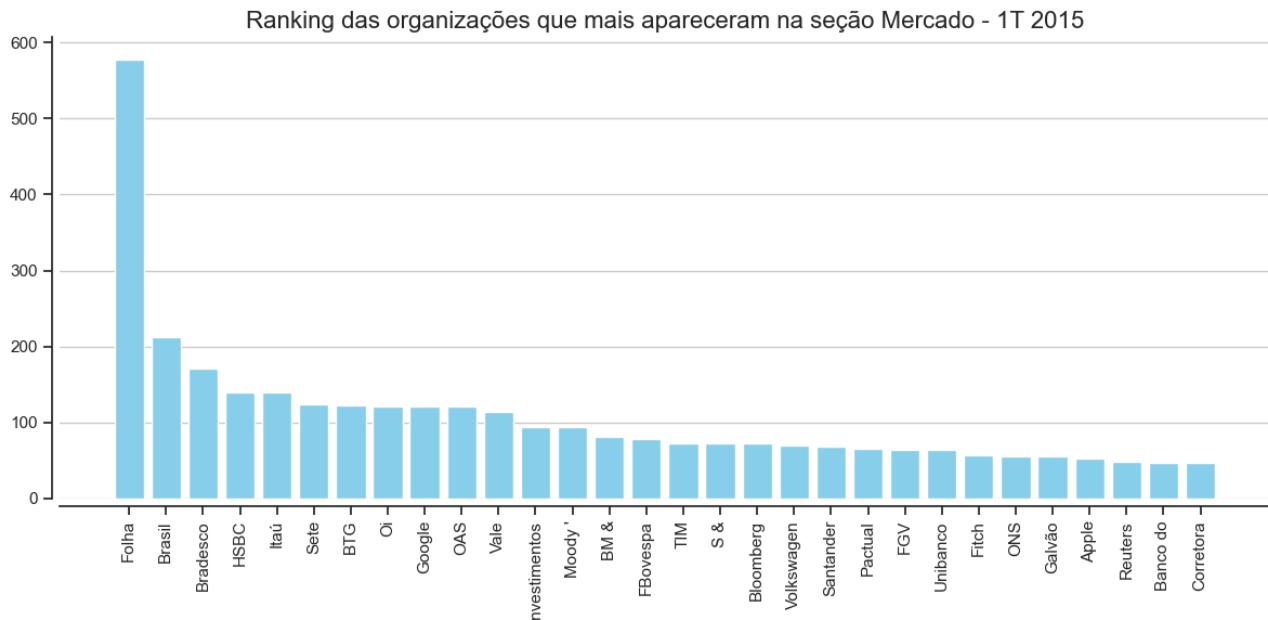
Submit You got all the answers!

Parte 3 - Análise de Dados com NER

- Questão 3) Baixe o conjunto de dados de notícias disponível em: Folha UOL News Dataset.
- o Código está disponível no github: https://github.com/ana-delfino/infnet_gen_ai/blob/main/ana_delfino_gen_ai_pd.ipynb
- 3.1) Utilize o modelo 'monilouise/ner_pt_br' para identificar e extrair entidades mencionadas nas notícias.

Resposta: O código foi desenvolvido no notebook: ana_delfino_gen_ai_pd.ipynb

- 3.2) Crie um ranking das organizações que mais apareceram na seção "Mercado" no primeiro trimestre de 2015.



- 3.3) Apresente os resultados em um relatório detalhado, incluindo a metodologia utilizada e visualizações para apoiar a análise.

Resposta: O dataset tem 2111 textos de mercado no primeiro trimestre de 2015. A estratégia de agregação dos tokens foi a max, outras como simples foram testadas mas produziam muitas palavras cortadas, dificultando a identificação da organização. Além disso, para entender as organizações que estão nos textos considerei apenas as palavras com mais de 1 caracter para evitar empresas que não conseguíramos interpretar.

Conforme imagem abaixo é possível ver que a organização mais citada a folha, pois as notícias são do folha news em seguida vemos a organização "Brasil" temos Brasil sendo muito citado dado as notícias são do Brasil e a palavra representa tanto o país como o banco do brasil e como uma instituição que faz ações que afetam o mercado. Em seguida vemos um grupo de bancos Bradesco, HSBC, Itaú . Também observamos a palavra 'sete' como organização isso acontece porquê o dataset parece ser um erro na coluna de data de publicação, notícias de setembro aparecem como se fossem do primeiro trimestre de 2015. Isso não foi corrigido devido a ser um exercício, se fosse na vida real teríamos que verificar e tratar. No geral as organizações identificadas pelo modelo fazem sentido para a seção selecionada.

Parte 4 - Engenharia de Prompts

- Questão 4) Analise os seguintes prompts e identifique por que eles poderiam gerar respostas insatisfatórias ou irrelevantes:

Exemplo 1: "Escreva sobre cachorros." Exemplo 2: "Explique física."

Subquestões:

- 4.1) Reformule cada prompt utilizando técnicas de engenharia de prompts para torná-los mais específicos e direcionados. Exemplo 1 - reformulado: Aja como um veterinário atendendo uma família com idosos e crianças Dê sugestões de 2 raça de cachorros que seja adequada para a família. Explica 2 motivos para escolha das raças. Dê dicas de como cuidar do cachorro

Exemplo 2 - reformulado: Aja como um professor de física para alunos crianças de 12 anos. Use 2 a 3 frases para explicar o que é física estuda. Dê 2 exemplos curtos da aplicação física no dia a dia.

- 4.2) Explique as melhorias feitas em cada caso e os motivos por trás das reformulações.

Resposta: As reformulações foram necessárias para tornar o prompt mais específico e com objetivo claro, os prompts originais estavão muito genérico. Para melhorar o prompt foi aplicado Chain of Thoughts (CoT), explicitando os usuários da informação e a persona do LLM.

- Questão 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. **Resposta:** O prompt abaixo direciona o modelo a refinar a quantidade de informações que devem ser trazidas ao usuário, que tipo de persona deve adotar na escrita e fornecendo o tipo de estrutura que a informação deve ser escrita. PROMPT: Haja como um aluno do ensino médio. Escreva um texto de no máximo 400 palavras sobre a história da internet e seu impacto da vida do ser humano. O texto deve ser estruturado ter Introdução, desenvolvimento, conclusão. Escolha um título adequado. Não cometa erros gramaticais.
- Questão 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. **Resposta:** A técnica de Chain of Thought (CoT) ajuda o prompt a fornecer uma resposta em subtarefas fazendo com que o modelo seja mais assertivo na resposta. PROMPT: Haja como vendedor de placas solares e explique como funciona a energia solar, abordando os seguintes pontos: Defina o que é energia solar. Cite os princípios básicos de funcionamento dos painéis solares, incluindo o efeito fotovoltaico. Explique como os painéis solares convertem a luz solar em eletricidade. Dê exemplos de uso da energia solar nas residências e empresas Descreva os benefícios e as limitações da energia solar em comparação com outras fontes de energia.

Projeto Prático com Streamlit, LLM e LangChain

- Questão 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.

Exemplos de Aplicação:

- Sumarizador de Artigos: Desenvolva um aplicativo que permita ao usuário inserir o texto de um artigo e obter um resumo conciso do conteúdo.
- Sistema de Perguntas e Respostas: Crie um sistema que permita ao usuário fazer perguntas sobre um tópico específico e receba respostas precisas e relevantes.
- Agente de Viagem: Desenvolva um agente virtual que possa ajudar usuários a planejar suas viagens, fornecendo informações sobre destinos, itinerários, e dicas de viagem.
- App de Auxílio em Aprendizagem: Crie um aplicativo que auxilie estudantes a aprender um novo assunto, fornecendo explicações, exemplos e quizzes interativos.

Subquestões:

- 7.1) Descreva a aplicação escolhida e os objetivos principais do projeto.

Resposta: O agente desenvolvido é um Real State que ajuda você conhecer melhor bairros que você deseja morar dando dicas de lugares e um breve opnião sobre o bairro.

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

Resposta: O agente foi criado usando a biblioteca Langchain, o modelo gemini e seguindo a estrutura template, foi criado um template (NeighborhoodTemplate) para buscar informações sobre o bairros e listar os 5 lugares interessantes , nesse template foi utilizado o one shot para dar um exemplo de resposta. Além disso, foi usado o template de mapping para mapear os endereços em coordenadas para serem colocadas no mapa. O streamlit é usado como interface para o agente.

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

Resposta: https://github.com/ana-delfino/infnet_gen_ai/tree/main/src Instruções para execução:

1. Clone o repositório do git
2. Crie o ambiente virtual, ative e instale os requirements do arquivo requirements.txt
3. Para executar a aplicação rode o comando abaixo:

```
cd src/  
streamlit run app.py
```

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

Resposta: Pelas imagens abaixo é possível ver que o app está funcionando bem, mesmo sem especificar o endereço, o modelo consegue fazer o que foi programado para executar.

Exemplo 1: Bairro Aclimação - Dando endereço específico

Chrome Arquivo Editar Visualização Histórico Favoritos Perfis Guia Janela Ajuda

localhost:8501

Deploy

Welcome to the Real State Agent App!

This app will help you find a good place live with detailed tips and information about the neighborhood.

Where would you like to live? Please describe your request in Portuguese.

Eu gostaria de morar na rua maracai 213 aclimacao sao paulo -sp. O que tem de legal nesse bairro?

Get neighborhood information

O bairro da Aclimação é conhecido por sua excelente qualidade de vida, oferecendo uma combinação de áreas verdes, conveniência e tranquilidade. É um local ideal para famílias e pessoas que buscam um ambiente mais calmo sem abrir mão da proximidade com o centro da cidade.

Aqui estão alguns lugares interessantes perto da Rua Maracai, 213:

- Parque da Aclimação: Comece explorando o famoso Parque da Aclimação, um dos principais atrativos do bairro. Perfeito para caminhadas, exercícios ao ar livre e momentos de relaxamento em meio à natureza. Endereço: Rua Muniz de Sousa, 1119 - Aclimação, São Paulo - SP

Exemplo 2: Cracolandia - Sem dar endereço específico

Chrome Arquivo Editar Visualização Histórico Favoritos Perfis Guia Janela Ajuda

localhost:8501

Deploy

Welcome to the Real State Agent App!

This app will help you find a good place live with detailed tips and information about the neighborhood.

Where would you like to live? Please describe your request in Portuguese.

Morar no bairro da cracolandia em SP é uma boa?

Get neighborhood information

O bairro da Cracolândia, em São Paulo, é uma região com muitos desafios sociais e urbanos. Embora existam projetos de revitalização em andamento, a área ainda enfrenta problemas como criminalidade e falta de infraestrutura. Morar lá pode ser difícil devido a essas questões, mas também pode oferecer acesso a oportunidades de baixo custo e uma comunidade vibrante.

Aqui estão alguns lugares interessantes próximos à Cracolândia:

- Pinacoteca do Estado de São Paulo: Um importante museu com obras de arte brasileiras dos séculos XIX e XX. Endereço: Praça da Luz, 2 - Luz, São Paulo - SP, 01020-020

Exemplo 3: Bairro Vila Madalena - Apenas mencionando o bairro.

Chrome Arquivo Editar Visualização Histórico Favoritos Perfis Guia Janela Ajuda

localhost:8501

Deploy :

Welcome to the Real State Agent App!

This app will help you find a good place live with detailed tips and information about the neighborhood.

Where would you like to live? Please describe your request in Portuguese.

Gostaria de morar na vila madalena em São Paulo. É uma boa ideia?

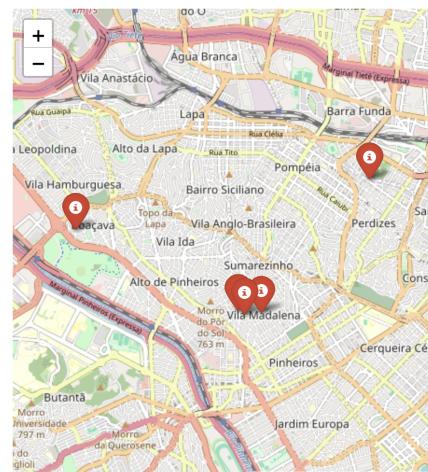
Get neighborhood information

O bairro é conhecido por sua vida noturna vibrante, cena artística e diversidade cultural. É uma área boêmia com muitos bares, restaurantes, galerias de arte e lojas descoladas.

Aqui estão alguns pontos de interesse próximos na Vila Madalena:

- **Beco do Batman:** Um famoso ponto turístico com murais de grafite coloridos e arte de rua. Endereço: R. Gonçalo Afonso - Vila Madalena, São Paulo - SP, 05436-010
- **Instituto Tomie Ohtake:** Um centro cultural com exposições de arte contemporânea e design. Endereço: R. Coropés, 88 - Pinheiros, São Paulo - SP, 05426-010

Feira da Vila Madalena - Uma feira de rua com artesanato, roupas, acessórios e



19 / 19