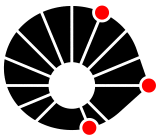

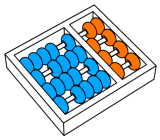

Avaliação de Modelos de Inteligência Artificial Multimodais Abertos em Vestibulares Brasileiros

Guilherme Zeferino Rodrigues Dobins
Rodrigo Frassetto Nogueira *Hélio Pedrini*

Relatório Técnico - IC-PFG-23-34
Projeto Final de Graduação
2023 - Dezembro

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

Avaliação de Modelos de Inteligência Artificial Multimodais Abertos em Vestibulares Brasileiros

Guilherme Zeferino Rodrigues Dobins* Rodrigo Frassetto Nogueira†
Hélio Pedrini*

Dezembro de 2023

Resumo

Este estudo investiga a eficácia de modelos de aprendizado de máquina multimodais abertos (*open-source*) na resolução de questões de vestibulares e exames nacionais brasileiros, utilizando um subconjunto do conjunto de dados BlueX [1] que combina texto e imagem. O foco é avaliar comparativamente o desempenho de modelos como OpenFlamingo [2], LLaVA 1.5 [3] e CogVLM [5], e analisar como estes se alinham com resultados em bases de dados reconhecidas, além de compará-los com modelos puramente textuais no mesmo domínio de questões. Um aspecto chave deste trabalho é um estudo de ablação com o Vicuna [13], um modelo baseado apenas em texto, para entender o impacto da multimodalidade nos resultados. Este estudo destaca a relevância da integração de informações textuais e visuais em modelos de inteligência artificial (IA), facilitando a compreensão sobre a evolução dos modelos de aprendizado de máquina multimodais, e sublinha a influência da multimodalidade em tarefas de VQA (*visual question answering*) que envolvem componentes textuais significativas.

1 Introdução

Nos últimos anos, os campos de aprendizado de máquina e de inteligência artificial têm testemunhado avanços notáveis e interesse público que extrapola as barreiras da comunidade da computação. Um dos primeiros avanços a chamar atenção para a área foi no domínio da visão computacional, impulsionado pela disponibilidade de amplas bases de dados, como a ImageNet [8]. Esse período se caracterizou pelo desenvolvimento de modelos avançados que detectam e identificam objetos em imagens e vídeos, culminando na criação de técnicas altamente eficazes.

O avanço de interesse público subsequente foi estimulado pela arquitetura de Transformers [24], impulsionando um significativo crescimento no processamento de linguagem natural (PLN). Um marco neste desenvolvimento foi o ChatGPT, que popularizou ainda mais a IA generativa. Após o surgimento do ChatGPT, diversos outros modelos foram concebidos na tentativa de superar os limites tecnológicos de geração e compreensão textual.

*Instituto de Computação, Universidade Estadual de Campinas, Campinas, SP, Brazil, 13083-852
†Maritaca AI

No entanto, um avanço particularmente notável nessa sequência de modelos novos foi a introdução da capacidade de compreensão de imagens em modelos como o GPT-4V(ision) [6] e o Google Gemini [7]. Esses anúncios foram marcantes e impactantes para a comunidade da computação, expandindo as capacidades dos modelos de linguagem para incluir inferências e respostas baseadas em imagens, um campo conhecido como *Visual Question Answering* (VQA). Apesar da proeminência do GPT-4V e do Gemini, eles não foram os pioneiros nessa capacidade, nem são os únicos modelos a possui-la. Modelos abertos como OpenFlamingo, LLaVA e CogVLM também demonstram habilidades similares. Paralelamente, conjuntos de dados foram desenvolvidos para avaliar especificamente o desempenho desses modelos em VQA.

Este trabalho foca na exploração de modelos multimodais abertos em um contexto particular: os exames de vestibular da UNICAMP (Universidade Estadual de Campinas) e USP (Universidade de São Paulo) no Brasil. Buscamos compreender a eficácia desses modelos em um domínio que combina desafios linguísticos e visuais, proporcionando uma perspectiva única sobre a evolução da IA multimodal em contextos que exigem uma integração profunda de texto e imagem.

2 Objetivos

Este estudo tem como objetivo principal investigar o desempenho de modelos de aprendizado de máquina selecionados para a resolução de questões de vestibulares da UNICAMP e USP, enfatizando particularmente aquelas que incluem imagens. Pretende-se compreender a capacidade destes modelos em um contexto específico de avaliação acadêmica e responder a várias questões secundárias surgidas dos resultados.

Inicialmente, será realizada uma comparação direta entre os modelos para identificar diferenças significativas em seu desempenho, visando determinar quais são mais eficazes em tarefas que exigem a compreensão conjunta de texto e imagem.

Além disso, o estudo busca entender se o sucesso desses modelos em outros conjuntos de dados, tais como o MM-Vet [9] é um indicativo confiável de seu desempenho nesta base de dados específica. Essa análise é crucial para avaliar a generalizabilidade e a aplicabilidade dos modelos em diferentes contextos.

Uma questão importante a ser explorada é se os modelos multimodais têm desempenho inferior aos modelos puramente de linguagem nas tarefas designadas. Isso será examinado comparando os resultados obtidos por modelos de linguagem no BlueX em questões sem imagens com os resultados deste estudo, fornecendo hipóteses sobre a qualidade dos modelos em suas tarefas específicas. Outro objetivo é avaliar se a inclusão da compreensão de imagens faz com que os modelos multimodais desempenhem melhor que modelos de linguagem, em questões do conjunto de dados que incluem uma característica visual.

Por fim, o estudo compara os resultados dos modelos com as notas humanas nas universidades mencionadas. Esta comparação busca estabelecer um paralelo entre o nível de conhecimento demonstrado pelos modelos de aprendizado de máquina e o conhecimento humano esperado para ingresso nessas instituições. Esta abordagem fornece uma métrica de avaliação relativa e uma perspectiva única sobre o estado atual da inteligência artificial

nesse contexto educacional.

3 Contexto

Nesta seção, aspectos relevantes são descritos a respeito dos conjuntos de dados e dos modelos utilizados neste trabalho.

3.1 Conjunto de Dados

Um componente chave deste trabalho é o conjunto de dados BlueX, que agrega questões dos vestibulares da UNICAMP e USP de 2018 a 2023. A escolha deste conjunto de dados é motivada pela sua diversidade em áreas do conhecimento e pela inclusão de questões que contêm imagens. Embora múltiplos LLMs tenham sido avaliados com esse conjunto de dados no artigo original, conforme visto nas Tabelas 1 e 2, a análise foi restrita a questões sem imagens, evidenciando uma lacuna na avaliação de VQA. Diferentemente de muitas bases de dados de VQA que focam primariamente na análise de imagens, as questões do BlueX requerem uma interpretação equilibrada da parte visual e textual.

Tabela 1: Comparação de desempenho dos modelos de linguagem em questões textuais na base de dados BlueX.

Model	BLUEX	UNICAMP	USP	MR	BK
Highest Cutoff Score	0.863	0.855	0.872	-	-
Average Human Score	0.521	0.530	0.511	-	-
GPT-4	0.748	0.749	0.747	0.447	0.854
Sabiá 65B	0.632	0.615	0.650	0.239	0.775
GPT-3.5-Turbo	0.582	0.580	0.583	0.277	0.764
LLaMA 65B	0.542	0.530	0.557	0.271	0.652
OPT 66B	0.223	0.246	0.197	0.186	0.258

Tabela 2: Desempenho comparativo por disciplina de modelos de linguagem em questões textuais na base de dados BlueX.

Modelo	Biologia	Química	Inglês	Geografia	História	Matemática	Física	Português
GPT-4	0.871	0.675	0.918	0.935	0.930	0.389	0.557	0.805
Sabiá 65B	0.771	0.350	0.837	0.774	0.883	0.278	0.257	0.755
GPT-3.5-Turbo	0.700	0.350	0.714	0.806	0.805	0.259	0.329	0.629
LLaMA 65B	0.657	0.350	0.816	0.677	0.719	0.306	0.286	0.572
OPT 66B	0.229	0.275	0.286	0.161	0.273	0.176	0.200	0.189

O BlueX consiste em mais de 1.000 questões de múltipla escolha. As questões, alternativas e imagens relacionadas foram extraídas automaticamente, seguidas de anotações manuais para correção de erros e adição de metadados, incluindo a posição das imagens.

Os metadados anotados abrangem:

- **Conhecimento Prévio (PRK)**: Indica a necessidade de conhecimento externo à questão.
- **Compreensão Textual (TU)**: Relacionado à necessidade de entender um texto específico.
- **Compreensão de Imagem (IU)**: Relevante para questões que exigem a compreensão de uma imagem.
- **Raciocínio Matemático (MR)**: Associado à habilidade de cálculo e manipulação simbólica.
- **Multilinguagem (ML)**: Para questões que requerem conhecimento em mais de um idioma.
- **Conhecimento Brasileiro (BK)**: Envolve aspectos específicos do Brasil, como história e cultura.
- **Disciplinas Relacionadas**: Lista de disciplinas pertinentes à questão.
- **Imagens Relacionadas**: Elenco das imagens vinculadas à questão. (Nota: o conjunto de dados apresenta uma limitação, exibindo apenas uma imagem por questão).

A anotação da posição das imagens nas questões é especialmente relevante para este estudo. Conforme mencionado no artigo original do conjunto de dados, muitas questões nos exames requerem uma compreensão contextual ou informativa das imagens, um desafio alinhado com as características a serem avaliadas em modelos multimodais. O BlueX, portanto, é proposto como uma ferramenta de avaliação crucial para esses modelos. Para este trabalho, o foco foi nas questões que incluem imagens, resultando em um tratamento específico do conjunto de dados para atender a este escopo.

3.2 Modelos

Neste trabalho, avaliamos três modelos multimodais abertos, cada um com capacidade única de compreensão de texto e imagem, e adequados para tarefas de VQA. Os modelos selecionados são:

- **OpenFlamingo**: O OpenFlamingo é uma versão aberta do modelo Flamingo [23], que representou um avanço na integração de visão computacional e processamento de linguagem natural, incorporando o *encoder* visual CLIP ViT-L/14 [10] com o modelo de linguagem MPT-1B [11] através de camadas de atenção cruzada. A arquitetura do OpenFlamingo, equipada com o *Perceiver Resampler*, permite que as representações visuais do CLIP sejam ajustadas para alinhar-se com as do modelo de linguagem. A intercalação de dados visuais e texto processado é central para o funcionamento do modelo, capacitando-o a gerar respostas textuais a partir de entradas que combinam imagem e texto.

- **LLaVA**: O LLaVA [4] é um modelo de aprendizado de máquina multimodal que também utiliza o *encoder* visual CLIP ViT-L/14, mas combinado ao modelo de linguagem Vicuna 7B v1.5. Na versão 1.5 do LLaVA, utilizada neste estudo, o modelo passou por melhorias importantes, incluindo uma matriz de projeção para conectar as modalidades visual e linguística. Ele foi submetido a um processo de *fine-tuning* em duas etapas, com foco em *Visual Chat* e *Science QA*. As otimizações adicionais incluem o uso de *prompts* de formatação de resposta para equilibrar respostas curtas e longas em tarefas de VQA e a substituição da projeção linear por um MLP de duas camadas, melhorando a capacidade multimodal do modelo. Estas atualizações tornam o LLaVA 1.5 eficaz para tarefas de VQA, especialmente em contextos acadêmicos e científicos, sendo um dos motivos da escolha para esse estudo.

- **CogVLM**: O CogVLM é um modelo mais complexo, composto por quatro componentes fundamentais: um *encoder* ViT, um adaptador MLP, um modelo de linguagem pré-treinado (GPT) e um módulo de especialista visual. Utiliza o EVA2-CLIP-E [12] pré-treinado como encoder ViT, removendo a última camada especializada em aprendizado contrastivo. O adaptador MLP mapeia as saídas do ViT para o mesmo espaço das características textuais. Para o processamento de linguagem, é adotado o Vicuna 7B v1.5, aplicando-se uma máscara causal a todas as operações de atenção. O módulo de especialista visual, adicionado em cada camada, permite um alinhamento profundo das características visuais e linguísticas. Esse design multifacetado do CogVLM propõe uma integração profunda e eficaz de informações textuais e visuais.

Cada um desses modelos apresenta características únicas que os tornam adequados para a análise de questões complexas que combinam texto e imagem. Além disso, todos os modelos avaliados são relevantes para o campo, e, devido ao fato de terem sido concebidos em momentos diferentes, um dos objetivos do trabalho também é avaliar como a tecnologia avançou com o tempo.

As principais informações sobre os modelos, assim como suas pontuações no conjunto de dados MM-Vet, podem ser vistas na Tabela 3.

Tabela 3: Principais informações sobre os modelos.

Modelo	Encoder Visual	Modelo de Linguagem	Quantidade de Parâmetros	Pontuação no MM-Vet
OpenFlamingo	OpenAI CLIP ViT-L/14	MPT 1B	3B	24.8*
LLaVA	OpenAI CLIP ViT-L/14	Vicuna 7B v1.5	9B	31.1
CogVLM	EVA2-CLIP-E	Vicuna 7B v1.5	17B	52.8

* Observação: Essa pontuação do MM-Vet foi obtida pela versão do Flamingo com 9B parâmetros, que utiliza o MPT-7B como modelo de linguagem. Por esse motivo, considerando que o modelo menor deve apresentar um desempenho igual ou inferior, o valor apresentado é uma estimativa de limite superior para a pontuação do OpenFLamingo 3B.

4 Método

Esta seção descreve a metodologia empregada neste estudo para avaliar o desempenho de modelos de aprendizado de máquina multimodais em tarefas de ViQA. Serão explorados em detalhes os processos de pré-processamento de dados, seleção e configuração dos modelos, desenvolvimento de experimentos e avaliações.

4.1 Recursos Computacionais

A execução de *Large Multimodal Models* (LMMs) apresenta desafios significativos devido ao seu extenso número de parâmetros. Tal complexidade exige infraestrutura de alto desempenho para a realização de inferências e, caso necessário, treinamentos ou *fine-tunings* adicionais. Neste estudo, a escolha dos equipamentos foi uma consideração crítica, viabilizando a execução eficaz dos modelos selecionados.

O experimento foi conduzido em uma máquina virtual especificamente configurada para suportar os experimentos realizados. O principal componente desse sistema para a tarefa foram duas unidades de processamento gráfico (GPUs) T4, cada uma com 24 GB de memória, reconhecidas por sua capacidade de processamento acelerado. 189 GB de RAM e um processador de 48 cores complementaram a configuração. Com um armazenamento total de 560 GB, o sistema ofereceu espaço suficiente para hospedar o conjunto de dados BlueX, os modelos de aprendizado de máquina em diversas etapas do desenvolvimento e outros dados essenciais para o experimento.

Esta infraestrutura de robusta foi fundamental para a realização dos experimentos, afetando diretamente a eficiência e a velocidade da inferência dos modelos multimodais, e assegurando a integridade e a confiabilidade dos resultados obtidos.

4.2 Tratamento do Conjunto de Dados

O conjunto de dados BlueX representa uma rica fonte de dados para avaliação de modelos de aprendizado de máquina. No entanto, para os propósitos deste estudo, focado em questões com componentes visuais, foi necessário um processo criterioso de seleção e filtragem das questões.

A seleção inicial abrangeu todas as questões que incluíam imagens. Contudo, uma restrição do BlueX, que documenta apenas a primeira imagem em cada pergunta, juntamente com as limitações de certos modelos multimodais que processam somente uma imagem por inferência, conduziu à exclusão de questões com múltiplas imagens. Assim, o conjunto resultante foi restringido a perguntas contendo uma única imagem e alternativas em texto.

Esta etapa de filtragem gerou um conjunto de 362 questões, que passou por uma verificação manual cuidadosa para assegurar a precisão dos dados. Durante a inspeção, duas questões com anotações incompletas ou incorretas foram identificadas e removidas, resultando em um total de 360 questões válidas para análise.

Adicionalmente, foi notada a duplicidade de *question_id*, que deveria ser um identificador único para cada item do conjunto de dados, em três instâncias. Embora isso represente uma inconsistência no conjunto de dados, optou-se pela manutenção dessas questões no estudo.

As discrepâncias encontradas foram reportadas aos mantenedores do BlueX, iniciando uma colaboração para aperfeiçoamento do conjunto de dados, uma contribuição secundária, mas significativa deste trabalho.

Com o conjunto de dados devidamente preparado, composto por 360 questões selecionadas, procedeu-se à criação de gráficos de distribuição para visualizar a distribuição de conhecimentos necessários e as disciplinas relacionadas a cada questão. Estes gráficos, ilustrados nas Figuras 1 e 2, fornecem um panorama detalhado da composição das questões e são instrumentais para a compreensão das demandas cognitivas impostas aos modelos.

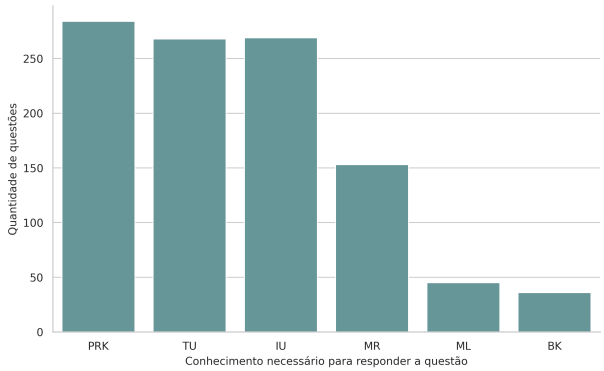

Figura 1: Distribuição das questões por conhecimento necessário para responder.

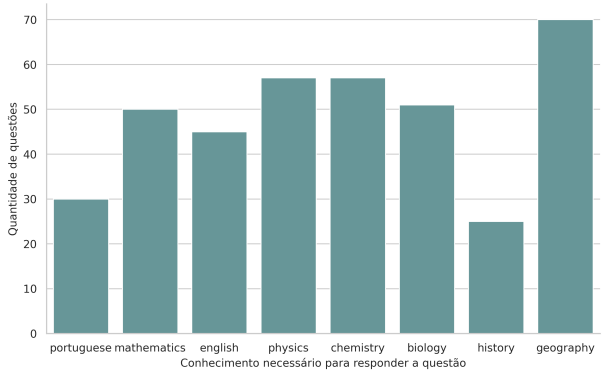

Figura 2: Distribuição das questões por disciplina.

4.3 Preparando a Entrada dos Modelos

Para avaliar os modelos de aprendizado de máquina com a base de dados BlueX, foi essencial estabelecer uma rotina de preparação das entradas que contemplasse tanto os aspectos textuais quanto visuais das questões. As questões do conjunto de dados vêm acompanhadas de informações como texto, ID, alternativas, resposta correta, metadados e imagens codificadas em base64.

Tratamento do Texto: Foi desenvolvida uma função geradora para processar de maneira eficaz esses elementos. A cada chamada, ela retorna uma tupla com o ID da questão, texto da pergunta com alternativas, a imagem associada e a resposta esperada, seguindo o critério de seleção estabelecido.

O tratamento textual começa com a substituição do marcador de posição da imagem no texto. Originalmente, o token "[IMAGE 0]" é utilizado, mas alguns modelos requerem tokens específicos. Por exemplo, o OpenFlamingo usa "<image>", enquanto o LLaVA não exigem nenhum específico, e o padrão foi mantido para ele. Para o CogVLM, após testes com diferentes tokens, "[IMAGEM]" foi a opção adotada por não apresentar variações significativas no desempenho. O texto é então tokenizado usando o tokenizador específico de cada modelo.

Tratamento da Imagem: Para as imagens, o processo começa com a decodificação do base64 para um objeto de imagem utilizável. Utilizamos as bibliotecas *base64* e *Pillow* (PIL) [14] da linguagem de programação Python para transformar a representação codificada em um objeto *Image*, que, para dois dos três modelos, é o formato esperado. Em seguida, cada imagem é pré-processada conforme os requisitos do modelo específico, o que é crucial para assegurar que as imagens estejam no formato correto e compatível.

Combinação de Texto e Imagem: Com o texto e as imagens devidamente processados, eles são combinados para formar a entrada final que será fornecido aos modelos para inferência, de acordo com a documentação de cada modelo.

Detalhes adicionais sobre o pré-processamento específico das imagens e as peculiaridades da inferência para cada modelo serão expostos na Subseção 4.5.

4.4 Análise de Estratégias de Inferência

Avaliar o desempenho de modelos de aprendizado de máquina em tarefas de VQA requer uma estratégia de inferência bem definida. Neste estudo, o impacto de diferentes configurações de inferência foi investigado – especificamente 0-*shot*, 1-*shot*, 2-*shot* e o uso de *Chain of Thought* [25] – no desempenho dos modelos avaliados.

OpenFlamingo: Para o OpenFlamingo, as estratégias de 1-*shot* e 2-*shot* foram implementadas. Nessas configurações, o modelo recebe um ou dois exemplos de perguntas e respostas antes da questão de teste, avaliando a influência de exemplos prévios na capacidade do modelo de compreender e responder a perguntas complexas. Inicialmente, a intenção era também utilizar a estratégia 0-*shot*, mas o modelo não apresentou respostas satisfatórias, independentemente das instruções e *prompts* testados, o que levou à exclusão desta abordagem. Esse detalhe é condizente com as expectativas do modelo, que foi criado para ser utilizado com *Few-Shot*.

LLaVA: Com o LLaVA, foi adotada exclusivamente a abordagem 0-*shot*, testando a habilidade do modelo de deduzir a resposta correta unicamente com base nas informações da pergunta e na imagem, sem exemplos prévios. Esta estratégia enfatiza a capacidade intrínseca do modelo de processar e responder a perguntas baseadas em VQA sem apoio adicional.

CogVLM: Para o CogVLM, houve a execução da avaliação em os dois modos disponíveis do modelo, os modos chat e VQA. O modo VQA tende a gerar respostas concisas e diretas,

enquanto o modo chat favorece respostas mais verbosas. No modo VQA, as configurações 0-*shot* e 1-*shot* foram testadas, enquanto no modo chat, além da abordagem 0-shot, foi aplicada a técnica *Chain of Thought*, que exige uma maior elaboração das respostas. Esta técnica busca eliciar um raciocínio detalhado e explícito, que pode melhorar a qualidade das respostas em modelos de linguagem.

Para garantir a clareza nas referências futuras, todos os *prompts* utilizados serão listados a seguir:

- Prompt 0-shot:

  > Você receberá uma pergunta que está relacionada à imagem, e sua tarefa é responder a alternativa correta.
  > Pergunta: `<question>`.
  > A resposta correta é:

- Prompt 1-shot:

  > `<sample_question>`. A resposta correta é: '`<sample_ground_truth>`'.
  > `<question>`. A resposta correta é:

- Prompt 2-shot:

  > `<sample_question1>`. A resposta correta é: '`<sample_ground_truth1>`'.
  > `<sample_question2>`. A resposta correta é: '`<sample_ground_truth2>`'.
  > `<question>`. A resposta correta é:

- Prompt Chain-of-thought:

  > A seguir, será fornecida uma questão para você, contendo uma imagem e as possíveis alternativas. Sua tarefa é escrever o passo a passo do seu raciocínio até a resposta, e por fim responder com a alternativa correta.
  > Questão: `<question>`.
  > Vamos pensar passo-a-passo:

4.5 Inferência dos Modelos

Com estratégias de inferência estabelecidas e o conjunto de dados preparado, a fase seguinte consiste na execução da inferência nos modelos selecionados. Cada modelo apresenta particularidades que requerem atenção detalhada, portanto, esta seção é dividida para discutir individualmente o processo aplicado a cada um deles.

4.5.1 Flamingo

O processo de inferência para o Flamingo inicia com a instalação da biblioteca necessária [18] e a instanciação do modelo, do processador de imagens e do tokenizador, conforme documentação oficial.

Código 1: Carregamento do OpenFlamingo na GPU

```
from open_flamingo import create_model_and_transforms

model, image_processor, tokenizer = create_model_and_transforms(
    clip_vision_encoder_path="ViT-L-14",
    clip_vision_encoder_pretrained="openai",
    lang_encoder_path="anas-awadalla/mpt-1b-redpajama-200b",
    tokenizer_path="anas-awadalla/mpt-1b-redpajama-200b",
    cross_attn_every_n_layers=1,
)

checkpoint_path = hf_hub_download("openflamingo/OpenFlamingo-3B-vitl-mpt1b", "checkpoint.pt")
model.load_state_dict(torch.load(checkpoint_path), strict=False)

model.to("cuda")
```

Uma vez carregado o *checkpoint* e o modelo, preparamos as entradas. Utilizando uma função geradora, como mencionado anteriormente, obtemos iterativamente as questões e as imagens.

Para o Flamingo, que utiliza abordagens 1-*shot* e 2-*shot*, funções geradoras auxiliares fornecem exemplos adicionais, garantindo a diversidade e a relevância dos exemplos para a inferência. Os exemplos obtidos são dependentes da questão que está sendo avaliada. Em todos os casos, os exemplos são questões da mesma universidade, mas de um ano diferente da questão analisada.

A seguir, o pré-processamento das imagens transforma-as em tensores compatíveis com o modelo. Para o texto, os *prompts* que moldam o formato desejado da inferência são criados e, em seguida, o tokenizador é utilizado para gerar os tensores correspondentes.

Código 2: Processamento das entradas visuais para o Flamingo

```
image_data = base64.b64decode(base64_image)
# Convert to an image
image = Image.open(io.BytesIO(image_data))

# Preprocess the image:
vision_x = [image_processor(sample_image).unsqueeze(0), image_processor(image).unsqueeze(0)]
# Convert vision_x to a tensor and move it to GPU
vision_x = torch.cat(vision_x, dim=0).cuda()  # or .cuda()
vision_x = vision_x.unsqueeze(1).unsqueeze(0)
```

Código 3: Processamento das entradas de texto para o Flamingo

```
tokenizer.padding_side = "left"
lang_x = tokenizer(
    [<prompt>],
    return_tensors="pt",
).to(device)
```

Finalmente, a função de geração do modelo é invocada com as entradas e configurações adequadas, como o limite de *tokens* e a temperatura.

Código 4: Geração da resposta do Flamingo.

```
gen_text = model.generate(
        vision_x=vision_x,
        lang_x=lang_x["input_ids"],
        attention_mask=lang_x["attention_mask"],
        max_new_tokens=20,
        num_beams=1,
        temperature=0.0,
    )
response = tokenizer.decode(gen_text[0])
```

A resposta gerada é registrada para análise posterior, marcando o fim de um ciclo completo de inferência para uma questão do conjunto de dados BlueX utilizando o modelo Flamingo. Esse ciclo se repete até que todas as questões tenham sido avaliadas.

4.5.2 LLaVA

O processo de inferência para o modelo LLaVA inicialmente considerou a utilização da *web demo* do projeto, que oferece uma interface de chat para interação com o modelo. Contudo, essa abordagem mostrou-se ineficiente e suscetível a erros humanos, além de ser problemática devido à ocorrência de descartes parciais das entradas pelo modelo, comprometendo a confiabilidade dos resultados.

Diante desses desafios, uma solução mais robusta foi adotada: a execução da inferência localmente, com o uso do pacote LLaVA [19] disponibilizado no GitHub.

Para a execução, seguiu-se a documentação do pacote, criando um dicionário com os argumentos necessários e invocando a função *eval_model* para a inferência.

A manipulação das imagens apresentou um obstáculo, uma vez que o modelo espera receber o caminho do arquivo da imagem em vez de um objeto Image. A biblioteca *tempfile* [17] foi empregada para criar arquivos temporários das imagens, cujos caminhos foram passados como argumentos para a função de avaliação.

Código 5: Código para salvar imagem em um arquivo temporário.

```
import base64
import tempfile

image_data = base64.b64decode(base64_image_representation)

with tempfile.NamedTemporaryFile(delete=False, suffix='.png', mode='
    wb') as temp_image:
        temp_image.write(image_data)
        temp_image_path = temp_image.name
```

Outro ponto de atenção foi a captura do texto gerado, que é exibido via *stdout* e não retornado pela função. Para solucionar esse problema, as bibliotecas *sys* [15] e *io* [16] foram

usadas para capturar e salvar as saídas do *stdout*, armazenando as respostas geradas para análise subsequente.

Código 6: Código para recuperar a resposta do stdout.

```
import sys
import io
original_stdout = sys.stdout
captured_output = io.StringIO()
sys.stdout = captured_output
...
sys.stdout = original_stdout
response = captured_output.getvalue()
```

Código 7: Código para realizar a inferência do LLaVA.

```
model_path = "liuhaotian/llava-v1.5-7b"
prompt = <0-shot prompt>
image_file = temp_image_path

args = type('Args', (), {
    "model_path": model_path,
    "model_base": None,
    "model_name": get_model_name_from_path(model_path),
    "query": prompt,
    "conv_mode": None,
    "image_file": image_file,
    "sep": ",",
    "temperature": 0,
    "top_p": 1,
    "num_beams": 3,
    "max_new_tokens": 100
})()

eval_model(args)
```

Esse procedimento assegura que as respostas geradas pelo LLaVA sejam coletadas de maneira sistemática e estruturada, facilitando a etapa de análise e avaliação dos resultados obtidos pelo modelo.

4.5.3 CogVLM

A inferência para o CogVLM começa com a configuração do modelo e do tokenizador, empregando a biblioteca Transformers [20] e HuggingFace Hub [21]. O LlamaTokenizer foi utilizado como tokenizador, especificando a versão do Vicuna correspondente.

Durante a instanciação do modelo, parâmetros críticos são definidos, incluindo a temperatura e o *top_p*, essenciais para controlar a aleatoriedade da geração de texto. A configuração do modelo em 16 bits com o formato *bfloat* do PyTorch [22] e a ativação do parâmetro *low_cpu_mem_usage* otimizam o uso do hardware disponível.

Devido às limitações de memória da GPU T4, foi necessário um mapeamento de dispositivos para aproveitar múltiplas GPUs e carregar o modelo de forma eficaz.

Código 8: Instanciação do CogVLM nas múltiplas GPUs.

```
tokenizer = LlamaTokenizer.from_pretrained('lmsys/vicuna-7b-v1.5')
with init_empty_weights():
    model = AutoModelForCausalLM.from_pretrained(
        'THUDM/cogvlm-chat-hf',
        torch_dtype=torch.bfloat16,
        low_cpu_mem_usage=True,
        trust_remote_code=True,
        temperature=0.0,
        top_p=1.0,
    )

device_map = infer_auto_device_map(model, max_memory={0:'18GiB',1:'18GiB','cpu':'18GiB'}, no_split_module_classes='CogVLMDecoderLayer')

model = load_checkpoint_and_dispatch(
    model,
    '/path/to/cached/model',    # typical, '~/.cache/huggingface/hub/models--THUDM--cogvlm-chat-hf/snapshots/balabala'
    device_map=device_map,
)
model = model.eval()
```

Com o modelo carregado, a preparação das entradas é o próximo passo. A função geradora fornece as perguntas e imagens, e o *prompt* é formatado de acordo com a técnica de inferência selecionada: 0-*shot*, 1-*shot* ou 0-*shot* com *Chain of Thought*.

O método *build_conversation_input_ids* é utilizado para organizar as entradas, preparando os tensores necessários para a inferência. Finalmente, o método *generate* é chamado para produzir as respostas.

Código 9: Preparação de entradas e realização da inferência do CogVLM.

```
query = <prompt>
image = <Image object>

inputs = model.build_conversation_input_ids(tokenizer, query=query, history=[], images=[image], template_version='chat') # template_version can also be 'vqa'
inputs = {
    'input_ids': inputs['input_ids'].unsqueeze(0).to('cuda'),
    'token_type_ids': inputs['token_type_ids'].unsqueeze(0).to('cuda'),
    'attention_mask': inputs['attention_mask'].unsqueeze(0).to('cuda'),
    'images': [[inputs['images'][0].to('cuda').to(torch.bfloat16)]],
}
```

```
gen_kwargs = {"max_length": 2048, "do_sample": False}
with torch.no_grad():
    outputs = model.generate(**inputs, **gen_kwargs)
    outputs = outputs[:, inputs['input_ids'].shape[1]:]
```

As respostas são armazenadas para análise posterior, concluindo o procedimento de inferência para o modelo CogVLM.

4.6 Coleta de Métricas

Após a fase de inferência, um componente crucial foi a análise e o pós-processamento das respostas geradas pelos modelos, visando à conversão para um formato padronizado, adequado para a análise de métricas.

Dada a diversidade nos formatos das respostas geradas, uma metodologia rigorosa foi necessária para garantir a consistência e a precisão na avaliação do desempenho dos modelos.

Inicialmente, todas as respostas foram submetidas a um processo automatizado para identificar padrões comuns de resposta, facilitando a extração automática da alternativa correta. Nos casos em que essa abordagem foi suficiente, a resposta foi extraída diretamente. No entanto, muitas respostas exigiram uma análise manual mais detalhada, principalmente quando as respostas estavam inseridas em textos sem estrutura clara ou definida, ou quando o modelo respondia com o texto da alternativa em vez da letra correspondente. Nesses casos, cada resposta foi cuidadosamente avaliada para determinar a alternativa correta, garantindo a consistência com o *ground_truth*, que contém apenas a letra da alternativa correta. As respostas que não puderam ser mapeadas para uma alternativa específica foram marcadas com um símbolo padrão "-" para manter a uniformidade, e indicar o erro do modelo.

Com as respostas formatadas adequadamente, procedi ao cálculo das métricas de desempenho. A principal métrica utilizada foi a acurácia, obtida pela comparação direta entre as respostas geradas pelos modelos e as respostas esperadas. Essa comparação permitiu avaliar a eficácia de cada modelo em fornecer respostas corretas, fornecendo uma base quantitativa sólida para a análise dos resultados.

5 Resultados

Os resultados detalhados deste estudo, após a aplicação da metodologia descrita, estão resumidos nas Tabelas 4 e 5. Para uma análise mais aprofundada, as saídas completas geradas por este trabalho estão disponíveis no GitHub, acessíveis através do link: `https://github.com/GuilhermeDobins/resultados_pfg`. A análise desses resultados revelou percepções valiosas sobre a eficiência dos modelos de aprendizado de máquina em tarefas de VQA no conjunto de dados BlueX.

Análise Comparativa de Acurácias: As acurácias gerais dos modelos, assim como as acurácias específicas por habilidades e disciplinas, foram calculadas. Observou-se que, na melhor configuração de cada modelo, o LLaVA em 0-*shot* obteve a maior acurácia, seguido pelo CogVLM em modo VQA 0-shot e pelo Flamingo 1-*shot*. Notavelmente, as diferenças de acurácia entre os modelos não foram tão acentuadas quanto as observadas em bases de

Tabela 4: Acurácia dos modelos por habilidade exigida pela questão.

Modelo	BLUEX	UNICAMP	USP	PRK	TU	IU	MR	ML	BK
Maior nota de corte	0.863	0.855	0.872	-	-	-	-	-	-
Nota humana média	0.521	0.530	0.511	-	-	-	-	-	-
Flamingo One-Shot	0.275	0.322	0.231	0.261	0.259	0.301	0.237	0.279	0.306
Flamingo Two-Shot	0.241	0.263	0.22	0.254	0.214	0.211	0.25	0.256	0.25
LLaVA	**0.364**	**0.351**	**0.376**	**0.336**	**0.395**	**0.342**	0.243	**0.535**	**0.444**
CogVLM (vqa)	0.317	0.327	0.306	0.293	0.316	0.305	0.257	0.395	0.361
CogVLM (chat)	0.314	0.339	0.29	0.3	0.293	0.301	0.257	0.395	0.333
CogVLM+CoT	0.258	0.281	0.237	0.24	0.286	0.256	0.211	0.233	0.306
CogVLM (vqa) 1-Shot	0.277	**0.351**	0.21	0.258	0.282	0.278	**0.27**	0.326	0.194

Tabela 5: Acurácia dos modelos por disciplina.

Modelo	Português	Matemática	Inglês	Física	Química	Biologia	História	Geografia
Flamingo 1-Shot	0.367	**0.32**	0.279	0.175	0.281	0.235	0.4	0.275
Flamingo 2-Shot	0.133	0.26	0.256	0.246	**0.333**	0.216	0.36	0.232
LLaVA	**0.53**	0.26	**0.535**	**0.281**	0.281	0.373	**0.48**	**0.377**
CogVLM	0.367	0.22	0.395	**0.281**	0.281	0.412	0.44	0.246
CogVLM (chat)	0.367	0.26	0.395	0.211	0.298	**0.431**	0.36	0.275
CogVLM+CoT	0.433	0.18	0.233	**0.281**	0.193	0.235	0.4	0.203
CogVLM 1-Shot	0.267	**0.32**	0.326	0.246	0.228	0.275	0.28	0.246

dados como o MM-Vet. Isso indica que o BlueX, com suas peculiaridades, pode desafiar os modelos de maneiras distintas dos conjuntos de dados tradicionais.

Reflexões e Inferências: O desempenho superior do LLaVA sobre o CogVLM no BlueX destaca que mesmo os grandes modelos gerados a partir de pré-treinamento em enormes bases de dados podem apresentar diferenças de desempenho em domínios específicos, permitindo que, em alguns conjuntos de dados, um dos modelos seja muito superior ao outro, mas em outras tarefas apresente uma acurácia menor.

Desafios da Multimodalidade: A ausência de modelos multimodais superando as notas humanas médias, contrastando com o sucesso de alguns LLMs em questões sem imagens, sugere que a integração eficaz de informações visuais e textuais permanece um desafio significativo, ao menos em modelos *open-source*.

Limitações de Modelos Multimodais: A discrepância no desempenho entre os modelos multimodais e os de linguagem pura pode indicar limitações intrínsecas dos primeiros em tarefas que requerem compreensão integrada e profunda de texto e imagem, com uma importância equilibrada para a geração da resposta correta. Isso pode ser corroborado pela superior qualidades dos modelos em conjuntos de dados com maior ênfase nas características visuais das perguntas.

Contexto e Domínio: A variação no desempenho dos modelos para as diferentes disciplinas presentes no conjunto de dados, e com os diferentes conhecimentos necessários para responder às perguntas, reforça a noção de que a eficácia dos modelos de IA ainda é, em certos casos, dependente do contexto e do domínio específico. Isso fica particularmente

evidenciado ao observarmos a variação da eficácia dos modelos Flamingo e CogVLM nas questões da UNICAMP e da USP.

Comparação com Desempenho Humano: Nenhum dos modelos testados superou a média das notas humanas no BlueX, nem alcançou metade da maior nota de corte humana. Esse resultado se torna ainda mais intrigante ao comparar com o desempenho de LLMs como o LLaMA 65B e o GPT-4 em questões do BlueX sem imagens, onde eles atingem ou ultrapassam a média humana.

Para explorar a hipótese de que a multimodalidade pode afetar negativamente a eficácia do componente textual, um teste de ablação foi conduzido. Os detalhes e resultados deste teste serão discutidos na Subseção 5.1.

5.1 Estudo de Ablação

O estudo de ablação foi conduzido para avaliar o impacto da multimodalidade nos resultados, comparando os modelos LLaVA e CogVLM com o Vicuna 7B v1.5, que serve como o componente de linguagem em ambos. O objetivo era determinar se a integração de um *encoder* visual e informações adicionais provenientes das imagens melhoraria o desempenho.

Implementação do Estudo: Para este propósito, o Vicuna foi utilizado para responder às mesmas questões do BlueX as quais os outros modelos foram submetidos, mas com a exclusão das imagens (visto que o Vicuna não seria capaz de processá-las). Em vez disso, a posição das imagens hipotéticas foram marcadas com o token "[IMAGEM 0]". A inferência foi realizada de forma similar ao melhor resultado anterior, o LLaVA 0-shot, fornecendo apenas a instrução seguida pela pergunta, sem a imagem correspondente. Os resultados podem ser vistos nas Tabelas 6 e 7.

Tabela 6: Resultados da ablação por habilidade exigida pela questão.

Modelo	BLUEX	UNICAMP	USP	PRK	TU	IU	MR	ML	BK
Vicuna 7B	**0.401**	**0.468**	0.339	**0.36**	**0.398**	**0.376**	**0.303**	**0.628**	**0.444**
LLaVA	0.364	0.351	**0.376**	0.336	0.395	0.342	0.243	0.535	**0.444**
CogVLM (vqa)	0.317	0.327	0.306	0.293	0.316	0.305	0.257	0.395	0.361
CogVLM (chat)	0.314	0.339	0.29	0.3	0.293	0.301	0.257	0.395	0.333

Tabela 7: Resultados da ablação por disciplina.

Modelo	Português	Matemática	Inglês	Física	Química	Biologia	História	Geografia
Vicuna 7B	0.4	**0.36**	**0.628**	0.228	0.228	0.412	**0.6**	**0.464**
LLaVA	**0.53**	0.26	0.535	**0.281**	0.281	0.373	0.48	0.377
CogVLM	0.367	0.22	0.395	**0.281**	0.281	0.412	0.44	0.246
CogVLM (chat)	0.367	0.26	0.395	0.211	**0.298**	**0.431**	0.36	0.275

Surpreendentemente, o Vicuna superou os modelos multimodais em termos de acurácia geral, apesar de ser superado em algumas disciplinas específicas. Esta observação é notável, pois indica que o Vicuna, um modelo baseado apenas em texto, foi capaz de superar os

modelos LLaVA e CogVLM, mesmo em questões que teoricamente beneficiariam da multimodalidade.

Algumas implicações e reflexões sobre os resultados obtidos são apresentadas a seguir:

- **Relevância do Texto no BlueX**: Os resultados sugerem que a componente textual das perguntas do BlueX é bastante informativa, a ponto de permitir que o Vicuna compense a falta de dados visuais. Isso implica que, para o BlueX, a multimodalidade, nos modelos avaliados, não só não trouxe melhorias significativas, como também pode ter prejudicado o desempenho.
- **Desafio da Integração Multimodal**: A eficácia limitada da multimodalidade no BlueX ressalta a dificuldade dos modelos atuais em processar e integrar eficientemente informações textuais e visuais, especialmente em contextos onde as imagens complementam, mas não dominam a questão.

6 Conclusões e Trabalhos Futuros

Ao longo deste trabalho, foi explorado o desempenho de modelos de IA multimodais no contexto específico do conjunto de dados BlueX, focado em questões de vestibulares brasileiros acompanhadas de imagens. As descobertas revelam aspectos críticos sobre a eficácia desses modelos em comparação com conjuntos de dados mais gerais, como o MM-Vet, e oferecem informações valiosas sobre a multimodalidade em tarefas de VQA.

1. **Desempenho Inferior em Comparação com Conjuntos de Dados Gerais**: Observou-se que os modelos testados apresentaram um desempenho significativamente inferior no BlueX em comparação com suas pontuações em conjuntos de dados mais abrangentes, como o MM-Vet. Essa discrepância sugere que o BlueX, com suas peculiaridades e exigências específicas, pode estar destacando diferentes aspectos das capacidades dos modelos que não são necessariamente avaliados em conjuntos de dados mais estabelecidos.
2. **Menor Disparidade de Pontuação e Surpreendente Desempenho do LLaVA**: Curiosamente, a diferença na pontuação entre os modelos no BlueX foi bem menor do que em outros conjuntos de dados. Notavelmente, o LLaVA, apesar de ter um desempenho inferior ao CogVLM em conjuntos de dados tradicionais, conseguiu obter uma pontuação maior no BlueX. Isso indica que o BlueX é capaz de avaliar características únicas dos modelos, possivelmente relacionadas à compreensão de nuances linguísticas e culturais específicas das questões de vestibular brasileiro, e também ao seu *fine-tuning* em questões com contextos científicos.
3. **Diferenças de Desempenho não Correspondem entre Conjuntos de Dados**: O desempenho dos modelos em outros conjuntos de dados não se traduziu diretamente para o BlueX. Essa constatação reforça a ideia de que diferentes conjuntos de dados testam habilidades variadas e que um bom desempenho em um cenário não garante resultados semelhantes em outro.

4. **Inferioridade dos Modelos Multimodais em Relação ao Modelo Textual**: Ao comparar o CogVLM e o LLaVA com o Vicuna, sua componente textual, notou-se que os modelos multimodais tiveram um desempenho inferior, mesmo em um contexto em que as informações visuais deveriam ser valiosas. Isso sugere que, no domínio do BlueX, a integração de informações visuais pode não ter sido efetivamente realizada ou que as perguntas textuais por si só já fornecem informações suficientes para a inferência de algumas perguntas. É possível especular que os modelos avaliados, em seu estado atual, podem ainda não estar otimizados para aproveitar plenamente as informações visuais de maneira que complemente e melhore o entendimento textual.

5. **Desempenho Geral dos Modelos Multimodais**: Além disso, é importante notar que o desempenho geral dos modelos multimodais no BlueX se mostrou inferior à média humana, e também ao desempenho de grandes modelos de linguagem, com a observação de que as questões avaliadas para os modelos de linguagem não foram as mesmas, e sim o outro subconjunto do BlueX que não contem imagens. Isso levanta questões sobre as limitações atuais da tecnologia multimodal e a necessidade de desenvolvimentos futuros que possam efetivamente integrar e utilizar múltiplos tipos de dados para melhorar a eficácia.

Além das conclusões obtidas a partir deste estudo, diversas oportunidades para pesquisas futuras emergem, visando ampliar o entendimento dos modelos de IA multimodais e seu potencial em aplicações educacionais e além.

1. **Investigação de Modelos Proprietários**: Uma direção promissora é a avaliação de modelos proprietários, como o GPT-4V da OpenAI ou o Google Gemini, no contexto do conjunto de dados BlueX. Isso possibilitaria a comparação direta entre as tecnologias de modelos de aprendizado de máquina multimodais *open-source* e *close-source*. O objetivo seria determinar se e em que medida os modelos proprietários superam os modelos *open-source* em tarefas de VQA específicas, contribuindo assim para uma compreensão mais abrangente do estado atual da tecnologia de IA. Além disso, alguns dos melhores resultados obtidos nas questões sem imagem do BlueX partiram de modelos como o GPT-3.5 e GPT-4, indicando que ainda existe uma distância entre a qualidade das alternativas *close-source* e *open-source* mais difundidas. Outra possibilidade que surge nessa proposta é um novo estudo de ablação, visto que, o GPT-4 possui versões puramente textuais e outra multimodal, permitindo uma comparação direta.

2. **Testes com Novos Modelos Multimodais Abertos**: Com o contínuo avanço da tecnologia de IA, é fundamental avaliar novos modelos multimodais *open-source* à medida que são lançados. Essa avaliação contínua permitirá rastrear o progresso e as melhorias no campo da IA multimodal ao longo do tempo, oferecendo informações sobre como as inovações mais recentes estão moldando as capacidades e eficácias dos modelos.

3. **Estudos de Ablação em Outros Conjuntos de Dados**: Expandir os estudos de ablação para incluir outros conjuntos de dados pode oferecer uma visão mais rica sobre

o comportamento dos modelos multimodais em diferentes contextos. Especialmente, explorar conjuntos de dados que variam em termos de conteúdo, complexidade, requisitos de multimodalidade e principalmente na relevância do componente textual para as respostas. Isso pode ajudar a entender as nuances do desempenho dos modelos e suas aplicações ideais.

Ao concluir este estudo, torna-se evidente que o campo da IA multimodal está em constante evolução, com vastas oportunidades para pesquisa e desenvolvimento. A exploração contínua de novos modelos, conjuntos de dados e técnicas de integração não só aprimorará nossa compreensão das capacidades atuais da IA, mas também abrirá caminho para avanços significativos na maneira como interagimos e utilizamos a tecnologia de IA em contextos educacionais e além.

Referências

[1] T. S. Almeida, T. Laitz, G. K. Bonás, e R. Nogueira, "BLUEX: A Benchmark based on Brazilian Leading Universities Entrance eXams", 2023. [Online]. Disponível: `https://arxiv.org/abs/2307.05410`.

[2] A. Awadalla, I. Gao, J. Gardner, J. Hessel, Y. Hanafy, W. Zhu, K. Marathe, Y. Bitton, S. Gadre, J. Jitsev, S. Kornblith, P. W. Koh, G. Ilharco, M. Wortsman, e L. Schmidt, "OpenFlamingo", mar. 2023, Zenodo, versão v0.1.1. [Online]. Disponível: `https://doi.org/10.5281/zenodo.7733589`.

[3] H. Liu, C. Li, Y. Li, e Y. J. Lee, "Improved Baselines with Visual Instruction Tuning", 2023. [Online]. Disponível: `https://arxiv.org/abs/2310.03744`.

[4] H. Liu, C. Li, Q. Wu, e Y. J. Lee, "Visual Instruction Tuning", 2023. [Online]. Disponível: `https://arxiv.org/abs/2304.08485`.

[5] W. Wang, Q. Lv, W. Yu, W. Hong, J. Qi, Y. Wang, J. Ji, Z. Yang, L. Zhao, X. Song, J. Xu, B. Xu, J. Li, Y. Dong, M. Ding, e J. Tang, "CogVLM: Visual Expert for Pretrained Language Models", 2023. [Online]. Disponível: `https://arxiv.org/abs/2311.03079`.

[6] OpenAI, "GPT-4V(ision) System Card", 25 de setembro de 2023. [Online]. Disponível: `https://cdn.openai.com/papers/GPTV_System_Card.pdf`.

[7] Gemini Team, Google, DeepMind, "Gemini", Disponível em: `https://deepmind.google/technologies/gemini`.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, e L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database", in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[9] W. Yu, Z. Yang, L. Li, J. Wang, K. Lin, Z. Liu, X. Wang, e L. Wang, "MM-Vet: Evaluating Large Multimodal Models for Integrated Capabilities", *arXiv preprint arXiv:2308.02490*, 2023. [Online]. Disponível: `https://arxiv.org/abs/2308.02490`.

[10] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, e I. Sutskever, "Learning Transferable Visual Models From Natural Language Supervision", *arXiv preprint arXiv:2103.00020*, 2021. [Online]. Disponível: `https://arxiv.org/pdf/2103.00020`.

[11] A. Awadalla, "mpt-1b-redpajama-200b", Hugging Face Models, [Online]. Disponível: `https://huggingface.co/anas-awadalla/mpt-1b-redpajama-200b`.

[12] Q. Sun, Y. Fang, L. Wu, X. Wang, e Y. Cao, "EVA-CLIP: Improved Training Techniques for CLIP at Scale", *arXiv preprint arXiv:2303.15389*, Beijing Academy of Artificial Intelligence, Huazhong University of Science and Technology, 2023. [Online]. Disponível: `https://arxiv.org/abs/2303.15389`.

[13] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, e E. P. Xing, "Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality", Março 2023. [Online]. Disponível: `https://lmsys.org/blog/2023-03-30-vicuna/`.

[14] "Pillow", Python Package Index (PyPI). [Online]. Disponível: `https://pypi.org/project/Pillow/`.

[15] Python Software Foundation, "sys — System-specific parameters and functions", Python 3 documentation. [Online]. Disponível: `https://docs.python.org/3/library/sys.html`.

[16] Python Software Foundation, "io — Core Tools for Working with Streams", Python 3 documentation. [Online]. Disponível: `https://docs.python.org/3/library/io.html`.

[17] Python Software Foundation, "tempfile — Generate temporary files and directories", Python 3 documentation. [Online]. Disponível: `https://docs.python.org/3/library/tempfile.html`.

[18] ML Foundations, "open_flamingo", GitHub, [Online]. Disponível: `https://github.com/mlfoundations/open_flamingo`.

[19] H. Liu, "LLaVA", GitHub, [Online]. Disponível: `https://github.com/haotian-liu/LLaVA`.

[20] Hugging Face, "Transformers: State-of-the-Art Natural Language Processing for Pytorch and TensorFlow 2.0", [Online]. Disponível: `https://github.com/huggingface/transformers`.

[21] Hugging Face, "huggingface_hub: Utilities for Hugging Face Hub." [Online]. Disponível: `https://github.com/huggingface/huggingface_hub`.

[22] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison,

A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, e S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library", in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[23] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Binkowski, R. Barreira, O. Vinyals, A. Zisserman, e K. Simonyan, "Flamingo: a Visual Language Model for Few-Shot Learning", *ArXiv preprint arXiv:2204.14198*, 2022. [Online]. Disponível: `https://arxiv.org/abs/2204.14198`.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All You Need", in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[25] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, D. Zhou, "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models", Google Research, Brain Team, 2022, arXiv:2201.11903.