Modelos de linguagem são poliglotas? Uma avaliação em tarefas de raciocínio sobre grafos

Erik J. Nascimento
Departamento de Computação
Instituto Federação do Ceará (IFCE)
Fortaleza, Brasil

Isabelly Pinheiro
Departamento de Computação
Instituto Federal do Ceará (IFCE)
Maracanaú, Brasil

Tarciano S. Filho
Programa de Pós-Graduação em Computação
Instituto Federal do Ceará (IFCE)
Fortaleza, Brasil

Amauri H. Souza
Departamento de Computação
Instituto Federal do Ceará (IFCE)
Fortaleza, Brasil

João V. de Souza Albuquerque
Departamento de Computação
Instituto Federal do Ceará (IFCE)
Maracanaú, Brasil

***Abstract*—Modelos de linguagem de larga escala (*large language models*, LLMs) têm se destacado como tecnologias revolucionárias por sua capacidade de gerar textos com qualidade similar à escrita humana em diversos idiomas e contextos. Embora originalmente projetados para dados textuais, esses modelos estão sendo cada vez mais utilizados para processar dados de diferentes modalidades, incluindo imagens e grafos. Nesse contexto, embora trabalhos recentes tenham se concentrado no desenvolvimento de novos benchmarks e métodos de codificação de grafos para LLMs, o impacto do idioma no desempenho preditivo desses modelos ainda permanece amplamente inexplorado. Este é o primeiro estudo (até onde se sabe) sobre o impacto de diferentes idiomas do *prompt* de entrada na performance de LLMs para tarefas de "raciocínio" envolvendo grafos. Mais especificamente, foi verificado como as acurácias de duas classes de LLMs (Llama e Gemma) variam em função do idioma ao longo de diferentes tipos de codificação de grafos (e.g., lista de adjacências) e tarefas (e.g., contagem de arestas). Além disso, foram considerados quatro idiomas: inglês, português, espanhol e mandarim. Notavelmente, os resultados mostram que existe um impacto significativo do idioma na performance dos LLMs, especialmente para codificadores textuais de grafos que produzem textos mais longos. Além disso, a escolha ótima de LLM para uma dada tarefa é altamente sensível à escolha de idioma e codificador. De maneira geral, o trabalho fornece evidências que LLMs possuem diferentes níveis de fluência a depender do idioma, da tarefa e da representação textual dos grafos de entrada.**

***Index Terms*—LLMs, avaliação, aprendizagem em grafos**

I. Introdução

Modelos de linguagem de larga escala (*large language models*, LLMs) [1, 2] vêm se mostrando uma das tecnologias mais revolucionárias dos últimos tempos. Treinados em dados extensivos, esses modelos atualmente conseguem produzir textos que se assemelham à escrita humana em vários idiomas e contextos. Eles exibem capacidade de entender e criar linguagem, tornando-os úteis para um amplo domínio de aplicações, incluindo tradução de linguagem [3, 4], geração de textos literários [5, 6], raciocínio complexo [7, 8] e, mais recentemente, em tarefas envolvendo dados estruturados em grafos [9–11].

Recentemente, tem crescido o interesse na utilização de LLMs em tarefas de raciocínio em grafos [9, 11, 12]. Apesar disso, a maioria dos estudos existentes têm se concentrado, principalmente, em três frentes relacionadas ao processo de engenharia de *prompt*: codificar a estrutura dos grafos em formatos compreensíveis por LLMs [9, 11], gerar *prompts* adaptados a dados em grafos [12], e desenvolver benchmarks específicos para avaliar esses métodos de codificação [9, 12]. Assim, uma questão crucial continua pouco explorada: **Qual o impacto do idioma do texto de entrada no desempenho preditivo de LLMs em tarefas envolvendo grafos?** Embora seja um fato que LLMs possuam capacidade de processar múltiplos idiomas e trabalhem bem com geração textual simples, ainda não está claro se esses modelos conseguem processar tarefas mais complexas com a mesma precisão em diferentes idiomas.

Neste trabalho, foi estudado de forma empírica **como o idioma do texto de entrada afeta a performance de LLMs em resolver tarefas envolvendo dados estruturados como grafos**. Para este fim, foi criada uma metodologia que consiste em transformar grafos de entrada, gerado pelo benchmark GraphQA [9], em uma representação textual; traduzir essa representação + descrição da tarefa para o idioma alvo; computar a acurácia do modelo (0 se o LLM errou e 1 caso contrário). São avaliados quatro idiomas: inglês, chinês, espanhol e português, dois LLMs (i.e., *gemma2-9b-it* e *llama-3.3-70b-versatile*), cinco tarefas preditivas (e.g., existência de ciclo, grau do nó) e quatro tipos de codificadores *grafo-texto* (e.g., lista de adjacências, redes sociais).

Os resultados experimentais mostram que, quando são utilizados codificadores que produzem um maior volume de texto (como codificadores de "amizade" e "rede social"), existe um impacto direto do idioma na qualidade da resposta do LLM. Além disso, é interessante observar que a língua inglesa, apesar de ser mais de 89% do corpus de treinamento dos modelos [13, 14], nem sempre é a que possui a maior acurácia. Para uma mesma tarefa, detectar grau de um vértice, o nível de concordância entre Português e Inglês pode variar

de 100% (codificador: "amizade", LLM: *Gemma*) para 62% (codificador: "rede social", LLM: *Llama*). Adicionalmente, os resultados demonstram uma alta sensibilidade no desempenho preditivo em função do codificador. Ou seja, para uma mesma tarefa, idioma, e LLM, resultados podem variar de 40% a 94% de acurácia. Portanto, a escolha do melhor modelo para uma dada tarefa varia significativamente com o tipo de codificador de grafo e idioma.

O restante desse artigo está dividido da seguinte forma: na Seção III, é apresentada uma visão geral do estado da arte sobre LLMs para tarefas preditivas (ou de raciocínio) em grafos, e o impacto do idioma no desempenho de LLMs; na Seção II, são apresentados alguns conceitos-chave sobre LLMs e grafos; na Seção IV, são fornecidos os detalhes da metodologia de avaliação deste estudo; na Seção V, são discutidos os resultados alcançados; e finalmente, na Seção VI, este estudo é concluído e são apontados os planos futuros para esse trabalho.

II. Fundamentação Teórica

A. Pre-Trained Large Language Models (LLMs)

Modelos de linguagem (*language models*) [15, 16] são modelos probabilísticos que atribuem probabilidades a sequências de palavras, decompondo a probabilidade de uma sequência no produto das probabilidades dos próximos *tokens*, dados os anteriores. Enquanto os primeiros modelos de linguagem se baseavam principalmente em N-gramas [17], os modelos mais recentes adotam abordagens baseadas em redes neurais [18, 19]. O aumento do poder oferecido pelos modelos de linguagem neural e o aumento no tamanho dos modelos e conjuntos de dados levaram a um novo paradigma de aprendizagem, no qual LLMs são pré-treinados de forma não-supervisionada em grandes quantidades de dados textuais e usados como modelos-base. Para cada aplicação nova, o modelo-base é ajustado com base em pequenas quantidades de dados específicos da tarefa para adaptá-lo a ela.

B. Idiomas e LLMs

Quando é falado de idiomas entendidos por LLMs, pode se citar duas grandes famílias: os *high-resource languages* (HRL), que são os idiomas mais comuns no corpus de treinamento do modelo [20], como inglês, espanhol, chinês, francês; e os *low-resource languages* (LRL) que são os idiomas menos comuns, como dialetos indígenas, línguas africanas, entre outros. Esse estudo é concentrado em analisar quatro HRLs: Inglês (EN); espanhol (ES); português (PT); e chinês simplificado (ZH).

C. Grafos

Grafos não-direcionados são estruturas matemáticas compostas por vértices (ou nós) e arestas (ou ligações), em que as conexões entre os vértices não têm direção. Formalmente, um grafo não-direcionado é representado como uma tupla $G = (V, E)$, onde V é um conjunto de vértices e E é um conjunto de pares não-ordenados de vértices, chamados de arestas. A *matriz de adjacência* de G é denotada por $A \in \{0,1\}^{n \times n}$, ou seja, A_{ij} é igual a 1 se $\{i, j\} \in E$ e zero caso contrário. Foi utilizado D para representar a *matriz diagonal de graus* de G, isto é, $D_{ii} = \sum_j A_{ij}$. Vale ressaltar que, neste trabalho, são considerados grafos sem atributos (*features*) de nós ou arestas.

D. Grafos e LLMs

Fatemi et al. [9] apresentam e avaliam algumas formas de representar grafos como texto. O processo de codificar grafos como texto consiste em uma etapa de codificação dos nós, e outra etapa de codificação das arestas. Os autores apresentarem 9 tipos de codificadores, entre eles: **Adjacência**, que usa codificação de nó inteiro e codificação de aresta entre parênteses; **Incidência**, que usa codificação de nó inteiro e codificação de aresta incidente; **Amizade**, que usa nomes próprios ingleses conhecidos como codificação de nó e codificação de aresta através de relação de amizade; **Coautoria**, que usa nomes próprios ingleses conhecidos como codificação de nó e codificação de aresta de coautoria; **Rede social**, que usa nomes próprios ingleses conhecidos e codificação de aresta de rede social; **Especialista**, que utiliza letras do alfabeto para codificação de nós e setas para codificação de arestas. Na Figura 1, estão destacados os 4 codificadores utilizados nesse trabalho, exemplificados em um grafo com 8 nós.

III. Trabalhos relacionados

Embora, até onde se sabe, não existam trabalhos com a proposta de avaliação do impacto do idioma no contexto de tarefas envolvendo grafos, alguns trabalhos abordaram o impacto do idioma, principalmente as *low-resource languages* (LRL), em outros domínios. Adelani et al. [21] avaliaram o impacto de LRLs brasileiros em responder perguntas sobre a bíblia. Hasan et al. [22] estudaram o impacto de LRLs em tarefas envolvendo análise de sentimentos e linguagem de ódio utilizando o *GPT-4*, *Llama-2* e *GEMINI*. Adebara et al. [23] avaliaram a performance de alguns LLMs em responder tarefas envolvendo a cultura africana utilizando *prompts* traduzidos para idiomas africanos.

A convergência de aprendizado com grafos e raciocínio com LLMs é um campo de pesquisa em rápido crescimento. Quando é falado de trabalhos que buscam usar LLMs para raciocinar sobre tarefas envolvendo grafos, é possível citar duas abordagens principais: abordagens de *hard-prompt* [9, 24], que focam em transformar o grafo em uma representação textual; e as abordagens *soft-prompt* [11, 25], que transformam o grafo diretamente em uma representação vetorial (i.e., não-textual). Ainda nessa linha, também podem ser citados trabalhos que exploram engenharia de *prompt* [12, 26] e trabalhos que buscam criar benchmarks [9, 27, 28] para avaliar essas tarefas.

Um destaque importante vai para o benchmark GraphQA, proposto por Fatemi et al. [9]. Esse benchmark é dedicado a avaliar métodos de codificação de grafos. GraphQA permite a geração de grafos sintéticos com características que reflitam a tarefa escolhida dentre as opções disponíveis. Os autores também avaliaram o impacto de diversos tipos

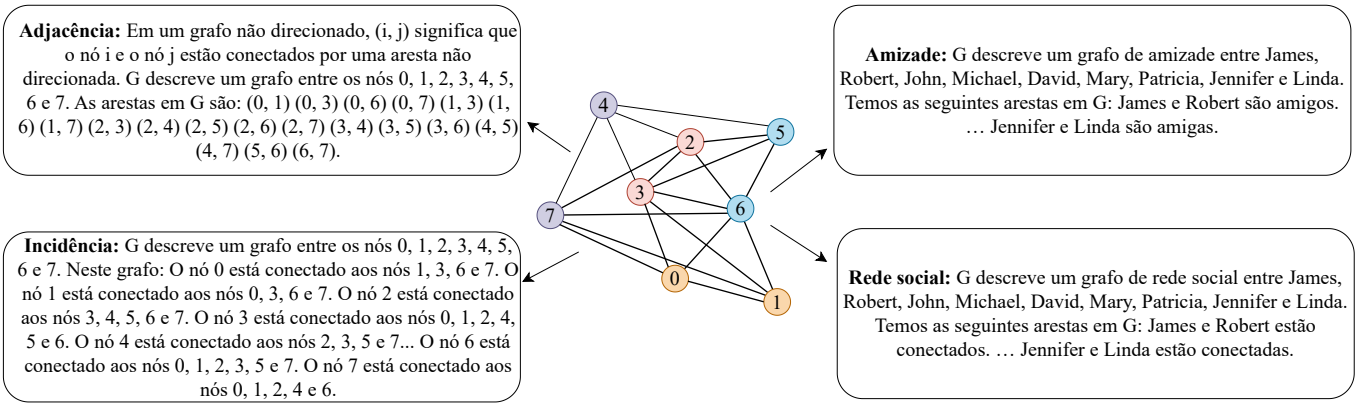

Fig. 1. Visão geral dos métodos de codificação de grafo para texto que foram utilizados nesse trabalho.

de codificadores textuais de grafos e tipos de *prompts* na performance de LLMs.

IV. Metodologia

Nesta seção, é apresentada a metodologia de avaliação empregada neste trabalho. Mais especificamente, são detalhadas as escolhas envolvendo LLMs, idiomas, grafos, tarefas, codificadores, e formas de avaliação.

A. Idiomas avaliados

Foi decidido não incluir LRLs neste estudo porque a literatura já mostrou uma clara tendência de *prompts* com esses idiomas fazerem os LLMs performarem abaixo do esperado. Além disso, os quatro idiomas escolhidos possuem origens, culturas, gramática e tendências linguísticas diferentes entre si (com exceção é claro do português e espanhol), e acredita-se que essas variações são capturadas pelos LLMs.

B. Modelos

Para investigar o impacto do idioma do *prompt* na performance dos LLMs em solucionar tarefas envolvendo grafos, foi considerado dois LLMs representativos. Primeiro, foi considerado o *llama-3.3-70b-versatile* [29], um modelo desenvolvido pela Meta com 70 bilhões de parâmetros pré treinado em aproximadamente 15 trilhões de tokens. A segunda escolha foi o popular *gemma2-9b-it* [30], desenvolvido pela Google com 9 bilhões de parâmetros e pré treinado em 8 trilhões de tokens. Ambos os LLMs estão disponíveis gratuitamente para acesso via API GroqCloud [31]. Aqui, é esperado que os resultados sejam relevantes para modelos com quantidades de parâmetros e arquiteturas variadas.

C. Conjunto de dados

Foi utilizado um conjunto de dados consistindo de 50 grafos gerados sinteticamente por meio do *framework* QraphQA. Foi considerado somente grafos não direcionados gerados por meio do modelo de geração Erdos-Rényi [32]. A quantidade de nós em cada grafo foi determinada de forma aleatória (uniforme) podendo ser qualquer valor entre 5 e 50. Além disso, o parâmetro de esparsidade das arestas também foi definido de forma aleatória (uniforme) entre 0 e 1.

D. Tarefas

Foram consideradas cinco tarefas básicas em nível de grafos que estão presentes no benchmark GraphQA: **Existência de arestas**, que determina se uma aresta específica existe em um grafo; **Grau do nó**, que calcula o grau de um nó específico em um grafo; **Contagem de arestas**, que conta o número de arestas em um grafo; **Nós conectados**, que encontra todos os nós que estão conectados a um determinado nó em um grafo; e **Verificação de ciclo**, que determina se um grafo contém pelo menos um ciclo. Nesse estudo, foi optado por desconsiderar as tarefas *contagem de nó*, pois ela é trivial, e *nós desconectados*, pela similaridade com a tarefa *nós conectados*, já incluída nos experimentos.

E. Métricas de avaliação

As respostas desejadas (alvo) geradas pelo benchmark GraphQA consistem em respostas simples e diretas, como ”sim” ou ”não” para tarefas de existência de alguma característica, números inteiros para tarefas de contagem, e índices dos nós para tarefas de conectividade. Dada essa característica, foi decidido adotar uma métrica de acurácia simples entre a saída do LLM e a resposta desejada. Por exemplo, para a tarefa de ”nós conectados”, se o LLM responder uma sequência hipotética 0,1,2,3,5 e o padrão desejado for 0,1,2,3,4,5, então essa resposta é marcada como erro e é atribuído o valor 0; caso contrário, é contabilizado um acerto (valor 1). Vale ressaltar ainda que as respostas foram verificadas individualmente de forma manual para evitar possíveis erros por conta da variabilidade da saída do LLM.

F. Convertendo grafos para texto

Em relação à codificação de cada grafo em texto, foram analisados quatro tipos de codificação: adjacência (**Adj**), incidência (**Icd**), amizade (**Amz**) e redes sociais (**Rsc**). Cada esquema expressa as conexões entre os nós de forma distinta, variando entre formatos mais técnicos (adjacência e incidência) e descrições mais naturais (amizades e conexões em redes sociais). Embora o trabalho de Fatemi et al. [9] considere formas adicionais de codificação, a escolha para este trabalho foi motivada pelo fato dos outros codificadores

TABELA I

RESULTADOS EXPERIMENTOS VARIANDO O TIPO DE CODIFICADOR E O IDIOMA PARA DIVERSAS TAREFAS ENVOLVENDO GRAFOS SINTÉTICOS DO BENCHMARK GRAPHQA. LLM LLAMA (ESQUERDA) E GEMMA (DIREITA). ACURÁCIA PARA 50 GRAFOS DIFERENTES. EM **NEGRITO** ESTÃO DESTACADOS OS MELHORES RESULTADOS PARA CADA TAREFA/CODIFICADOR.

		Llama					Gemma				
	Idioma	**Existe Ciclo**	**Existe Aresta**	**Grau Nó**	**Total Arestas**	**Nós Conectados**	**Existe Ciclo**	**Existe Aresta**	**Grau Nó**	**Total Arestas**	**Nós Conectados**
Adj	EN	**86,0**	**80,0**	**58,0**	36,0	64,0	**92,0**	76,0	**44,0**	**44,0**	42,0
	PT	**86,0**	74,0	48,0	34,0	66,0	88,0	78,0	40,0	42,0	36,0
	ES	**86,0**	78,0	48,0	32,0	**72,0**	**92,0**	78,0	40,0	40,0	40,0
	ZH	**86,0**	78,0	48,0	**38,0**	60,0	90,0	**78,0**	38,0	40,0	**46,0**
Amz	EN	**92,0**	78,0	**48,0**	20,0	46,0	86,0	**78,0**	28,0	42,0	22,0
	PT	78,0	86,0	38,0	26,0	48,0	84,0	76,0	28,0	**44,0**	22,0
	ES	82,0	80,0	40,0	22,0	50,0	86,0	76,0	26,0	40,0	20,0
	ZH	88,0	**88,0**	40,0	**26,0**	**50,0**	**88,0**	72,0	**30,0**	42,0	**24,0**
Icd	EN	**84,0**	92,0	92,0	**14,0**	90,0	**82,0**	76,0	88,0	8,00	82,0
	PT	82,0	**98,0**	92,0	10,0	**96,0**	**82,0**	76,0	**90,0**	**10,0**	80,0
	ES	**84,0**	96,2	92,0	12,0	92,0	**82,0**	**80,0**	90,0	2,00	76,0
	ZH	82,0	96,0	**94,0**	10,0	88,0	**82,0**	76,0	**90,0**	4,00	**86,0**
Rsc	EN	**90,0**	84,0	**66,0**	26,0	**62,0**	86,0	**80,0**	**24,0**	**42,0**	26,0
	PT	82,0	86,0	56,0	32,0	52,0	86,0	76,0	24,0	42,0	24,0
	ES	**90,0**	86,0	52,0	**34,0**	60,0	86,0	78,0	20,0	42,0	**28,0**
	ZH	86,0	**90,0**	50,0	28,0	54,0	**88,0**	76,0	20,0	44,0	16,0

serem baseados em personagens de filmes/séries (e.g., *game of thrones*) ou artigos científicos. Essa característica poderia servir como um viés favorável ao idioma inglês, uma vez que esses filmes e os artigos são em grande parte escritos em inglês.

G. Prompts utilizados

As descrições dos grafos, as perguntas para as tarefas, e as respostas foram originalmente geradas em inglês pelo framework GraphQA. Inicialmente, para traduzir as descrições dos grafos combinados com a instrução da tarefa, foi utilizado um *prompt* de tradução simples, listado abaixo, e um LLM dedicado para isso — esse *prompt* permitiu ao LLM traduzir o texto de entrada sem responder a pergunta associada a ele.

Prompt de tradução

Agora você é um especialista em tradução de textos descritivos de grafos. Tenha cuidado para que a tradução respeite as particularidades descritivas, culturais e nominais do idioma alvo. Traduza, do inglês para "**idioma alvo**", a seguinte descrição de um grafo e a pergunta associada a ela. Retorne apenas a tradução sem responder a pergunta fornecida:

"Insira a descrição do grafo e a pergunta."

No entanto, durante os experimentos, foi notado que os LLMs retornavam respostas equivalentes quando eram passadas as tarefas traduzidas isoladas ou eram passadas em inglês mais o *prompt* de tradução. Considerando esse cenário, e a possibilidade de poupar recursos, foi decidido manter esse *prompt* agrupado como padrão para os experimentos.

Assim, o *prompt* final para os experimentos ficou como no exemplo a seguir.

Prompt em PT para solucionar a tarefa "total de arestas" com grafo convertido via "Adjacência"

Traduza o texto fornecido do inglês para "**idioma alvo**" e responda a pergunta. Retorne apenas a resposta final.

Texto: "Em um grafo não direcionado, (i,j) significa que o nó i e o nó j estão conectados por uma aresta não direcionada. G descreve um grafo entre os nós 0, 1, 2, 3, 4, 5, 6 e 7. As arestas em G são: (0, 1) (0, 3) (0, 6) (0, 7) (1, 3) (1,6) (1, 7) (2, 3) (2, 4) (2, 5) (2, 6) (2, 7) (3, 4) (3, 5) (3, 6) (4, 5) (4, 7) (5, 6) (6, 7). P. Quantas arestas existem neste grafo?"

V. RESULTADOS E DISCUSSÃO

Na Tabela I pode ser observada a comparação da acurácia do modelo *Llama*, e *Gemma*, entre diferentes idiomas para cada tarefa e codificador analisado. No geral, pode ser vista uma clara diferença de acurácia quando o idioma do *prompt* de entrada é modificado. Por exemplo, para a tarefa "Nós Conectados", pode ser vista uma diferença significativa na acurácia quando o idioma é alterado e mantendo o codificador de "Adjacência" (**Adj**). Surpreendentemente esse resultado é refletido em ambos os LLMs utilizados. Outro resultado notável pode ser observado na tarefa "Grau do Nó" com o codificador de "Rede Social" (**Rsc**), no qual existe uma clara tendência do LLM performar melhor quando utilizado o idioma inglês (**EN**).

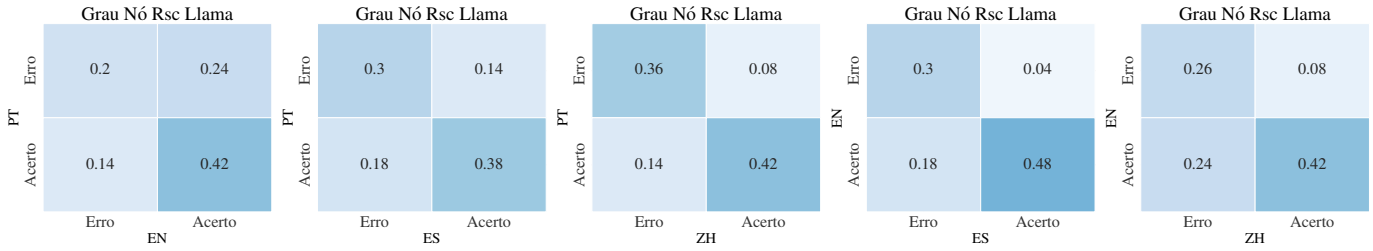

Fig. 2. Distribuição conjunta de probabilidade das predições de pares de modelos: casos de maior discordância em função do idioma.

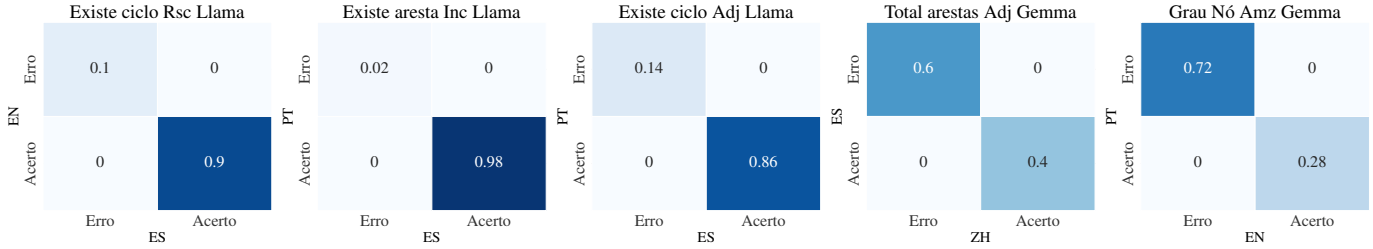

Fig. 3. Distribuição conjunta de probabilidade das predições de pares de modelos: casos de concordância igual a 100%.

Ainda observando a Tabela I, pode ser observado que, no geral, o LLM *Llama* é superior ao LLM *Gemma* em solucionar as tarefas, com exceção da tarefa "Total de Arestas". Também pode ser notado que *prompts* escritos com o idioma inglês tornam os LLMs mais acurados em resolver tarefas de "Existência de ciclo" e "Grau do Nó". Também pode ser notado que o idioma inglês aparece 17 vezes com o melhor resultado (**negritos**), empatado com o chinês, e seguido pelo espanhol com 9 e o português com 7.

TABELA II
ACURÁCIA MÉDIA PARA CADA IDIOMA/CODIFICADOR AO LONGO DAS TAREFAS.

	Cod	EN	PT	ES	ZH
Llama	Adj	**81,0**	77,0	79,0	77,5
	Amz	71,0	69,0	68,5	**73,0**
	Icd	93,0	**94,5**	**94,5**	92,5
	Rsc	**82,0**	68,5	80,5	77,0
Gemma	Adj	**74,5**	71,0	72,5	73,0
	Amz	**64,0**	63,5	62,0	**64,0**
	Icd	84,0	**84,5**	82,5	**84,5**
	Rsc	**64,5**	63,0	63,5	61,0

Na Tabela II estão exibidos os resultados da acurácia média ao longo das tarefas variando o codificador e o idioma. Pode ser visto que para o codificador "Adjacência", o idioma inglês é o que mais se destaca, com 81% para *Llama* e 74,5% para *Gemma*. O mesmo fenômeno ocorre para o codificador "Rede social", com 82% e 64,5% respectivamente. Já o idioma chinês tem maior destaque quando são resolvidas tarefas utilizando o codificador de "Amizade". Também pode ser visto que a maior diferença, em acurácia, devido ao idioma é produzido entre o idioma inglês com o codificador "Rede social" (82%) e o idioma português com o mesmo codificador (68,5%).

De modo geral, dado o impacto significativo quando o idioma do *prompt* é alternado entre HRLs, pode ser acreditado que o impacto seria ainda mais perceptível se fossem considerados os LRLs.

Um detalhe interessante é que mesmo o idioma inglês compondo mais de 89% do corpus de treinamento desses LLMs, um *prompt* com essa língua nem sempre gera a melhor performance. Isso sugere que a capacidade de "raciocínio" dos LLMs é em parte ditada pelo idioma em questão. Tome como exemplo o português e o inglês. Enquanto em português é falada a frase "Eu tenho 30 anos de idade", no inglês é falado "*I'm 30 years old*" ou "Eu sou 30 anos mais velho". Essa diferença entre "ter algo" e "ser algo", que são geradas em frases iguais, mas em idiomas diferentes, pode produzir esse fenômeno dos LLMs raciocinarem melhor para certas tarefas em idiomas X e em outras com idiomas Y.

Ainda na Tabela II, é interessante notar que o codificador "Incidência" (**Icd**) apresenta resultados mais uniformes ao longo dos idiomas em ambos os LLMs. Isso sugere que esse codificador é o que apresenta a menor variabilidade linguística entre os demais. Esse comportamento pode ser associado ao fato deste codificador representar de forma mais acurada como os grafos são descritos nos corpus de treinamento desses LLMs.

Os resultados relacionados à acurácia da Tabela I não torna possível avaliar o grau de concordância entre os diferentes modelos em função do idioma. Para isso, foi proposto observar a distribuição conjunta de probabilidade das predições (correto vs. errado) entre pares de idiomas. As Figuras 2 e 3 ilustram essas distribuições de probabilidade (estimada via máxima verossimilhança) para diferentes combinações de tarefas, codificadores, idiomas e LLMs — observe que essa figura de mérito é semelhante à matriz de confusão, com a diferença de que a predição de um modelo em relação à predição de outro está sendo contrastada, e não em relação à resposta

desejada. Mais especificamente, a Figura 2 ilustra casos de alta discordância em função do idioma. Observe que para a configuração: tarefa "Grau nó", modelo *Llama*, e codificador "Rede Social", as predições provenientes dos idiomas PT e EN concordam em somente 62% dos grafos. Ou seja, o nível de discordância, casos em que uma língua acertou a tarefa mas a outra errou, é de 38%. Chama a atenção o fato de que as cinco maiores taxas de discordância entre as línguas analisadas aconteceram para essa mesma configuração — note que na figura, apenas os idiomas mudam. Houve duas outras combinações que somaram 0,22 de discordâncias, ambas envolvendo o LLM *Gemma* na tarefa "Nós Conectados": uma com o codificador "Adjacência" e a outra, também, com o codificador "Redes Sociais". É interessante notar que, entre as 7 maiores taxas de discordância, seis envolveram o codificador "Redes Sociais".

A Figura 3 ilustra os casos com menor discordância (maior concordância) entre idiomas. Vale ressaltar que somente 29 das 240 combinações atingiram nível de concordância de *1,00*, indicando total concordância entre os idiomas, tanto nos acertos quanto nos erros. A quantidade de acertos e erros compartilhados variou entre as combinações, como ilustrado na Figura 3. Ainda assim, foi observada uma tendência de concordância quando os resultados de cada língua apresentaram mais acertos do que erros: 76% das combinações com taxa de *1,00* tiveram taxas de acerto comum superiores a *0,80*. Isso sugere que, em requisições mais fáceis, a performance é semelhante independentemente da língua utilizada. Apenas a tarefa "Nós Conectados" não apresentou nenhuma combinação com essa taxa de concordância.

Além disso, mais do que uma análise técnica, esta pesquisa, em conjunto com outros trabalhos nessa mesma temática, corroboram com a ideia de que LLMs podem fornecer resultados enviesados em função do idioma, com potencial negativo em relação a idiomas menos representados nos dados utilizados para treinamento dos modelos. Como resultado, o fato desses modelos serem mais suscetíveis a produzirem resultados errôneos com *prompts* em línguas menos comuns pode aumentar ainda mais a distância entre os povos falantes desses idiomas e a ciência de qualidade.

VI. Conclusão

Neste trabalho foi avaliado o impacto do idioma do *prompt* na performance dos LLMs em solucionar tarefas envolvendo grafos. Foi usado o benchmark GraphQA para gerar os grafos, tarefas, e codificadores utilizados no estudo. Os *prompts* foram traduzidos para 4 idiomas e foram avaliados dois LLMs utilizando a acurácia entre a saída real e saída do LLM.

Os resultados mostraram que existe um impacto real do idioma na qualidade da resposta dos LLMs. Esse impacto é mais expressivo quando são utilizados codificadores de grafo menos técnicos, como os baseados em "Amizade" e "Redes sociais". Este estudo também mostrou que esse impacto não depende exclusivamente do LLM, e parece estar fortemente atrelado ao tipo de codificar utilizado, mais especificamente para codificadores que geram representações do grafo mais naturais.

Enquanto esta pesquisa focou em uma análise mais limitada envolvendo um único benchmark, uma direção interessante para trabalhos futuros seria expandir os experimentos para outros benchmarks disponíveis na literatura, analisar tarefas mais complexas como "classificação de nó" e "caminho mais curto", utilizar novos LLMs, possivelmente com menos parâmetros, e expandir a quantidade de idiomas, incluindo *low-resource languages*.

References

[1] R. OpenAI, "Gpt-4 technical report. arxiv 2303.08774," *View in Article*, vol. 2, no. 5, 2023.

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[3] R. Koshkin, K. Sudoh, and S. Nakamura, "Transllama: Llm-based simultaneous translation system," *arXiv preprint arXiv:2402.04636*, 2024.

[4] S. Donthi, M. Spencer, O. Patel, J. Y. Doh, E. Rodan, K. Zhu, and S. O'Brien, "Improving llm abilities in idiomatic translation," in *Future of Information and Communication Conference*. Springer, 2025, pp. 361–375.

[5] R. Zhang and S. Eger, "Llm-based multi-agent poetry generation in non-cooperative environments," *arXiv preprint arXiv:2409.03659*, 2024.

[6] C. Yu, L. Zang, J. Wang, C. Zhuang, and J. Gu, "Charpoet: A chinese classical poetry generation system based on token-free llm," *arXiv preprint arXiv:2401.03512*, 2024.

[7] J. Zhang, X. Wang, W. Ren, L. Jiang, D. Wang, and K. Liu, "Ratt: A thought structure for coherent and correct llm reasoning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 25, 2025, pp. 26 733–26 741.

[8] S. Hao, Y. Gu, H. Luo, T. Liu, X. Shao, X. Wang, S. Xie, H. Ma, A. Samavedhi, Q. Gao *et al.*, "Llm reasoners: New evaluation, library, and analysis of step-by-step reasoning with large language models," *arXiv preprint arXiv:2404.05221*, 2024.

[9] B. Fatemi, J. Halcrow, and B. Perozzi, "Talk like a graph: Encoding graphs for large language models," *arXiv preprint arXiv:2310.04560*, 2023.

[10] H. Sansford, N. Richardson, H. P. Maretic, and J. N. Saada, "Grapheval: A knowledge-graph based llm hallucination evaluation framework," *arXiv preprint arXiv:2407.10793*, 2024.

[11] B. Perozzi, B. Fatemi, D. Zelle, A. Tsitsulin, M. Kazemi, R. Al-Rfou, and J. Halcrow, "Let your graph do the talking: Encoding structured data for llms," *arXiv preprint arXiv:2402.05862*, 2024.

[12] H. Wang, S. Feng, T. He, Z. Tan, X. Han, and Y. Tsvetkov, "Can language models solve graph problems

in natural language?" *Advances in Neural Information Processing Systems*, vol. 36, pp. 30 840–30 861, 2023.

[13] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[14] OpenAI, "Gpt-3 dataset statistics," 2023. [Online]. Available: https://github.com/ openai/gpt-3/tree/master/dataset statistics

[15] R. Rosenfeld, "Two decades of statistical language modeling: Where do we go from here?" *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1270–1278, 2000.

[16] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, vol. 1, no. 2, 2023.

[17] P. Majumder, M. Mitra, and B. Chaudhuri, "N-gram: a language independent approach to ir and nlp," in *International conference on universal knowledge and language*, vol. 2, 2002.

[18] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[20] S. Wu and M. Dredze, "Are all languages created equal in multilingual bert?" *arXiv preprint arXiv:2005.09093*, 2020.

[21] D. I. Adelani, A. S. Doğruöz, A. Coneglian, and A. K. Ojha, "Comparing llm prompting with cross-lingual transfer performance on indigenous and low-resource brazilian languages," *arXiv preprint arXiv:2404.18286*, 2024.

[22] M. A. Hasan, P. Tarannum, K. Dey, I. Razzak, and U. Naseem, "Do large language models speak all languages equally? a comparative study in low-resource settings," *arXiv preprint arXiv:2408.02237*, 2024.

[23] I. Adebara, H. O. Toyin, N. T. Ghebremichael, A. Elmadany, and M. Abdul-Mageed, "Where are we? evaluating llm performance on african languages," *arXiv preprint arXiv:2502.19582*, 2025.

[24] Y. Wen, N. Jain, J. Kirchenbauer, M. Goldblum, J. Geiping, and T. Goldstein, "Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery," *Advances in Neural Information Processing Systems*, vol. 36, pp. 51 008–51 025, 2023.

[25] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," *arXiv preprint arXiv:2101.00190*, 2021.

[26] J. Huang, X. Zhang, Q. Mei, and J. Ma, "Can llms effectively leverage graph structural information: When and why," 2023.

[27] Y. Li, P. Wang, X. Zhu, A. Chen, H. Jiang, D. Cai, V. W. Chan, and J. Li, "Glbench: A comprehensive benchmark for graph with large language models," *Advances in Neural Information Processing Systems*, vol. 37, pp. 42 349–42 368, 2024.

[28] X. Dai, H. Qu, Y. Shen, B. Zhang, Q. Wen, W. Fan, D. Li, J. Tang, and C. Shan, "How do large language models understand graph patterns? a benchmark for graph pattern comprehension," *arXiv preprint arXiv:2410.05298*, 2024.

[29] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, and A. A.-D. et al, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[30] G. Team, "Gemma," 2024. [Online]. Available: https://www.kaggle.com/m/3301

[31] G. Cloud, "Api groq cloud," 2025. [Online]. Available: https://console.groq.com/home

[32] S. Chatterjee and S. S. Varadhan, "The large deviation principle for the erdős-rényi random graph," *European Journal of Combinatorics*, vol. 32, no. 7, pp. 1000–1017, 2011.