Proposta recebida em Outubro 2024 e aceite para publicação em Novembro 2024.

Extração de Informação Aberta com LLM para a Língua Portuguesa

Open Information Extraction with LLM for the Portuguese Language

Bruno Cabral
Universidade Federal da Bahia

Marlo Souza
Universidade Federal da Bahia

Daniela Barreiro Claro
Universidade Federal da Bahia

Resumo

Neste estudo, investigamos a aplicação de Modelos de Linguagem de Grande Escala (LLMs) para Extração de Informação Aberta (EIA) em língua portuguesa. Enquanto a maioria dos métodos de EIA foi desenvolvida visando a língua inglesa, poucos trabalhos na literatura exploram cenários multilíngues e interlinguísticos. Embora haja um crescente interesse em métodos de EIA para o português, o uso de LLMs especificamente focados em EIA nesta língua ainda é pouco explorado. Analisamos a viabilidade de incorporar LLMs abertos e comerciais utilizando engenharia de *prompts* com poucos exemplos para EIA em português. Fornecemos uma análise detalhada do desempenho desses LLMs em tarefas de EIA, demonstrando que eles alcançam métricas de desempenho comparáveis aos sistemas de última geração. Além disso, refinamos e lançamos um LLM aberto para EIA, denominado PortOIE-Llama, que supera os LLMs comerciais em nossos experimentos. Nossos resultados destacam o potencial dos LLMs em tarefas de EIA em português e sugerem que um refinamento e ajuste fino de modelos maiores podem aprimorar ainda mais esses resultados.

Palavras chave

EIA; LLM; extração de informação; corpus

Abstract

In this study, we investigate the application of Large Language Models (LLMs) for Open Information Extraction (OpenIE) in the Portuguese language. While most OpenIE methods have been developed with a focus on the English language, few works in the literature explore multilingual and cross-linguistic scenarios. Although there is a growing interest in OpenIE methods for Portuguese, the use of LLMs specifically focused on OpenIE in this language remains underexplored. We analyze the feasibility of incorporating both open and commercial LLMs using few-shot prompt engineering for OpenIE in Portuguese. We provide a detailed analysis of the performance of these LLMs in OpenIE tasks, demonstrating that they achieve performance metrics comparable to state-of-the-art systems. Additionally, we refine and release an open LLM for OpenIE, named PortOIE-Llama, which outperforms commercial LLMs in our experiments. Our results highlight the potential of LLMs in OpenIE tasks in Portuguese and suggest that further refinement and fine-tuning of larger models can further enhance these outcomes.

Keywords

LLM; OpenIE; information extraction; corpus

1. Introdução

A Extração de Informação Aberta (EIA) é uma tarefa de Processamento de Linguagem Natural (PLN) que identifica estruturas semânticas a partir de dados não estruturados (Etzioni et al., 2008). Os sistemas de EIA extraem fatos em linguagem natural que representam relações semânticas entre entidades.

Desde 2007, com o TEXTRUNNER (Etzioni et al., 2008), vários sistemas de EIA foram propostos para diferentes idiomas. Esses sistemas tiveram diferentes tipos de abordagens, sendo a maioria baseada em abordagens não supervisionadas ou em regras criadas manualmente para identificar relações (Glauber et al., 2018). De acordo com Fader et al. (2011), os sistemas de Extração de Relações Tradicionais geram um modelo capaz de extrair cada relação alvo a partir de exemplos de treinamento anotados. Esses sistemas dependem do domínio de aplicação e sua adaptação a um novo domínio requer extenso trabalho manual. Com o objetivo de superar esses problemas, Etzioni et al. (2008) introduziu o paradigma de extração independente de domínio que emprega padrões generalizados para extrair os relacionamentos entre entidades (Li et al., 2011).

Recentemente, no entanto, impulsionados pelos avanços das redes neurais, além da disponibilidade de novos corpora, métodos de EIA baseados em aprendizado supervisionado foram propostos (Stanovsky et al., 2018; Cui et al., 2018; Sun et al., 2018; Zhang et al., 2017), alcançando novos resultados para a língua inglesa e avançando o estado da arte.

Especificamente nos últimos anos, avanços significativos nos modelos de redes neurais, principalmente modelos gerativos, tais como o GPT-3 (Brown et al., 2020), foram impulsionados pela disponibilidade de dados e pelo poder computacional necessário para processá-los, que foram barateados (Gozalo-Brizuela & Garrido-Merchan, 2023). Progressos notáveis foram observados no Processamento de Linguagem Natural e na geração de imagens digitais. O ChatGPT, por exemplo, alcançou o título de "Aplicativo de Crescimento Mais Rápido de Todos os Tempos" ao acumular 100 milhões de usuários ativos mensais em apenas dois meses.[1]

Embora a maioria dos avanços ocorra na língua inglesa, conforme observado por Glauber et al. (2018), isso pode introduzir um viés em direção a características específicas desse idioma. Como apontado por Bender (2009, 2019), essa ênfase no inglês pode limitar a aplicabilidade dos métodos a outros idiomas. Portanto, é essencial validar os métodos em um conjunto diversificado de idiomas. No entanto, a realização de uma verificação multilíngue é desafiadora devido à escassez de corpora disponíveis para EIA em diferentes línguas. A criação manual desses corpora é complexa (Glauber et al., 2018; Léchelle et al., 2018), principalmente pela falta de uniformização dos conceitos na tarefa de EIA (Xavier et al., 2015; Léchelle et al., 2018; Stanovsky & Dagan, 2016).

Para a língua portuguesa, avanços significativos na EIA ocorreram nos últimos anos Sena & Claro (2019, 2020); Cabral et al. (2022), embora a aplicação de Modelos de Linguagem de Grande Escala (LLMs) ainda seja pouco explorada. Os LLMs demonstraram a capacidade de gerar textos mais próximos aos humanos, indicando um caminho promissor para tarefas de EIA. Este trabalho examina o potencial da utilização de LLMs comerciais e abertos para a tarefa de EIA em português.

Neste trabalho, relatamos nossa experiência na investigação do desempenho de modelos de linguagem de grande escala (LLMs) para a tarefa de EIA em português. Para isso, analisamos o desempenho dos principais modelos abertos e comerciais, e propomos um método de ajuste de LLMs para EIA em português, que consiste em ajustar um modelo de linguagem de grande escala pré-treinado para a tarefa de EIA em português.

[1]ChatGPT estabelece recorde para a base de usuários que mais cresce, https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/, acessado em 5 de novembro de 2023

Além disso, analisamos o uso de ajuste de modelos (fine-tuning) para a tarefa de EIA. Diferentes desafios foram impostos, tais como: a criação de um corpus representativo com tamanho suficiente para a tarefa de EIA em português, a criação de um *prompt* eficaz e a avaliação dos modelos ajustados.

A principal contribuição deste trabalho é a análise, avaliação do desempenho e disponibilização de um LLM ajustado para EIA (Llama-3-PortOIE) e a metodologia de criação de corpus de EIA em português de forma semiautomatizada.

Este artigo está estruturado da seguinte forma: A Seção 2 revisa os trabalhos relacionados; a Seção 3 descreve a metodologia e abordagem empregadas; a Seção 4 apresenta nossos experimentos, resultados e discussões; e a Seção 5 conclui nossos achados e discute direções futuras de pesquisa.

2. Trabalhos Relacionados

A Extração de Informação Aberta tem se tornado um campo de pesquisa cada vez mais relevante na área de Processamento de Linguagem Natural. Com o avanço e a popularização das Redes Neurais em sistemas de EIA (Stanovsky et al., 2018; Cui et al., 2018; Sun et al., 2018; Zhang et al., 2017), novos desafios e oportunidades surgiram, incluindo a necessidade de corpora anotados, mesmo para línguas amplamente estudadas como o inglês (Claro et al., 2019).

Neste contexto de evolução constante, Zhou et al. (2022) realizou uma análise abrangente dos métodos de EIA, identificando duas categorias principais de abordagens: modelos baseados em etiquetagem de sequência (Sequence tagging) e modelos gerativos. Esta categorização fornece uma estrutura útil para compreender o panorama atual da pesquisa em EIA e suas direções futuras.

Os modelos baseados em etiquetagem formulam o problema do EIA como uma tarefa de etiquetagem de sequência (*Sequence tagging*), atribuindo a cada palavra ou unidade linguística (*token*) em uma sentença uma etiqueta que indica sua função (como argumento e predicado) (Zhou et al., 2022). Nesse modelo, cada token na sentença recebe uma marcação específica que identifica seu papel na estrutura da informação extraída. As etiquetas mais comuns são "argumento" (para os elementos sobre os quais se fala) e "predicado" (para a ação ou relação entre os argumentos).

Por exemplo, o predicado

$$Caçou(o\ gato, o\ rato)$$

na sentença "O gato caçõu o rato" pode ser codificado através de etiquetas, tal como representado na Figura 1, utilizando a codificação BIO, na qual a etiqueta 'B-ARG' indica o início de um argumento, 'I-ARG' a continuação de um argumento, e 'PRED' o predicado.

O → B-ARG1
gato → I-ARG1
caçou → PRED
o → B-ARG2
rato → I-ARG2

Figura 1: Exemplo de etiquetagem de tokens em uma sentença, com a notação BIO (Begin-Inside-Outside) em que 'B' representa o início de um fragmento, 'I' indica a continuação do mesmo, e 'O' significa que a palavra não faz parte de um fragmento.

Esses modelos geralmente incorporam uma camada de *embedding* para representações de *tokens*, um *encoder* para representações contextuais de *tokens* como o BERT (Devlin et al., 2018) e um decodificador de etiquetas para a previsão de etiquetas com base na representação do *token* e no esquema de etiquetagem, como um classificador Conditional Random Fields (Lafferty et al., 2001).

Stanovsky et al. (2018) propuseram neurais para, introduzindo uma arquitetura baseada em Redes Neurais Recorrentes (RNN), considerando tal tarefa como uma etiquetagem de sequência semelhante ao Reconhecimento de Entidades Nomeadas. A partir de 2020, vários trabalhos empregaram arquiteturas Transformer diretamente ou em conjunto com o BERT (Devlin et al., 2018). Hohenecker et al. (2020) analisaram várias arquiteturas de EIA baseadas em redes neurais e introduziram um modelo de *embedding* ALBERT (Lan et al., 2020).

Por outro lado, abordagens gerativas para EIA modelam o problema como geração de sequência que produz uma sequência de extrações (Cui et al., 2018). Autores em Cui et al. (2018) e Zhang et al. (2017) exploraram essa abordagem, empregando uma estrutura de codificador-decodificador para aprender argumentos de alta confiança e tuplas de relação inicializadas a partir de um sistema EIA. Estudos contemporâneos integraram embeddings BERT em seus modelos gerativos. Por exemplo, Kolluru et al. (2020a) e Kolluru et al. (2020b) lançaram OpenIE6 e IMoJIE, respectivamente, para a língua inglesa, empregando embeddings BERT e um decodificador LSTM para abordar o problema de extrações redundantes em modelos gerativos de EIA.

Os esforços para desenvolver sistemas EIA interlinguísticos e multilíngues têm sido relativamente escassos. Os autores Zhang et al. (2017) propuseram uma abordagem interlinguística semi-supervisionada, enquanto Multi2OIE (Ro et al., 2020) utilizou M-BERT para *embedding* e extração de predicados, construindo assim extratores para múltiplas línguas, incluindo inglês, português e espanhol.

Os sistemas EIA para a língua portuguesa apresentaram diferentes abordagens ao longo do tempo. Inicialmente, foram desenvolvidos métodos baseados em regras para análise de dependência (Oliveira et al., 2022) e padrões linguisticamente orientados (Sena & Claro, 2019, 2020), que ofereciam uma ampla cobertura de fenômenos linguísticos. Mais recentemente, surgiram aplicações de aprendizado supervisionado com redes neurais profundas, como visto em trabalhos como Multi2OIE (Ro et al., 2020) e PortNOIE (Cabral et al., 2022). Essas abordagens neurais, embora focadas em tarefas mais específicas, demonstraram resultados promissores dentro de seus escopos definidos.

A aplicação de LMs para EIA é uma tendência crescente. Existem usos em diversos campos, como em Resposta Automática a Perguntas (Question Answering) e Extração de Relações e Extração de Informação. Autores em Xu et al. (2023) exploraram a aplicação de LLMs para extração de relações com poucos exemplos (*few-shots*). Oppenlaender & Hämäläinen (2023), por outro lado, investigaram a aplicação de um LLM para resposta a perguntas sobre um corpus de texto em larga escala, obtendo resultados promissores. Wei et al. (2023b) examinaram o uso de sistemas LLM para extração de informação sem exemplos (*zero-shot*), propondo enquadrá-lo como um problema de resposta a perguntas de múltiplas etapas. Kolluru et al. (2022) investigaram o uso de Modelos de Linguagem, nomeadamente BERT e mT5 Xue et al. (2021), para criar um modelo gerativo de EIA em duas etapas, que inicialmente identifica relações e depois monta as extrações para cada relação. Por fim, Cabral et al. (2024) realizaram o ajuste fino utilizando datasets existentes em um LLM em português.

Diferente das abordagens acima, o presente trabalho explora técnicas de criação de datasets de forma semiautomática, e analisa LLMs abertas e comerciais através de engenharia de *prompts* e ajuste de modelos, para avaliar sua viabilidade para EIA em português. Especificamente, analisou-se o impacto do corpus na qualidade do modelo LLM para EIA em Língua Portuguesa.

3. Construção de Corpora para Extração de Informação Aberta

A qualidade dos conjuntos de dados desempenha um papel crucial no treinamento de modelos para tarefas de PLN, especialmente em tarefas complexas como a EIA. No caso de modelos de linguagem, é necessário um volume adequado de dados para que o refinamento (fine-tuning) seja viável (Goodfellow et al., 2016).

A construção manual desses corpora, no entanto, pode ser um processo extremamente lento e oneroso, especialmente para idiomas com menos recursos disponíveis, como o português. Para contornar essas dificuldades, analisamos a criação de corpora sintéticos empregando modelos de linguagem de larga escala e técnicas avançadas de engenharia de prompts.

Aqui, detalhamos a metodologia utilizada para desenvolver um conjunto de dados sintético robusto, destinado ao treinamento de modelos LLM para a tarefa de EIA em língua portuguesa. Todos os códigos utilizados para a criação do corpus e o corpus em si estão disponíveis no GitHub.[2]

3.1. Definição de EIA

Seja $X = (x_1, x_2, \ldots, x_n)$ uma sequência ordenada de tokens x_i que compõem uma sentença. Um extrator de triplas EIA é uma função f que mapeia X para um conjunto $Y = \{y_1, y_2, \ldots, y_j\}$, onde cada elemento y_i é uma tripla $(rel_i, arg_{1_i}, arg_{2_i})$. Cada tripla y_i representa uma relação extraída da sentença X, onde rel_i é o tipo de relação, $arg1_i$ é o primeiro argumento (geralmente o sujeito), e $arg2_i$ é o segundo argumento (geralmente o objeto). Formalmente:

$$f : X \rightarrow Y$$

onde $X = (x_1, x_2, \ldots, x_n)$ e

$$Y = \{(rel_1, arg1_1, arg2_1), \ldots, (rel_j, arg1_j, arg2_j)\}$$

Assumimos que as tuplas estão sempre no formato $y = (arg_1, rel, arg_2)$, com arg_1 e arg_2 sendo sintagmas nominais criados a partir dos tokens em X, e rel representando uma relação entre arg_1 e arg_2. Por simplicidade, como é comum na área, não consideramos extrações que consistem em relações n-árias.

3.2. Corpora Existentes

Na busca por corpora em português que não fossem oriundos de traduções, foram identificadas várias fontes relevantes para a pesquisa em EIA. Os corpora descritos a seguir foram escolhidos por serem anotados diretamente em português, oferecendo uma perspectiva autêntica e diversificada dos fenômenos linguísticos presentes na língua.

O principal corpus identificado foi o *PUD 100* (Souza et al., 2024), uma versão refinada do *PUD 200* (Souza et al., 2024), ambos baseados no corpus de Dependências Universais Paralelas (PUD) em português (Nivre et al., 2020). O *PUD 100* consiste em 100 sentenças anotadas com 136 extrações, representando uma variedade de fontes de notícias e conteúdo da Wikipedia. Este conjunto é notável pela sua diversidade e complexidade linguística, características que são fundamentais para a análise e desenvolvimento de modelos de EIA. Os corpora identificados foram:

- **Pragmático**: Derivado do trabalho de Sena & Claro (2020), este corpus é composto por 400 sentenças de notícias anotadas manualmente, resultando em 485 extrações. A natureza pragmática e o contexto específico das notícias adicionam uma camada de relevância e desafio para a tarefa de extração de informação aberta.
- **Gamalho**: Utilizando textos em português, este corpus reúne extrações feitas por cinco diferentes sistemas de EIA, posteriormente validadas por especialistas. Com 103 sentenças e 346 extrações, ele utilizou recursos linguísticos discutidos nos estudos de Del Corro & Gemulla (2013) e Gamallo & Garcia (2015), oferecendo uma rica base para análises.
- **PUD 200**: Similar ao *PUD 100*, mas em uma escala maior, este conjunto inclui 200 sentneças com 337 extrações, abrangendo tanto notícias quanto conteúdo da Wikipedia, extraídas da parte portuguesa do corpus PUD (Souza et al., 2024).

[2] https://github.com/FORMAS

	# Sentenças	# Extrações
Gamalho (Gamallo & Garcia, 2015)	103	346
Pragmático (Sena & Claro, 2020)	400	485
PUD 200 (Souza et al., 2024)	200	337
PUD 100 (Souza et al., 2024)	100	136

Tabela 1: Estatísticas do Corpora

Na Tabela 1 são apresentadas as estatísticas dos corpora utilizados. Cada um oferece uma visão sobre os desafios de processar textos em português, destacando a importância de desenvolver métodos para a tarefa de EIA. A Figura 2 apresenta um exemplo deste corpus, ilustrando a complexidade das extrações e as nuances que os modelos de EIA precisam considerar.

Frase: *Eles vão atuar no sábado, 10 de junho.*

Extração 1:
Arg_0 = *"Eles"* V = *"vão atuar em"*
Arg_1 = *"o sábado"*

Extração 2:
Arg_0 = *"Eles"* V = *"vão atuar em"*
Arg_1 = *"o sábado, 10 de junho"*

Figura 2: Exemplo de frase e suas extrações no corpus

Observe na Extração 2, uma expansão da contração "no" para "em o" e a inclusão da data completa. Essa variação ilustra a complexidade do conjunto de dados e destaca a necessidade de modelos robustos e sofisticados para lidar com tais nuances. Além disso, a segunda extração sugere que o sábado mencionado é especificamente o dia 10 de junho, o que acrescenta uma camada adicional de interpretação temporal que deve ser considerada pelos modelos de EIA.

Essa complexidade evidencia a importância de desenvolver técnicas avançadas capazes de capturar e interpretar corretamente essas variações linguísticas, garantindo uma extração de informação precisa e contextualizada.

3.3. Seleção de Dados

Para a construção do corpus sintético, foram utilizados artigos da Wikipédia (Wikipedia contributors, 2024) em português. A Wikipédia é uma fonte e diversificada de informação, abrangendo uma grande quantidade de tópicos e domínios. Além disso, os artigos da Wikipédia são licenciados sob a Creative Commons Attribution-ShareAlike License[3], o que permite utilizar o conteúdo de forma livre e legal.

A Wikipédia também disponibiliza um arquivo contendo todos os artigos em uma determinada data, conhecido como *dump* da Wikipédia.[4] Esse arquivo contém o texto completo dos artigos, além de metadados adicionais, como categorias, *links* e referências. Para selecionar os artigos, foi utilizado o *dump* em português, e desenvolvemos um módulo para consultar a API da Wikipédia a fim de obter a quantidade de visitas do mês para cada artigo. Com base nesses dados, foram selecionados os 100 artigos mais visitados. Essa abordagem garantiu que os tópicos escolhidos fossem de alto interesse e relevância para o público em geral. Os artigos selecionados abrangem uma variedade de áreas, como ciência, tecnologia, cultura, história e entretenimento, proporcionando uma base diversificada para a construção do corpus sintético.

Para a criação desse corpus, os dados foram obtidos em Fevereiro de 2024, e os artigos mais visitados nesse mês. O artigo mais visitado foi "Facebook" com 1.850.173 visitas, seguido por "Jogo do bicho", "Brasil" e "Akira Toriyama", o que mostra a diversidade de tópicos abordados.

3.4. Pré-Processamento

Os artigos da Wikipédia presentes no *dump* contêm uma variedade de elementos de formatação, como macros, listas e tabelas, na linguagem chamada Wikitext[5]. Esses elementos, específicos da Wikipédia, podem interferir na tarefa de EIA, introduzindo ruído e complexidade desnecessária aos textos. Para mitigar esse problema, foi desenvolvido uma série de etapas para limpeza e processamento. Existem ferramentas para o inglês, como a `wikitextparser`[6], mas não identificamos uma ferramenta equivalente para o português.

As macros presentes no texto, como mostrado no exemplo a seguir, não podem utilizar as expansões em inglês:

```
{{PEPB|Desporto|esporte}} é toda a forma de
praticar [[Exercício físico|atividade
física]]
```

[3] https://creativecommons.org/licenses/by-sa/4.0/
[4] https://dumps.wikimedia.org/
[5] https://en.wikipedia.org/wiki/Help:Wikitext
[6] https://github.com/5j9/wikitextparser

A macro `PEPB` é específica da Wikipédia em português e expande para "**Desporto**(português europeu) ou **esporte**(português brasileiro)". Essas macros precisam ser expandidas para que os textos mantenham seu sentido completo. Para abordar esse problema, foi desenvolvido um sistema específico para expandir esses termos e limpar elementos desnecessários, como links internos.

Outra tarefa necessária para preparar os dados foi remover seções dos artigos onde o texto era de baixa qualidade para a criação desse *corpus*. Foram identificadas e removidas seções que não seriam úteis para a tarefa de EIA, como "Ver também", "Referências", "Bibliografia" e "Ligações externas". Por fim, foram removidas listas e tabelas, que podem interferir na tarefa de EIA. Este processo de limpeza foi essencial para assegurar que o texto resultante fosse adequado para a extração de informação.

3.5. Geração de Dados Sintéticos

Após a preparação inicial dos textos, foi utilizado um Modelo de Linguagem comercial, o OpenAI GPT-4 (OpenAI, 2023), para gerar as anotações necessárias para o treinamento de modelos de EIA. Para isso, foi usado um *prompt* contendo os 10 primeiros exemplos extraídos do corpus PUD-200, que serviu como guia para o modelo identificar e extrair informações semelhantes dos textos limpos. O processo é ilustrado na Figura 3. Após diversas iterações, o *prompt* utilizado pode ser visto no Prompt 1.

O GPT-4 foi então solicitado a selecionar sentenças dos artigos e a criar extrações baseadas nos exemplos mostrados. Este método de poucos exemplos (*few-shot learning*) permitiu que o modelo generalizasse a partir de um pequeno número de instâncias anotadas manualmente para produzir um grande volume de anotações sintéticas coerentes.

O corpus resultante, denominado aqui como **WikiPUD-Portuguese**, consistiu em 4.451 sentenças com um total de 4.458 extrações. Este volume representa um aumento significativo em relação aos corpora existentes, permitindo um treinamento mais robusto e eficaz dos modelos LLM de EIA. A Figura 4 apresenta um exemplo deste conjunto de dados, mostrando uma sentença e suas respectivas extrações.

4. *Pipeline* PT-EIA para LLMs

O processo desenvolvido para implementar o pipeline de EIA utilizando LLMs pode ser visto de forma resumida na Figura 3.

```
Baseado nos seguintes exemplos de
    Extração de Informação Aberta:

S: "Teoricamente,␣um␣casal␣poderia␣
    abrir␣quatro␣contas␣Tesco␣e␣
    ganhar␣3%␣em␣12,000␣-␣360."

Extração 0:
ARG0="um␣casal" V="poderia␣abrir"
    ARG1="quatro␣contas␣Tesco"
Extração 1:
ARG0="um␣casal" V="poderia␣abrir"
    ARG1="quatro␣contas␣Teoricamente"

.... O RESTANTE DOS EXEMPLOS .....

Realize extrações utilizando frases
    em Português do Brasil no texto
    abaixo.
Inclua também as frases sem
    extrações.
Mantenha o formato acima para a
    saída, como por exemplo:
S: "Frase"
Extração X:
ARG0="X" V="X" ARG1="X"

{{ artigo_wikipedia }}
```

Prompt 1: Criação do corpus.

4.1. Seleção de Modelos

Avaliamos Modelos de Linguagem de Grande Escala (LLMs) tanto de código aberto quanto comerciais. Para selecionar os modelos de melhor desempenho na data de redação deste documento (Abril de 2024), utilizamos o Chatbot Arena Leaderboard (Zheng et al., 2023). Os modelos de destaque incluíram o OpenAI GPT-4 (OpenAI, 2023), Anthropic Claude 3 Opus (Anthropic, 2023), e OpenAI GPT-3.5-turbo (Brown et al., 2020), todos sendo modelos comerciais.

O acesso a esses modelos é possível apenas através de uma API REST privada, que possui um custo elevado por chamada. Entretanto, também consideramos importante avaliar o desempenho de modelos de código aberto, que oferecem acesso irrestrito e podem ser operados localmente.

No Chatbot Arena Leaderboard, os modelos de código aberto mais performáticos e com licenças permissivas no momento são o LLaMA 3 (AI@Meta, 2024) e LLaMA 2 (Touvron et al., 2023). Esses modelos são treinados em extensos corpora de texto da internet e são capazes de prever a próxima palavra em uma sentença, o que lhes permite gerar textos com qualidade semelhante à humana a partir de entradas fornecidas.

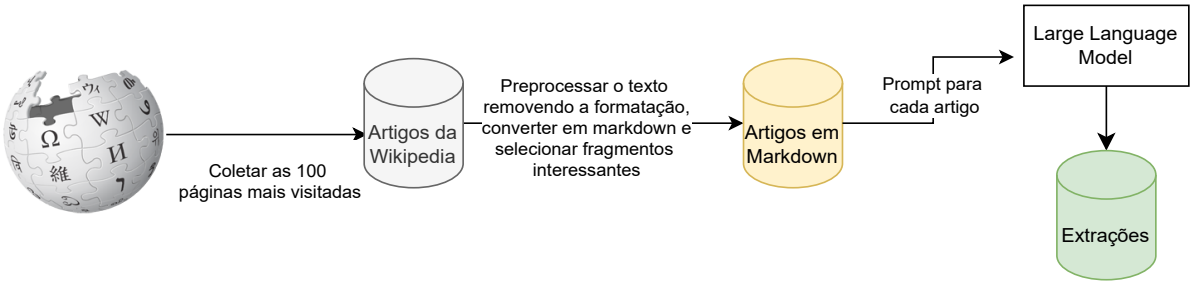


Figura 3: Diagrama ilustrando o processo de limpeza e preparação dos artigos da Wikipédia, seguido pela geração de um corpus sintético utilizando o modelo de linguagem GPT-4.

Frase: *Facebook é uma mídia social e rede social virtual lançada em 4 de fevereiro de 2004, operado e de propriedade privada da Meta, Inc.*

Extração 1:
Arg_0 = "*Facebook*" V = "*é*"
Arg_1 = "*uma mídia social e rede social virtual*"

Extração 2:
Arg_0 = "*Facebook*" V = "*foi lançado em*"
Arg_1 = "*4 de fevereiro de 2004*"

Extração 3:
Arg_0 = "*Facebook*"
V = "*é operado e de propriedade privada de*"
Arg_1 = "*Meta, Inc.*"

Figura 4: Exemplo de frase e suas extrações no corpus WikiPUD-Portuguese.

Por outro lado, os modelos comerciais geralmente alcançam desempenhos superiores devido a diversos fatores, incluindo o ajuste de alinhamento. Este processo visa tornar os modelos consistentes com as expectativas humanas. Por exemplo, o GPT-4 foi treinado utilizando um conjunto de dados baseado em instruções e também foi submetido a uma técnica de *fine-tuning* denominada *Reinforcement Learning from Human Feedback* (RLHF) para melhor alinhamento com as preferências humanas (Ouyang et al., 2022).

Os modelos mais populares à época com suporte ao Português na plataforma HuggingFace[7], como o Sabiá-7b (Pires et al., 2023) e o Lloro 7B[8] foram testados, mas os resultados não foram satisfatórios. Não conseguimos fazer com que o modelo acertasse uma única extração manualmente.

Os modelos LLaMA 3 e LLaMA 2 também possuem versões ajustadas para instruções com RLHF, conhecidas como LLaMA2 chat e LLaMA3 Instruct, respectivamente. A Tabela 2 resume os modelos utilizados neste estudo.

4.2. Ajuste Fino com Supervisão

O Ajuste Fino com Supervisão, do inglês *Supervisioned Finetuning* (SFT), é uma técnica de aprendizado de máquina que consiste em ajustar um modelo pré-treinado em uma tarefa geral para uma tarefa específica, utilizando um conjunto de dados anotado manualmente. Este método é amplamente utilizado para melhorar o desempenho de LLMs em tarefas específicas (Wei et al., 2022), permitindo que eles se adaptem melhor às nuances e requisitos particulares de um domínio ou aplicação específica.

O processo de SFT começa com um modelo que já aprendeu representações generalizáveis durante um extenso pré-treinamento em grandes conjuntos de dados, geralmente com uma variedade de textos gerais. O ajuste fino envolve a continuação do treinamento do modelo em um novo conjunto de dados que é menor, mas altamente relevante para a tarefa específica em questão. Durante o SFT, as camadas do modelo, incluindo seus pesos, são sutilmente ajustadas para minimizar o erro de previsão em relação ao conjunto de dados de destino, frequentemente envolvendo ajustes em parâmetros como a taxa de aprendizado e o número de épocas de treinamento.

O ajuste fino foi realizado utilizando o framework Axolotl.[9] Este método foi realizado em uma máquina com uma placa NVIDIA H100. O SFT oferece uma abordagem direta ao ajuste fino, permitindo atualizações mais precisas e direcionadas do modelo em comparação com métodos não supervisionados.

Utilizamos para o treinamento os *datasets* "Gamalho", "Pragmático", "PUD 200" e o *dataset* sintético descrito na seção anterior, o "WikiPUD-Portuguese". Convertemos as sentenças e extrações do *Dataset* para o formato Alpaca (Taori et al., 2023), que consiste em uma

[7] https://huggingface.co/
[8] https://huggingface.co/semantixai/Lloro
[9] Disponível em https://github.com/OpenAccess-AI-Collective/axolotl

instrução, uma entrada e uma saída. Para a instrução, utilizamos a sentença *"Dada uma sentença S, você faz extrações no formato ARG0, V, ARG1. Realize a extração para a frase abaixo:"*. Como entrada, utilizamos a sentença precedida por "S:", e a saída usa o formato: Extração N: ARG0= "..." V= "..." ARG1= "...".

Fornecemos um exemplo real de uma sentença com sua extração do conjunto de dados PUD 200:

- **instrução**: Dada uma frase S você consegue fazer extrações no formato ARG0, V, ARG1. Realize a extração para a frase abaixo:
- **entrada**: S: "Organismos que vivem em biomas marinhos devem estar adaptados ao sal presente na água."
- **saída**: Extração 0: ARG0 ="Organismos" V="vivem em" ARG1="biomas marinhos"

Para o ajuste fino, operamos o modelo Llama 3-Chat com 8B de parâmetros. As configurações principais para o treinamento incluíram o uso de um otimizador 8 bits AdamW (Loshchilov & Hutter, 2019), um tamanho de **batch** de 8, e um agendador de taxa de aprendizagem cosseno partindo de uma taxa inicial de 0,00002. Utilizamos o maior batch possível para o hardware utilizado. O treinamento também utilizou técnicas de empacotamento de amostras e **checkpointing** de gradientes, o que permitiu uma utilização eficiente da memória e aceleração do treinamento. O modelo está disponível publicamente em HuggingFace.[10]

Nome	Licença	Tamanho
OpenAI GPT-4	Comercial	N/A
Anthropic Claude 3 Opus	Comercial	N/A
OpenAI GPT-3.5	Comercial	N/A
LLaMA-2-Chat	Não comercial	7/70B
LLaMA 3	Não comercial	8/70B

Tabela 2: Resumo dos modelos LLM utilizados

4.3. Engenharia de *Prompt* para Modelos de Linguagem

A engenharia de *prompt* é essencial quando se utiliza LLMs que não passaram por um processo de ajuste fino para tarefas específicas, como a EIA. Esta técnica consiste em otimizar a formulação do *prompt* para melhorar a execução de uma tarefa específica, fornecendo instruções precisas e exemplos pertinentes (Brown et al., 2020). Essa abordagem tem se mostrado eficaz para aprimorar o desempenho dos LLMs em tarefas de PLN.

4.3.1. Metodologia para Derivação do Prompt

O desenvolvimento do *prompt* ideal para a tarefa de EIA foi um processo iterativo e sistemático. Inicialmente, foram utilizadas as primeiras cinco sentenças do conjunto de dados PUD 200 para ajustar o *prompt*. É importante destacar que essas sentenças foram utilizadas apenas como um referencial para o refinamento do *prompt*, e não como parte da avaliação ou como exemplos em uma configuração de *few-shots*.

De início, foi utilizado um *prompt* básico que instruia ao modelo a realização de extraçção de informação de uma sentenças específica. Esta abordagem inicial, revelou-se ineficaz, indicando a necessidade de instruções mais detalhadas para que o modelo compreendesse adequadamente a tarefa.

Para aprimorar essa compreensão, foi adicionado ao *prompt* um exemplo concreto de extração. Esta modificação resultou em uma melhoria significativa na capacidade do modelo de realizar a tarefa solicitada. Continuamos a refinar o *prompt*, incluindo uma definição explícita do papel do sistema, com a sentença: "Você é um sistema OpenIE extremamente inteligente e preciso...". Este ajuste demonstrou ser vantajoso, inclusive em configurações de um único exemplo (*one-shot*), sugerindo que o modelo responde bem a definições claras de seu papel.

Apesar desses avanços, observamos que a inclusão de uma definição extensiva da tarefa EIA, como a encontrada na Wikipedia, dentro do *prompt*, não melhorou os resultados para as sentenças de validação. Por fim, adotamos uma codificação dos exemplos em um formato de chave-valor com quebras de linha, o que tornou as respostas do modelo menos conversacionais, como "Sim, eu posso fazer uma extração...", e mais estruturadas, facilitando a análise subsequente.

Após várias iterações de ajustes, o *prompt* finalizado foi o seguinte, como mostrado na Figura 5.

Neste *prompt*, [**FRASE**] representa a sentença a ser processada, e [**Exemplos do PUD 200**] são alguns exemplos de extrações de EIA no contexto da sentença. Esta abordagem é conhecida como aprendizado com poucos exemplos (*few-shot learning*), no qual o modelo de linguagem é fornecido com um pequeno número de exemplos de extrações de EIA para facilitar seu entendimento da tarefa (Wang et al., 2020).

[10] https://huggingface.co/bratao/llama7b-finetuned-openie-lora

Você é um sistema muito inteligente e preciso de OpenIE. Dada uma frase S, você consegue realizar extrações no formato ARG0, V, ARG1, como por exemplo:
S: "Maria é Professora de Banco de Dados"
Extração 1:
ARG0 = 'Maria'
V = 'é'
ARG1 = 'Professora de Banco de Dados'
[**Exemplos do PUD 200**]
Realize a extração para a frase abaixo:
S: [**FRASE**]

Figura 5: Prompt final utilizado para extrações de EIA

Também exploramos outros *prompts* e técnicas, como o *prompt* de *Chain of Thought* (Cadeia de Pensamento)(Wei et al., 2023a). No entanto, em nossos experimentos, descobrimos que o *prompt* proposto consistentemente produzia saídas esperadas para os exemplos analisados. Como um trabalho futuro, fica a possibilidade de como explorar outras formas de engenharia de *prompt* em LLMs para o problema de Extração de Informação Aberta.

5. Experimentos

Nesta seção, detalhamos a validação empírica para a extração de triplas utilizando LLMs selecionados, incluindo o novo modelo ajustado, o **LLaMA-3-8B-FT**.

5.1. Configuração do ambiente

Cada etapa do *pipeline* foi implementada em Python, versão 3.10, utilizando as bibliotecas da OpenAI e Anthropic para acessar a API dos LLMs. Para LLMs locais de código aberto, foi utilizado o projeto Llama.cpp (Gerganov, 2023) para carregar os modelos e processar as saídas.

Os experimentos utilizaram como configuração para geração a *temperatura* de 0.2 e *max˙tokens* em 1000. Já os parâmetros *top˙p*, *frequency˙penalty* e *presence˙penalty* foram configurados como 0, assegurando que nenhuma penalidade fosse aplicada a tokens frequentes nas saídas.

Os modelos locais foram executados em um servidor na nuvem equipado com um processador AMD EPYC 7003, 30 vCPUs, uma GPU NVIDIA H100 com 92 GB de VRAM e 200 GB de RAM.

Cada sentença no *Corpus PUD 100* foi tokenizada e submetida a cada LLM junto com o *prompt* especificado na Seção 4.3. Os modelos geraram saídas de texto, das quais foram extraídas as triplas. O **LLaMA-3-7B-FT**, o modelo ajustado, utilizou um *prompt* personalizado idêntico ao usado em seu treinamento; portanto, não foi aplicada a estratégia de *prompt* de poucos exemplos para este modelo.

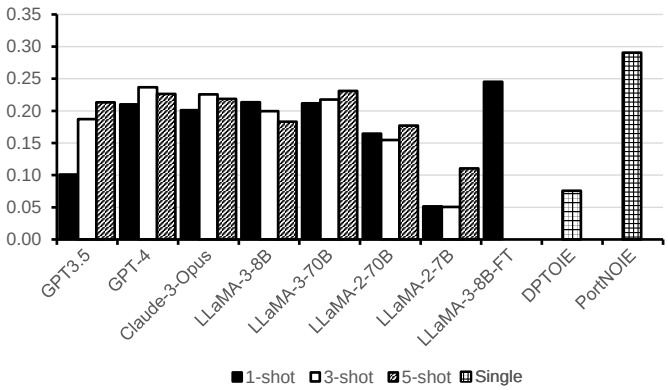
Para a comparação dos resultados com outros sistemas de EIA em português, foram selecionados o DptOIE (Oliveira et al., 2022) e o PortNOIE (Cabral et al., 2022). O DptOIE emprega uma busca em profundidade na árvore de dependência para a extração, enquanto o PortNOIE utiliza uma rede neural profunda e reporta os melhores resultados de F1 para o português.

Foram utilizadas as métricas de precisão (P), recall (R) e F1 para avaliar a qualidade dos extratores. O código de avaliação foi adaptado de Stanovsky et al. (2018), uma metodologia amplamente utilizada em estudos subsequentes (Ro et al., 2020; Kolluru et al., 2020a). Por padrão, esse benchmark emprega um método de pontuação chamado **Correspondência Léxica**, que considera as palavras da tripla como correspondentes se forem pelo menos 50% similares, independentemente da ordem. Também foi utilizada a estratégia de **Correspondência Perfeita**, que considera as strings idênticas após a remoção de pontuação.

Essas métricas comparam as triplas extraídas por cada modelo com as triplas padrão ouro no *Corpus PUD 100*. Uma correspondência exata com a tripla padrão ouro foi considerada uma correspondência. Para a correspondência léxica, foi adotada uma estratégia de correspondência mais relaxada, considerando uma correspondência se pelo menos dois componentes da tripla (arg1, rel, arg2) corresponderem ao padrão ouro.

5.2. Resultados

A análise dos resultados está organizada em duas partes. Primeiramente, são apresentados os escores F1 para correspondências perfeitas e léxica em diferentes modelos, utilizando estratégias de prompt de 1-shot, 3-shot e 5-shot. Em seguida, é realizada uma análise detalhada do desempenho dos modelos empregando sua melhor estratégia de prompt no *Corpus PUD 100*. Os resultados são exibidos na Tabela 3 e na Tabela 4, respectivamente, com uma comparação visual dos escores F1 dos diferentes modelos utilizando a melhor estratégia de prompt.



	Correspondência Perfeita F1 ↑			Correspondência Léxica F1 ↑		
Modelos	**0/1-shot**	**3-shot**	**5-shot**	**0/1-shot**	**3-shot**	**5-shot**
GPT3.5	0.0301	0.1007	0.0955	0.1005	0.1870	0.2132
GPT-4	0.1013	0.1065	0.0978	0.2094	0.2366	0.2262
Claude3-Opus	0.1003	0.1290	0.1125	0.2007	0.2258	0.2185
LLaMA-3-8B	0.0828	0.0828	0.0944	0.2130	0.1994	0.1833
LLaMA-3-70B	0.1114	0.1146	0.1375	0.2111	0.2177	0.2310
LLaMA-2-70B	0.0447	0.0619	0.0655	0.1641	0.1547	0.1770
LLaMA-2-7B	0.0255	0.0144	0.0158	0.0510	0.0505	0.1106
LLaMA-3-8B-FT (PortOIE-Llama3) (0-shot)	**0.1290**	N/A	N/A	**0.2446**	N/A	N/A
DPTOIE (0-shot)	0.0112	N/A	N/A	0.0757	N/A	N/A
PortNOIE (0-shot)	0.0979	N/A	N/A	0.2905	N/A	N/A

Tabela 3: Medidas F1 de Diferentes Modelos para o conjunto de dados PUD100 usando 1, 3 e 5 exemplos para Correspondência Perfeita e Correspondência Léxica.

Model	**Precisão ↑**	**Recall ↑**	**F1 ↑**	**Custo 1k ↓**	**Tempo ↓**
GPT3.5(5-shot)	0.2132	0.2132	0.2132	$1.20	2.7 segs
GPT-4(3-shot)	0.1980	0.2941	0.2366	$36.80	4.2 segs
Claude3-Opus(3-shot)	0.2011	0.2573	0.2258	$20.25	5.2 segs
LLaMA-3-8B (1-shot)	0.1782	0.2647	0.2130	$0.42	1.4 segs
LLaMA-3-70B (3-shot)	0.2269	0.2352	0.2310	$1.11	3.7 segs
LLaMA-2-70B(5-shot)	0.1597	0.1985	0.1770	$1.16	3.8 segs
LLaMA-2-7B(5-shot)	0.1196	0.1029	0.1106	$0.45	1.5 segs
LLaMA-3-8B-FT(PortOIE-Llama3)	0.28	0.2058	0.2446	$0.42	1.4 segs
DPTOIE	0.0408	0.0787	0.0757	$1.62	5.3 segs
PortNOIE	**0.3269**	0.2615	**0.2905**	**$0.15**	**0.5** segs

Tabela 4: Medidas F1 de Diferentes Modelos para o conjunto de dados PUD100 usando a estratégia de solicitação de melhor desempenho para Correspondência Léxica.

Considerando apenas os LLMs no cenário **Correspondência Perfeita**, o modelo LLaMA-2-7B-FT, a versão refinada do modelo LLaMA-2-7B, chamado PortOIE-Llama3, supera outros modelos em todos os cenários com uma pontuação de 0.1271 conforme Tabela 3. O modelo original, o LLaMA-2-7B, apresenta um desempenho consideravelmente pior, com o maior F1 de 0.0255, um aumento de desempenho de 5 vezes. OEste modelo refinado é melhor que o segundo melhor modelo, o modelo comercial GPT-4, com pontuações de 0.1013, 0.1065 e 0.0978.

Considerando o cenário de **Correspondência Perfeita**, o modelo LLaMA-3-8B-FT, nossa versão refinada do modelo LLaMA-3-8B, denominada PortOIE-Llama3, supera outros modelos com uma pontuação F1 de 0.1290 no cenário de 0/1-shot, conforme mostrado na Tabela 3. Este desempenho é notavelmente superior ao do modelo base LLaMA-3-8B, que alcança um F1 máximo de 0.0944 no cenário de 5-shot, demonstrando a eficácia do nosso processo de *fine-tuning*.

Entre os modelos comerciais, o Claude3-Opus apresenta o melhor desempenho em correspondência perfeita, com um F1 de 0.1290 no cenário de 3-shot, superando ligeiramente o GPT-4 (F1 de 0.1065 no 3-shot). O GPT3.5 mostra uma melhoria significativa do cenário 1-shot para o 3-shot, mas seu desempenho é inferior aos modelos mais avançados.

Na **Correspondência Léxica**, o PortOIE-Llama3 mantém sua superioridade com um F1 de 0.2446 no cenário de 0-shot. O LLaMA-3-70B apresenta um desempenho consistente, alcançando seu melhor F1 de 0.2310 no cenário de 5-shot. Entre os modelos comerciais, o GPT-4 lidera com um F1 de 0.2366 no cenário de 3-shot, seguido de perto pelo Claude3-Opus.

É interessante observar que o aumento no número de exemplos nos *prompts* nem sempre resulta em melhor desempenho. Por exemplo, o GPT-4 atinge seu pico no cenário de 3-shot e apresenta uma leve queda no 5-shot, tanto para correspondência perfeita quanto léxica. Isso sugere que a qualidade e relevância dos exemplos podem ser mais importantes do que a quantidade, alinhando-se com achados de outros estudos na literatura, como Fu et al. (2023).

Na análise de desempenho detalhada usando a melhor estratégia de prompt de cada modelo no conjunto de dados PUD100 para Correspondência Léxica (Tabela 4), observamos resultados interessantes:

- O modelo PortNOIE exibe a maior pontuação de precisão (0.3269) e a maior pontuação F1 (0.2905). Além disso, possui o menor custo por 1k tokens ($0.15) e o menor tempo médio de previsão (0.5 segundos), tornando-o o modelo mais eficiente em termos de custo-benefício.
- O PortOIE-Llama3 (LLaMA-3-8B-FT) apresenta o segundo maior F1 (0.2372) entre os LLMs, com um custo significativamente menor ($0.42 por 1k *tokens*) e um tempo de previsão mais curto (1.4 segundos) em comparação com modelos comerciais como GPT-4 e Claude3-Opus.
- O GPT-4 alcança o terceiro maior F1 (0.2366), mas com o custo mais alto ($36.80 por 1k tokens) e um tempo de previsão relativamente longo (4.2 segundos).
- O LLaMA-3-70B mostra um bom equilíbrio entre desempenho (F1 de 0.2310) e eficiência, com um custo moderado ($1.11 por 1k *tokens*) e tempo de previsão razoável (3.7 segundos).

Uma análise dos resultados revela um cenário desafiador para a tarefa de Extração de Informação Aberta em português. Os escores F1 obtidos, mesmo pelos melhores modelos, são objetivamente baixos, com o melhor resultado atingindo apenas 0,2905 (PortNOIE) para correspondência léxica. Esses números indicam que a EIA em português ainda está longe de ser um problema resolvido, possivelmente devido à complexidade inerente da língua portuguesa, com suas nuances sintáticas e semânticas, além das limitações dos modelos atuais em capturar relações semânticas complexas.

Nossos resultados destacam que modelos refinados, como o PortOIE-Llama3, podem superar modelos comerciais de ponta em tarefas específicas, oferecendo uma alternativa eficiente e de baixo custo. Um aspecto particularmente notável é o desempenho superior do PortNOIE, um sistema especializado, sobre os LLMs genéricos. Isto sugere que, no atual estado da tecnologia, abordagens específicas para domínio ainda podem ser mais efetivas que modelos de linguagem grandes e genéricos para tarefas específicas como EIA.

A variação no desempenho entre diferentes estratégias de *prompt* (1-shot, 3-shot, 5-shot) indica que a otimização de *prompts* é crucial e que mais exemplos nem sempre levam a melhores resultados. Isso ressalta a importância de uma seleção cuidadosa e experimentação com diferentes estratégias de *prompt* para cada modelo e tarefa específica.

Por fim, ao considerar a implementação prática desses modelos, é essencial equilibrar o desempenho com fatores como custo e velocidade de inferência. Modelos como PortNOIE e PortOIE-Llama3 oferecem um excelente equilíbrio entre esses fatores, tornando-os opções atraentes para aplicações em larga escala ou com restrições de recursos.

5.3. Análise Qualitativa

Os resultados qualitativos foram avaliados para os modelos de linguagem na tarefa de Extração de Informação Aberta. A sentença de entrada foi: "Teoricamente, um casal poderia abrir quatro contas Tesco e ganhar 3% em £12,000—£360", e as extrações esperadas eram de acordo com o corpus PUD 100 as da Figura 6.

GPT-3.5: Este sistema respondeu que não conseguiu realizar a tarefa, pois não é projetado para Extração de Informação Aberta. Sugeriu o uso de outras ferramentas ou bibliotecas para esse propósito.

Extração 0:
Arg_0 = "*um casal*" V = "*poderia abrir*"
Arg_1 = "*quatro contas Tesco*"
Extração 1:
Arg_0 = "*um casal*" V = "*poderia abrir*"
Arg_1 = "*quatro contas Teoricamente*"

Figura 6: Extrações esperadas de acordo com o corpus PUD 100.

GPT-4: Este sistema teve um bom desempenho, mas extraiu informações adicionais que não faziam parte da saída esperada. A saída foi a apresentada na Figura 7.

Arg_0 = "*um casal*"
V = "*poderia abrir*"
Arg_1 = "*quatro contas Tesco*"
Arg_2 = "*e ganhar*"
Arg_3 = "*3% em £12,000 – £360*"

Figura 7: Extração realizada pelo GPT-4.

Embora tenha identificado corretamente o Arg_0, V e Arg_1 da primeira extração esperada, adicionou informações extras como Arg_2 e Arg_3.

Claude-3-Opus: Este sistema demonstrou uma compreensão mais sofisticada, fornecendo duas extrações relevantes (Figura 8).

Extração 1:
Arg_0 = "*um casal*"
V = "*poderia abrir*"
Arg_1 = "*quatro contas Tesco*"

Extração 2:
Arg_0 = "*um casal*"
V = "*ganhar*"
Arg_1 = "*3% em £12,000 – £360*"

Figura 8: Extrações realizadas pelo Claude-3-Opus.

A primeira extração corresponde exatamente à extração 0 esperada, enquanto a segunda captura informações adicionais relevantes da sentença.

LLaMA-3-8B: Este modelo forneceu múltiplas extrações, demonstrando uma abordagem mais abrangente (Figura 9).

Embora tenha capturado aspectos relevantes, nenhuma das extrações corresponde exatamente às esperadas. A extração 3, em particular, apresenta uma interpretação incorreta.

Extração 1:
Arg_0 = "*um casal*"
V = "*poderia*"
Arg_1 = "*abrir quatro contas Tesco e ganhar 3% em £12,000 – £360*"

Extração 2:
Arg_0 = "*um casal*"
V = "*ganhar*"
Arg_1 = "*3% em £12,000 – £360*"

Extração 3:
Arg_0 = "*Tesco*"
V = "*abrir*"
Arg_1 = "*quatro contas*"

Figura 9: Extrações realizadas pelo LLaMA-3-8B.

LLaMA-3-70B: Este modelo forneceu duas extrações precisas de acordo com a Figura 10.

Extração 1:
Arg_0 = "*um casal*"
V = "*poderia abrir*"
Arg_1 = "*quatro contas Tesco*"

Extração 2:
Arg_0 = "*um casal*"
V = "*ganhar*"
Arg_1 = "*3% em £12,000 – £360*"

Figura 10: Extrações realizadas pelo LLaMA-3-70B.

A primeira extração corresponde exatamente à extração 0 esperada, enquanto a segunda captura colocou como V = "*ganhar*", omitindo o "*poderia*".

LLaMA-2-70B: Este modelo produziu duas extrações, ambas relevantes, mas com algumas diferenças em relação ao esperado (Figura 11)

Extração 1:
Arg_0 = "*um casal*"
V = "*poderia*"
Arg_1 = "*abrir quatro contas Tesco*"

Extração 2:
Arg_0 = "*ganhar*"
V = "*3%*"
Arg_1 = "*£12,000 – £360*"

Figura 11: Extrações realizadas pelo LLaMA-2-70B.

A primeira extração é próxima da esperada, mas separa "*poderia*" e "*abrir.*" A segunda extração apresenta uma interpretação diferente da estrutura da sentença.

LLaMA-2-7B: Este modelo não conseguiu realizar a tarefa, demonstrando uma compreensão limitada do problema e solicitando mais entrada em vez de fornecer uma extração.

LLaMA-3-8B-FT (PortOIE-Llama3): Este modelo ajustado com o corpus proposto para a tarefa, teve um bom desempenho, produzindo extrações muito próximas das esperadas (Figura 12).

Extração 0:
Arg_0 = "*um casal*" V = "*poderia abrir*"
Arg_1 = "*quatro contas Tesco*"
Extração 1:
Arg_0 = "*um casal*" V = "*ganhar*"
Arg_1 = "*3% em £12,000 – £360*"

Figura 12: Extrações realizadas pelo LLaMA-3-8B-FT (PortOIE-Llama3).

A extração 0 corresponde exatamente à esperada, enquanto a extração 1 cometeu o mesmo erro do LLaMA-3-70B, omitindo o "*poderia*".

Em resumo, LLaMA-3-8B-FT, LLaMA-3-70B e Claude-3-Opus demonstraram os melhores desempenhos, produzindo extrações precisas e relevantes. GPT-4 e LLaMA-2-70B também tiveram bom desempenho, mas com algumas inconsistências. LLaMA-3-8B forneceu extrações variadas, mas nem todas precisas. GPT-3.5 e LLaMA-2-7B não conseguiram realizar a tarefa adequadamente. Notavelmente, o ajuste do LLaMA-3-8B com o corpus proposto resultou em uma melhoria significativa no desempenho, fazendo-o superar modelos muito maiores e destacando a importância da adaptação específica para a tarefa.

5.4. Limitações

A primeira limitação de nossa abordagem é o conjunto de dados relativamente compacto, que pode limitar a ampla aplicabilidade de nossas conclusões. Além disso, a qualidade do *prompt* influencia diretamente a eficácia dos LLMs. Como mencionado anteriormente, *prompts* variados podem produzir resultados diversos. Quanto à tarefa, a estrutura de extração de relações binárias que empregamos pode não capturar completamente a complexidade de algumas sentenças. Além disso, o conjunto de dados empregado se baseia apenas em relações binárias. Por último, os LLMs podem estar sujeitos a vieses intrínsecos nos dados de treinamento, potencialmente afetando a qualidade e a justiça das tarefas de EIA.

6. Conclusão e Trabalhos Futuros

Esta pesquisa explorou a eficácia dos Modelos de Linguagem de Grande Escala (LLMs) no contexto da Extração de Informação Aberta (EIA) para o Português. Foram realizados experimentos empregando diversas estratégias de *prompt* e comparando o desempenho de vários modelos, nomeadamente GPT-4, GPT-3.5, LLaMA-3, Claude-3 Opus, LLaMA-2, e o modelo ajustado (*fine-tuned*) proposto neste trabalho, LLaMA-3-8B-FT (PortOIE-Llama3), comparado com outros sistemas de EIA em Português como DptOIE e PortNOIE.

Os resultados revelaram que o LLM ajustado proposto (PortOIE-Llama3) superou consistentemente outros LLMs em pontuações F1 em cenários de correspondência perfeita e léxica, superando inclusive modelos comerciais de ponta como GPT-4 e Claude-3 Opus.

No entanto, apesar das altas pontuações F1 alcançadas pelos LLMs, eles permanecem intensivos em recursos. O PortNOIE demonstrou desempenho superior não apenas em termos de métricas de desempenho, mas também em custo-efetividade e velocidade de previsões, alcançando a maior pontuação de precisão, a maior pontuação F1, o menor custo para 1k previsões e o menor tempo médio de previsão. Isso sugere que, embora LLMs como GPT-4 e LLaMA-3 possam oferecer desempenho notável, um modelo especializado permanece a escolha ótima para EIA em Português.

O modelo LLaMA-3-8B, ajustado para EIA (nosso PortOIE-Llama3), demonstra um potencial significativo para exploração futura. É importante notar que este modelo foi refinado utilizando um conjunto de dados com um número limitado de exemplos em língua portuguesa, partindo do modelo original LLaMA-3, que foi treinado predominantemente com dados em língua inglesa. Considerando essas limitações, é razoável antecipar que um ajuste mais abrangente, empregando um volume maior de dados em português, juntamente com a utilização de um LLM de maior capacidade e melhor adaptado ao idioma português, além de ser especificamente projetado para a tarefa de EIA, poderia resultar em um desempenho superior.

Em conclusão, este trabalho contribui para o entendimento da aplicação de Modelos de Linguagem de Grande Escala para EIA para o Português. Os achados desta pesquisa têm implicações práticas para a criação de sistemas de EIA eficientes em relação ao custo para o Português. Pesquisas futuras poderiam explorar a

otimização usando várias estratégias de *prompt* e avaliar o desempenho desses modelos em conjuntos de dados maiores e mais diversos. Além disso, investigar o impacto de diferentes técnicas de ajuste fino e a incorporação de conhecimento específico do domínio nos LLMs para melhorar ainda mais o desempenho em tarefas de EIA seria uma direção promissora.

Este modelo está disponível publicamente no HuggingFace[11]. Os dados e o código estão disponíveis no GitHub[12].

Referências

AI@Meta. 2024. Llama 3 model card. Github.

Anthropic. 2023. Meet claude. Accessed: 2023-04-03.

Bender, Emily. 2019. English isn't generic for language, despite what NLP papers might lead you to believe [slides]. Em *Symposium on Data Science and Statistics (SDSS)*,

Bender, Emily M. 2009. Linguistically naïve != language independent: Why NLP needs linguistic typology. Em *Workshop on the Interaction between Linguistics and Computational Linguistics: Virtuous, Vicious or Vacuous?*, 26–32.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever & Dario Amodei. 2020. Language models are few-shot learners. arXiv [cs.CL]. 10.48550/arXiv.2005.14165

Cabral, Bruno, Daniela Claro & Marlo Souza. 2024. Exploring open information extraction for Portuguese using large language models. Em *16th International Conference on Computational Processing of Portuguese (PROPOR)*, 127–136.

Cabral, Bruno, Marlo Souza & Daniela Barreiro Claro. 2022. PortNOIE: A neural framework for open information extraction for the Portuguese language. Em *International Conference on Computational Processing of the Portuguese Language (PROPOR)*, 243–255. 10.1007/978-3-030-98305-5_23

Claro, Daniela, Marlo Souza, Clarissa Castellã Xavier & Leandro Oliveira. 2019. Multilingual open information extraction: Challenges and opportunities. *Information* 10(7). 228. 10.3390/info10070228

Cui, Lei, Furu Wei & Ming Zhou. 2018. Neural open information extraction. arXiv [cs.CL]. 10.48550/arXiv.1805.04270

Del Corro, Luciano & Rainer Gemulla. 2013. ClausIE: clause-based open information extraction. Em *22nd International conference on World Wide Web (WWW)*, 355–366. 10.1145/2488388.2488420

Devlin, Jacob, Ming-Wei Chang, Kenton Lee & Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv [cs.CL]. 10.48550/arXiv.1810.04805

Etzioni, Oren, Michele Banko, Stephen Soderland & Daniel S. Weld. 2008. Open information extraction from the web. *Communications of the ACM* 51(12). 68–74. 10.1145/1409360.1409378

Fader, Anthony, Stephen Soderland & Oren Etzioni. 2011. Identifying relations for open information extraction. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1535–1545.

Fu, Yao, Hao Peng, Ashish Sabharwal, Peter Clark & Tushar Khot. 2023. Complexity-based prompting for multi-step reasoning. arXiv [cs.CL/cs.AI/cs.LG]. 10.48550/arXiv.2210.00720

Gamallo, Pablo & Marcos Garcia. 2015. Multilingual open information extraction. Em *Portuguese Conference on Artificial Intelligence (EPIA)*, 711–722. 10.1007/978-3-319-23485-4_72

Gerganov, Georgi. 2023. LLaMA C++. GitHub repository.

Glauber, Rafael, Leandro Souza de Oliveira, Cleiton Fernando Lima Sena, Daniela Barreiro Claro & Marlo Souza. 2018. Challenges of an annotation task for open information extraction in Portuguese. Em *International Conference on Computational Processing of the Portuguese Language (PROPOR)*, 66–76. 10.1007/978-3-319-99722-3_7

[11] https://huggingface.co/bratao/Llama-PortOIE3
[12] https://github.com/FORMAS/openie_generative

Goodfellow, Ian J., Yoshua Bengio & Aaron Courville. 2016. *Deep learning*. MIT Press

Gozalo-Brizuela, Roberto & Eduardo C. Garrido-Merchan. 2023. ChatGPT is not all you need. a state of the art review of large generative AI models. arXiv [cs.LG/cs.AI]. 10.48550/arXiv.2301.04655

Hohenecker, Patrick, Frank Mtumbuka, Vid Kocijan & Thomas Lukasiewicz. 2020. Systematic comparison of neural architectures and training approaches for open information extraction. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 8554–8565. 10.18653/v1/2020.emnlp-main.690

Kolluru, Keshav, Vaibhav Adlakha, Samarth Aggarwal, Soumen Chakrabarti et al. 2020a. OpenIE6: Iterative grid labeling and coordination analysis for open information extraction. arXiv [cs.CL]. 10.48550/arXiv.2010.03147

Kolluru, Keshav, Samarth Aggarwal, Vipul Rathore, Mausam & Soumen Chakrabarti. 2020b. IMoJIE: Iterative memory-based joint open information extraction. Em *58th Meeting of the Association for Computational Linguistics (ACL)*, 5871–5886. 10.18653/v1/2020.acl-main.521

Kolluru, Keshav, Muqeeth Mohammed, Shubham Mittal, Soumen Chakrabarti & Mausam . 2022. Alignment-augmented consistent translation for multilingual open information extraction. Em *60th Meeting of the Association for Computational Linguistics (ACL)*, 2502–2517. 10.18653/v1/2022.acl-long.179

Lafferty, John D., Andrew McCallum & Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Em *International Conference on Machine Learning*, 282–289.

Lan, Zhenzhong, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma & Radu Soricut. 2020. ALBERT: A Lite BERT for self-supervised learning of language representations. Em *International Conference on Learning Representations*,

Léchelle, William, Fabrizio Gotti & Philippe Langlais. 2018. WiRe57: A fine-grained benchmark for open information extraction. arXiv [cs.CL]. 10.48550/arXiv.1809.08962

Li, Haibo, Danushka Bollegala, Yutaka Matsuo & Mitsuru Ishizuka. 2011. Using graph based method to improve bootstrapping relation extraction. Em *Computational Linguistics and Intelligent Text Processing (CICLing)*, 127–138. 10.1007/978-3-642-19437-5_10

Loshchilov, Ilya & Frank Hutter. 2019. Decoupled weight decay regularization. arXiv [cs.LG]. 10.48550/arXiv.1711.05101

Nivre, Joakim, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajic, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers & Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. Em *12th Language Resources and Evaluation Conference (LREC)*, 4034–4043.

Oliveira, Leandro, Daniela Barreiro Claro & Marlo Souza. 2022. DptOIE: A portuguese open information extraction based on dependency analysis. *Artificial Intelligence Review* 56(7). 7015–7046. 10.1007/s10462-022-10349-4

OpenAI. 2023. GPT-4 technical report. arXiv [cs.CL/cs.AI]. 10.48550/arXiv.2303.08774

Oppenlaender, Jonas & Joonas Hämäläinen. 2023. Mapping the challenges of HCI: An application and evaluation of ChatGPT and GPT-4 for cost-efficient question answering. arXiv [cs.HC/cs.AI]. 10.48550/arXiv.2306.05036

Ouyang, Long, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike & Ryan Lowe. 2022. Training language models to follow instructions with human feedback. arXiv [cs.CL/cs.AI/cs.LG]. 10.48550/arXiv.2203.02155

Pires, Ramon, Hugo Abonizio, Thales Sales Almeida & Rodrigo Nogueira. 2023. Sabiá: Portuguese large language models. Em *Brazilian Conference on Intelligent Systems (BRACIS)*, 226–240. 10.1007/978-3-031-45392-2_15

Ro, Youngbin, Yukyung Lee & Pilsung Kang. 2020. Multi^{2}OIE: Multilingual open information extraction based on multi-head attention with BERT. arXiv [cs.CL/cs.LG]. 10.48550/arXiv.2009.08128

Sena, Cleiton Fernando Lima & Daniela Barreiro Claro. 2019. InferPortOIE: A Portuguese open information extraction system with inferences. *Natural Language Engineering* 25(2). 287–306. 10.1017/S135132491800044X

Sena, Cleiton Fernando Lima & Daniela Barreiro Claro. 2020. PragmaticOIE: A pragmatic open information extraction for Portuguese language. *Knowledge and Information Systems* 62. 3811–3836. 10.1007/s10115-020-01442-7

Souza, Marlo, Bruno Cabral, Laís do Nascimento & Daniela Barreiro Claro. 2024. Challenges on enlarging portuguese resources. Unpublished manuscript

Stanovsky, Gabriel & Ido Dagan. 2016. Creating a large benchmark for open information extraction. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2300–2305. 10.18653/v1/D16-1252

Stanovsky, Gabriel, Julian Michael, Luke Zettlemoyer & Ido Dagan. 2018. Supervised open information extraction. Em *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 885–895. 10.18653/v1/N18-1081

Sun, Mingming, Xu Li, Xin Wang, Miao Fan, Yue Feng & Ping Li. 2018. Logician: a unified end-to-end neural approach for open-domain information extraction. Em *11th ACM International Conference on Web Search and Data Mining*, 556–564. 10.1145/3159652.3159712

Taori, Rohan, Ishaan Gulrajani, Tianhao Zhang, Yves Dubois, Xiang Li, Carlos Guestrin, Percy Liang & Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. Relatório técnico. Stanford Center for Research on Foundation Models.

Touvron, Hugo, Louis Martin & the Llama 2 Team. 2023. Llama 2: Open foundation and fine-tuned chat models. arXiv [cs.CL/cs.AI]. 10.48550/arXiv.2307.09288

Wang, Yaqing, Quanming Yao, James Kwok & Lionel M. Ni. 2020. Generalizing from a few examples: A survey on few-shot learning. arXiv [cs.LG/cs.AI]. 10.48550/arXiv.1904.05046

Wei, Jason, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai & Quoc V. Le. 2022. Finetuned language models are zero-shot learners. arXiv [cs.CL]. 10.48550/arXiv.2109.01652

Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le & Denny Zhou. 2023a. Chain-of-thought prompting elicits reasoning in large language models. arXiv [cs.CL/cs.AI]. 10.48550/arXiv.2201.11903

Wei, Xiang, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, Yong Jiang & Wenjuan Han. 2023b. Zero-shot information extraction via chatting with ChatGPT. arXiv [cs.CL]. 10.48550/arXiv.2302.10205

Wikipedia contributors. 2024. Wikipedia, the free encyclopedia. [Online; accessed 9-June-2024].

Xavier, Clarissa Castellã, Vera Lúcia Strube de Lima & Marlo Souza. 2015. Open information extraction based on lexical semantics. *Journal of the Brazilian Computer Society* 21. 4. 10.1186/s13173-015-0023-2

Xu, Xin, Yuqi Zhu, Xiaohan Wang & Ningyu Zhang. 2023. How to unleash the power of large language models for few-shot relation extraction? arXiv [cs.CL/cs.AI/cs.DB/cs.IR/cs.LG]. 10.48550/arXiv.2305.01555

Xue, Linting, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua & Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. arXiv [cs.CL]. 10.48550/arXiv.2010.11934

Zhang, Sheng, Kevin Duh & Benjamin Van Durme. 2017. MT/IE: Cross-lingual open information extraction with neural sequence-to-sequence models. Em *15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 64–70.

Zheng, Lianmin, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez & Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and chatbot Arena. arXiv [cs.CL/cs.AI]. 10.48550/arXiv.2306.05685

Zhou, Shaowen, Bowen Yu, Aixin Sun, Cheng Long, Jingyang Li, Haiyang Yu, Jian Sun & Yongbin Li. 2022. A survey on neural open information extraction: Current status and future directions. arXiv [cs.CL]. 10.48550/arXiv.2205.11725