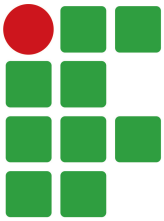

INSTITUTO FEDERAL
Goiás
Câmpus Anápolis

Daniel Vitor Ferreira Silva

Avaliação de Modelos Generativos para Extração de Informações em Processos da Pró-Reitoria de Desenvolvimento Institucional e Recursos Humanos do Instituto Federal de Goiás

Anápolis-GO

2025

Daniel Vitor Ferreira Silva

Avaliação de Modelos Generativos para Extração de Informações em Processos da Pró-Reitoria de Desenvolvimento Institucional e Recursos Humanos do Instituto Federal de Goiás

Projeto de Pesquisa apresentado ao curso de Bacharelado em Ciência da Computação, como requisito para obtenção do grau final na disciplina de Projeto Final de Curso.

Instituto Federal de Goiás - Câmpus Anápolis

Orientador: Prof. Dr. Daniel Xavier de Sousa

Anápolis-GO

2025

Dados Internacionais de Catalogação na Publicação (CIP)

S586a SILVA, Daniel Vitor Ferreira

Avaliação de modelos generativos para extração de informações em processos da pró-reitoria de desenvolvimento institucional e recursos humanos do Instituto Federal de Goiás./ Daniel Vitor Ferreira Silva. – Anápolis: IFG, 2025.

66 f. color.

Orientador: Prof. Mr. Daniel Xavier de Sousa.

Trabalho de Conclusão de Curso – Instituto Federal de Educação, Ciência e Tecnologia de Goiás, Campus Anápolis – Bacharelado em Ciência da Computação, 2025.

1. Modelos de linguagem de grande escala. 2. Extração de informação. 3. Processamento de linguagem natural. 4. Engenharia de *prompt*. 5. Gestão pública.

I. SOUSA, Daniel Xavier de (orient.).

II. Título.

CDD 006.35

Ficha catalográfica elaborada pelo bibliotecário Matheus Rocha Piacenti CRB1/02992
IFG Campus Anápolis

INSTITUTO FEDERAL
Goiás

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
SISTEMA INTEGRADO DE BIBLIOTECAS

TERMO DE AUTORIZAÇÃO PARA DISPONIBILIZAÇÃO NO REPOSITÓRIO DIGITAL DO IFG - ReDi IFG

Com base no disposto na Lei Federal nº 9.610/98, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia de Goiás, a disponibilizar gratuitamente o documento no Repositório Digital (ReDi IFG), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IFG.

Identificação da Produção Técnico-Científica

[ ] Tese	[ ] Artigo Científico
[ ] Dissertação	[ ] Capítulo de Livro
[ ] Monografia – Especialização	[ ] Livro
[X] TCC - Graduação	[ ] Trabalho Apresentado em Evento
[ ] Produto Técnico e Educacional - Tipo: _____	

Nome Completo do Autor: Daniel Vitor Ferreira Silva
Matrícula: 20191060140118
Título do Trabalho: Avaliação de Modelos Generativos para Extração de Informações em Processos da Pró-Reitoria de Desenvolvimento Institucional e Recursos Humanos do Instituto Federal de Goiás.

Autorização - Marque uma das opções

1. (X) Autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso aberto);
2. ( ) Autorizo disponibilizar meu trabalho no Repositório Digital do IFG somente após a data ___/___/_____ (Embargo);
3. ( ) Não autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso restrito).

Ao indicar a opção **2 ou 3**, marque a justificativa:

( ) O documento está sujeito a registro de patente.
( ) O documento pode vir a ser publicado como livro, capítulo de livro ou artigo.
( ) Outra justificativa: _____

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O/A referido/a autor/a declara que:

i. o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
ii. obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia de Goiás os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
iii. cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia de Goiás.

Documento assinado digitalmente
gov.br
DANIEL VITOR FERREIRA SILVA
Data: 09/12/2025 08:22:39-0300
Verifique em https://validar.iti.gov.br

Anápolis, 09/12/2025.
Local Data

_____
Assinatura do Autor e/ou Detentor dos Direitos Autorais

INSTITUTO FEDERAL
Goiás

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE GOIÁS
CÂMPUS ANÁPOLIS

ATA DE DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO

Na presente data realizou-se a sessão pública de defesa do Trabalho de Conclusão de Curso intitulada **Avaliação de Modelos Generativos para Extração de Informações em Processos da Pró-Reitoria de Desenvolvimento Institucional e Recursos Humanos do Instituto Federal de Goiás**, sob orientação de Daniel Xavier de Sousa , apresentada pelo aluno **Daniel Vitor Ferreira Silva (20191060140118)** do Curso **Bacharelado em Ciência da Computação (Câmpus Anápolis)**. Os trabalhos foram iniciados às **15:00** do dia **10/09/2025** pelo Professor presidente da banca examinadora, constituída pelos seguintes membros:

- **Daniel Xavier de Sousa** (Presidente)
- **Douglas Rolins de Santana** (Examinador Interno)
- **Sergio Daniel Carvalho Canuto** (Examinador Interno)

A banca examinadora, tendo terminado a apresentação do conteúdo do Trabalho de Conclusão de Curso, passou à argüição do candidato. Em seguida, os examinadores reuniram-se para avaliação e deram o parecer final sobre o trabalho apresentado pelo aluno, tendo sido atribuído o seguinte resultado:

[X] Aprovado [ ] Reprovado Nota : 9,7

Observação / Apreciações:

O aluno deverá corrigir as observações apontadas pelos avaliadores.

Proclamados os resultados pelo presidente da banca examinadora, foram encerrados os trabalhos e, para constar, eu **Daniel Xavier de Sousa** lavrei a presente ata que assino juntamente com os demais membros da banca examinadora.

Documento assinado eletronicamente por:

- **Daniel Xavier de Sousa** (828.643.731-49), em **10/09/2025 16:49:37** com chave **485794088e7f11f0a95a005056a537a4.**
- **Douglas Rolins de Santana** (018.074.461-58), em **10/09/2025 17:06:52** com chave **b146326a8e8111f098f3005056a537a4.**
- **Sergio Daniel Carvalho Canuto** (020.497.821-10), em **10/09/2025 18:16:51** com chave **77fb61c48e8b11f0a4e4005056a537a4.**

Este documento foi emitido pelo SUAP. Para comprovar sua autenticidade, faça a leitura do QrCode ao lado ou acesse https://suap.ifg.edu.br/comum/autenticar_documento/ e informe os dados a seguir.

Tipo de Documento: Ata de Projeto Final

Data da Emissão: 11/09/2025

Código de Autenticação: 895b8a

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

API	Interface de Programação de Aplicações (*Application Programming Interface*)
ASS	Afastamento para Pós-Graduação Stricto Sensu
CoT	Cadeia de Pensamento (*Chain of Thought*)
CSV	Valores Separados por Vírgula (*Comma-Separated Values*)
EAD	Ensino a Distância
GPT	Transformador Pré-treinado Generativo (*Generative Pre-trained Transformer*)
HTML	Linguagem de Marcação de Hipertexto (*HyperText Markup Language*)
IA	Inteligência Artificial
EI	Extração de Informação (*Information Extraction*)
IFG	Instituto Federal de Goiás
JSON	Notação de Objetos JavaScript (*JavaScript Object Notation*)
KDD	Descoberta de Conhecimento em Bases de Dados (*Knowledge Discovery in Databases*)
LCA	Licença para Cursos de Aprimoramento
LCPG	Licença para Cursos de Pós-Graduação
LLM	Modelo de Linguagem de Grande Escala (*Large Language Model*)
NER	Reconhecimento de Entidades Nomeadas (*Named Entity Recognition*)
PLN	Processamento de Linguagem Natural
PRODIRH	Pró-Reitoria de Desenvolvimento Institucional e Recursos Humanos
RAG	Geração Aumentada por Recuperação (*Retrieval-Augmented Generation*)
SUAP	Sistema Unificado de Administração Pública

Resumo

A eficiência na gestão administrativa é um desafio crescente no setor público, especialmente diante do volume de documentos não estruturados que exigem análise manual. No Instituto Federal de Goiás (IFG), processos como a Licença para Capacitação demandam que servidores dediquem tempo significativo à localização de dados dispersos em anexos e despachos. Este trabalho propõe e avalia uma solução baseada em Modelos de Linguagem de Grande Escala (LLMs) para a Extração de Informação (EI) automática desses processos. A metodologia adotou uma arquitetura de duas etapas (Geração e Validação) e investigou o impacto de quatro estratégias de seleção de texto (Corpo Completo, Requerimento, Segmentação por Sentenças e por Tokens) sobre o desempenho de modelos comerciais (família GPT e DeepSeek) e gratuitos (GPT-OSS, DeepSeek). Os experimentos demonstraram que a estratégia de seleção de texto exerce maior influência na qualidade da extração do que a escolha do modelo. A abordagem de "Corpo Completo"revelou-se a mais robusta, permitindo que modelos capturem relações semânticas complexas perdidas nas estratégias de fragmentação . Os resultados indicam que, embora modelos comerciais como o GPT-4o-mini ofereçam maior consistência geral, modelos gratuitos, quando combinados com a estratégia de contexto amplo, atingem índices de similaridade competitivos (superiores a 0,90 em cenários específicos). O estudo entrega uma sistemática de análise validada para o domínio, demonstrando a viabilidade técnica da automação inteligente na gestão de pessoas do IFG.

Palavras-chave: Modelos de Linguagem de Grande Escala (LLMs). Extração de Informação. Processamento de Linguagem Natural. Engenharia de Prompt. Gestão Pública.

Sumário

1 Introdução

1.1 Contextualização

A gestão administrativa eficiente tem se tornado um desafio cada vez maior nas instituições públicas. No Instituto Federal de Goiás (IFG), essa situação é ainda mais crítica devido ao aumento em volume e complexidade dos processos administrativos com o crescimento de novos câmpus e entrada de novos servidores. Em específico, gestores e servidores da área de Gestão de Pessoas dedicam-se um tempo significativo à revisão de processos e à elaboração de pareceres, com demandas diárias geralmente maiores que sua capacidade de trabalho, consequentemente gerando atrasos operacionais. Nesse cenário, a Inteligência Artificial (IA), especialmente os Modelos Generativos baseados em Large Language Models (LLM), aparecem como uma alternativa estratégica.

Devido ao treinamento com grande volume de dados, os modelos generativos (NG; JORDAN, 2001) emergem como uma solução promissora, capazes de sintetizar informações e oferecer respostas bastante assertivas e contextualizadas. Esses modelos vêm sendo amplamente utilizados para tarefas de Extração de Informação em domínios especializados e para a elaboração de textos jurídicos (BHAMBHORIA et al., 2024), alinhando-se às legislações e normas internas de forma eficiente.

Um exemplo desse tipo de abordagem é a ferramenta *Berna*[1], desenvolvida pelo Tribunal de Justiça do Estado do Ceará (TJCE). A solução realiza uma busca em registros usando linguagem natural, o que tem reduzido o tempo de tramitação ao automatizar a análise de petições iniciais e a movimentação de processos. O sucesso de iniciativas como a ferramenta *Berna* evidencia o potencial transformador da IA no setor público brasileiro.

Destacamos também que o trabalho aqui descrito se apresenta como um dos módulos de uso de inteligência artificial dentro da Plataforma Aurora. Essa plataforma é um projeto da PRODIRH para construção automatizada de pareceres jurídicos, e o módulo descrito nesse trabalho analisa principalmente a etapa de extração de dados, que servirá para futura geração dos pareceres jurídicos.

Por fim, o trabalho está alinhado com as iniciativas nacionais de transformação digital, como a Lei nº 14.129/2021, que promove a modernização dos serviços públicos por meio de soluções tecnológicas. O módulo descrito contribui diretamente para a eficiência administrativa do IFG, fortalecendo a gestão institucional e otimizando os processos de tomada de decisão.

[1] <https://portaladmin.tjce.jus.br/manuais-usuario/index.php/BERNA_-_Busca_Eletr%C3%B4nica_em_Registros_usando_linguagem_Natural>

1.2 Problema de Pesquisa

Motivados por essa nova realidade, diversos trabalhos vêm analisando formas distintas de automatizar atividades normalmente realizadas por seres humanos em repartições públicas (REIS; SANTO; MELãO, 2019). Uma dessas atividades é a extração dos dados contidos em longos documentos e processos.

Por exemplo, em processos de licença para capacitação — um benefício que permite ao servidor público se afastar por até três meses para realizar cursos de desenvolvimento profissional — as informações extremamente relevantes para a análise (como a justificativa de correlação com as atribuições do cargo ou detalhes do conteúdo programático dispersos em anexos) nem sempre se apresentam de forma clara. Frequentemente, esses dados ficam dispersos ou diluídos em meio a diversos outros documentos. Durante o processo de análise, o servidor responsável precisa dedicar um tempo considerável para localizar essas informações, e essa identificação manual representa um desafio significativo que dificulta o trabalho. Tal desafio é acentuado pela escassez de literatura técnica sobre extração de dados em processos de gestão de pessoas, um domínio que, diferentemente de petições jurídicas ou contratos comerciais, ainda carece de abordagens específicas de automação.

1.3 Objetivos

Para mitigar esse problema, este trabalho concentra-se na etapa crítica da Extração de Informações (IE) de processos administrativos, fazendo a aplicação em um caso prático nos processos da Pró-Reitoria de Desenvolvimento Institucional e Recursos Humanos (PRODIRH) do IFG.

1.4 Objetivo Geral

O objetivo principal é avaliar sistematicamente métodos baseados em LLMs para extrair, de forma automática, um conjunto predefinido de informações de documentos processuais não estruturados.

1.5 Objetivos Específicos

Para alcançar o objetivo geral, foram definidos os seguintes objetivos específicos:

- Investigar como diferentes estratégias de seleção de texto e técnicas de engenharia de prompt podem ser combinadas para guiar o modelo e obter a melhor performance, sem a necessidade de um treinamento específico ou ajuste-fino;

- Construir uma base de dados de avaliação específica para o contexto, partindo de processos reais disponibilizados pela PRODIRH e estabelecendo um gabarito através de anotação manual;
- Conduzir uma análise experimental detalhada, comparando quantitativamente o desempenho de diferentes estratégias de texto e modelos de linguagem, identificando as abordagens mais precisas e robustas para a tarefa.

1.6 Organização do Trabalho

O restante deste trabalho está organizado da seguinte forma: os Capítulos 2 e 3 apresentam o referencial teórico e os trabalhos relacionados que fundamentam a pesquisa. O Capítulo 4 detalha a metodologia adotada. O Capítulo 5 expõe e analisa os resultados experimentais, comparando a performance das diferentes estratégias e modelos. Em seguida, o Capítulo 6 discute as limitações da pesquisa. Por fim, o Capítulo 7 apresenta as conclusões finais do trabalho.

2 Referencial teórico

Nesta seção, serão examinados os fundamentos teóricos que sustentam as tecnologias e metodologias abordadas neste trabalho.

2.1 Processamento de Linguagem Natural (PLN)

O Processamento de Linguagem Natural (PLN) é um subcampo da Inteligência Artificial (IA) e Ciência da Computação que tem como objetivo ajudar os computadores a compreender, interpretar e gerar linguagem humana de maneira eficiente. Essa disciplina combina diversas áreas, incluindo aprendizado de máquina, linguística computacional e estatística, para criar sistemas capazes de interagir com os seres humanos de forma natural, seja através de texto ou fala (JURAFSKY; MARTIN, 2023).

No contexto de PLN, o entendimento e a geração de linguagem são dois componentes centrais, frequentemente abordados por arquiteturas baseadas em codificadores (encoders) e decodificadores (decoders), popularizadas pela arquitetura de Transformers (VASWANI et al., 2017).

O processo de entendimento de linguagem, associado à fase de codificação, envolve transformar textos em representações numéricas que capturem informações sintáticas e semânticas, conhecidas como embeddings. O codificador de um Transformer utiliza camadas de atenção para identificar e ponderar as relações entre as palavras. Assim, ele é capaz de "entender"um texto ao mapear suas palavras para vetores de alta dimensionalidade, que representam o significado individual das palavras e os contextos nos quais elas podem aparecer (VASWANI et al., 2017).

A geração de linguagem, por outro lado, é associada ao processo de decodificação, no qual o modelo transforma essas representações numéricas internas de volta em texto. Aqui, o decodificador utiliza mecanismos de atenção que lhe permitem referenciar as saídas do codificador. Isso possibilita que o modelo utilize as informações contextuais geradas durante a codificação para produzir sequências de texto que sejam linguisticamente coerentes e relevantes, assegurando uma geração que reflete o entendimento previamente adquirido (VASWANI et al., 2017).

O campo de PLN abrange uma vasta gama de tarefas que se beneficiam dessas capacidades, tais como, a Sumarização de Textos, Tradução Automática, Análise de Sentimentos e a Resposta a Perguntas(QA). Embora essas aplicações sejam amplas, neste trabalho, o foco de PLN está exclusivamente na tarefa de **Extração de Informação**. A EI visa identificar e extrair dados estruturados (como nomes, datas e valores) de um

texto não estruturado. No nosso caso, o objetivo foi extrair um conjunto predefinido de informações dos processos e organizá-las em um formato JSON padronizado.

2.2 Large Language Models (LLM)

Os Grandes Modelos de Linguagem, ou *Large Language Models (LLMs)*, representam uma evolução significativa no campo do Processamento de Linguagem Natural (PLN). Trata-se de modelos de redes neurais pré-treinados em vastas quantidades de dados textuais, como livros, artigos e páginas da web, com o objetivo de compreender e gerar linguagem humana de forma eficaz (ZHAO et al., 2023). O adjetivo "grande"refere-se tanto ao volume massivo de dados usados no treinamento quanto à enorme quantidade de parâmetros que compõem suas arquiteturas, o que os capacita a lidar com a complexidade da linguagem natural. A Figura 1 ilustra a tendência de crescimento exponencial no número de parâmetros desses modelos ao longo do tempo.

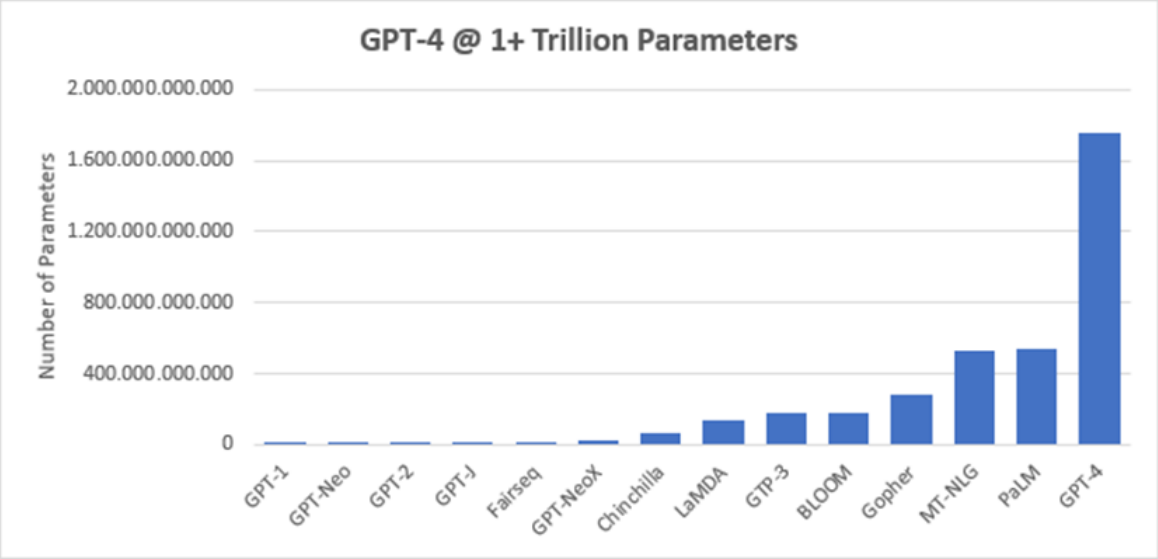

Figura 1 – Crescimento no número de parâmetros dos LLMs ao longo do tempo. Fonte: Siemens[1]

O sucesso dos LLMs é fortemente baseado no aprendizado profundo (deep learning). Durante o pré-treinamento, os modelos são expostos a bilhões de palavras e frases, cobrindo uma ampla gama de tópicos e estilos de escrita. Com essa exposição massiva, eles aprendem a reconhecer padrões linguísticos e a forma como palavras e sentenças se relacionam com base no contexto, sem supervisão humana explícita (SHAKI; KRAUS; WOOLDRIDGE, 2023).

[1] <https://blogs.sw.siemens.com/art-of-the-possible/the-potential-impact-of-llms-on-cae/>

2.2.1 Arquitetura *Transformer* e Mecanismo de Autoatenção

A grande revolução no desempenho dos LLMs deve-se à arquitetura *Transformer* e, especificamente, ao mecanismo de **Autoatenção (*Self-Attention*)** (VASWANI et al., 2017). Diferente de redes recorrentes anteriores (RNNs), que processavam palavras sequencialmente, o *Transformer* processa a sequência inteira de uma vez (paralelismo), permitindo o treinamento em escalas muito maiores.

O mecanismo de Autoatenção é o componente que permite ao modelo ponderar a importância de cada palavra em relação a todas as outras na mesma frase, independentemente da distância entre elas. Matematicamente, para cada token, o modelo calcula três vetores: *Query* (Q), *Key* (K) e *Value* (V). A atenção é calculada como uma soma ponderada dos valores, onde os pesos são determinados pela compatibilidade entre a consulta (Q) de um token e as chaves (K) dos demais (VASWANI et al., 2017):

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Essa capacidade permite que o modelo capture dependências de longo prazo e nuances semânticas complexas — essenciais para tarefas de extração de informação onde o contexto (como um despacho anterior) altera o significado do texto atual.

A tarefa fundamental durante o treinamento, facilitada por essa arquitetura, é prever a próxima palavra em uma sequência de texto. Ao realizar essa tarefa repetidamente em escala, o modelo não apenas memoriza a linguagem, mas aprende a construir representações internas (vetores) que capturam o contexto e os relacionamentos entre as palavras. É essa capacidade de contextualização que permite ao modelo, por exemplo, ao receber o início de uma frase como "O clima hoje está...", prever que palavras como "ensolarado"ou "chuvoso"são continuações prováveis, enquanto termos como "biblioteca"ou "matemática"seriam inadequados.

Um desafio crítico no uso de LLMs é o fenômeno conhecido como alucinação — a produção de informações factualmente incorretas, mas com aparência plausível (HUANG et al., 2023). É importante ressaltar que esse fenômeno não é apenas reflexo de dados de baixa qualidade, mas decorre principalmente da natureza probabilística e generativa desses modelos. Como os LLMs operam prevendo a sequência de palavras estatisticamente mais provável, e não consultando uma base de conhecimento factual verificada, eles podem gerar respostas linguisticamente coerentes e convincentes, porém desconectadas da realidade.

2.2.2 Parâmetros de Geração e Determinismo

Ao utilizar LLMs para tarefas de inferência, o comportamento do modelo não é controlado apenas pelos seus pesos internos, mas também por hiperparâmetros de geração

configurados no momento da inferência. O principal parâmetro de controle da aleatoriedade é a **Temperatura**.

A temperatura é um escalar que modifica a distribuição de probabilidade das próximas palavras (tokens) previstas pelo modelo. Antes de aplicar a função *softmax* (que normaliza as probabilidades), os valores de saída (logits) são divididos pela temperatura T:

- **Temperaturas Altas ($T > 1$):** Aumentam a entropia, tornando a distribuição mais "plana". Isso dá mais chance a palavras menos prováveis serem escolhidas, resultando em textos mais criativos e variados, porém com maior risco de alucinação.
- **Temperaturas Baixas ($T \to 0$):** Acentuam as probabilidades dos tokens mais prováveis. Quando $T = 0$ (ou muito próximo disso), o modelo opera de forma *Greedy* (gulosa), escolhendo sempre o token de maior probabilidade.

Para tarefas de Extração de Informação e estruturação de dados, onde a precisão factual e a consistência são prioritárias sobre a criatividade, recomenda-se o uso de temperaturas baixas para garantir a reprodutibilidade dos resultados (BROWN et al., 2020).

A principal característica desses modelos que é aproveitada neste trabalho é sua capacidade de realizar tarefas complexas a partir de instruções em linguagem natural, sem a necessidade de um novo treinamento. Essa habilidade de generalização permite que um LLM pré-treinado seja aplicado diretamente a problemas específicos, como a Extração de Informação, que é o foco deste trabalho e discutido na seção seguinte.

2.3 Extração de Informação

A Extração de Informação (EI), é a tarefa de converter texto não estruturado, como o encontrado em documentos, em nosso caso, processos do IFG, em dados estruturados e organizados para análise computacional, como um objeto JSON (SARAWAGI, 2008). Historicamente, as abordagens para esta tarefa enfrentavam dois grandes obstáculos práticos. Os sistemas baseados em regras manuais eram precisos, mas extremamente frágeis a variações no texto e de alto custo de manutenção. Por outro lado, os métodos de aprendizado de máquina supervisionado, embora mais flexíveis, exigiam a criação de grandes conjuntos de dados anotados manualmente — um processo lento, caro e que tornava a automação inviável para domínios, como o dos processos administrativos do IFG (NADEAU; SEKINE, 2007).

A capacidade dos LLMs de compreender instruções em linguagem natural permitiu a transição de um paradigma que necessitava de treinamento explícito para um que requer

apenas instruções claras. Essa nova abordagem viabilizou a extração de dados com pouca ou nenhuma amostra de exemplo, um conceito conhecido como few-shot e zero-shot learning (ZHAO et al., 2023), mostrando-se uma alternativa flexível e viável para nosso domínio de processos administrativos.

Neste novo paradigma, a EI é tratada como uma tarefa de geração de saída estruturada. O LLM recebe o texto não estruturado como entrada e, por meio de um comando ou prompt, é instruído a extrair as informações de interesse e a retorná-las em um formato específico, como o JSON. Estudos recentes, como os de (POLAK; MORGAN, 2024), demonstraram com sucesso a extração de dados altamente técnicos de artigos científicos utilizando LLMs de propósito geral guiados apenas por prompts, alcançando altíssima precisão em um domínio complexo e sem um padrão textual rígido.

A eficácia desta nova abordagem de EI depende criticamente da qualidade e da clareza das instruções fornecidas ao modelo, bem como dos exemplos utilizados para guiá-lo. O conjunto de técnicas para a elaboração desses comandos é conhecido como Engenharia de Prompts, que será o foco da próxima seção.

2.4 Engenharia de Prompts

A *Engenharia de Prompts* é o processo de formular e otimizar as instruções fornecidas aos LLMs a fim de maximizar a qualidade e a relevância de suas saídas (MIN et al., 2022). A eficácia desses modelos é significativamente influenciada pela maneira como as solicitações são apresentadas; para maximizar seu potencial, é fundamental estruturar os prompts de maneira que os resultados atendam às expectativas desejadas (SAHOO et al., 2024).

Um prompt é uma instrução textual que orienta um LLM a executar uma tarefa específica sem a necessidade de alterar seus parâmetros internos (SCHULHOFF et al., 2024). Para que os modelos ofereçam respostas que se aproximem do resultado ideal, é essencial que os prompts sejam elaborados com clareza e especificidade. A engenharia de prompt busca otimizar essas instruções, melhorando a capacidade dos LLMs em lidar com tarefas complexas, como a extração de informação estruturada de processos administrativos, que é o foco deste trabalho.

No contexto deste projeto, a engenharia de prompt envolveu a aplicação de diversas estratégias para maximizar a precisão da extração. As principais foram:

- **Claridade e Especificidade:** As instruções devem ser diretas e precisas. Em vez de um comando vago como "Encontre os dados no processo", um **prompt** eficaz define a tarefa de forma precisa: "Extraia as seguintes informações do texto do processo e retorne-as EXATAMENTE no formato JSON especificado."

- **Contextualização:**A contextualização do **prompt** envolve fornecer ao modelo toda a informação necessária para a tarefa. Isso é feito ao inserir diretamente no **prompt** tanto o texto do processo a ser analisado quanto um conjunto fixo de exemplos retirados de outros processos. Esses exemplos contextualizam a tarefa, ensinando o formato e a lógica da extração e garantindo que o modelo opere apenas sobre as informações fornecidas.

A estrutura geral de um prompt utilizado neste trabalho pode ser exemplificada da seguinte forma:

```
template_de_saida = {
    "data_para_realizacao": {
        "inicio": "",
        "fim": ""
    },
    "esta_matriculado": True,
}

Extraia as informações do texto a seguir e retorne-as no formato JSON especificado.

Exemplo 1:
Entrada: [Trecho de texto de um processo similar]
Saída: [Objeto JSON no formato template_de_saida preenchido]

Exemplo 2:
Entrada: [Outro trecho de texto]
Saída: [Outro objeto JSON no formato template_de_saida preenchido]

Texto para análise:
[Conteúdo do processo do IFG a ser analisado]

Instruções Adicionais:
- Retorne APENAS o JSON válido, se algum campo não puder ser preenchido, use valores vazios.
- Extraia a data de início e de fim requerida para a licença para realização de pós-graduação,
e devolva no formato yyyy-mm-dd.

JSON de Saída:
exemplo_de_saida = {
    "data_para_realizacao": {
        "inicio": "2024-05-12",
        "fim": "2024-06-13"
    },
    "esta_matriculado": True,
}

```

Listing 1: Estrutura conceitual de um prompt utilizado para a extração de informação.

Como ilustrado, prompts eficientes comunicam a intenção da tarefa, definem o formato de saída esperado e guiam o modelo com exemplos.

2.4.1 Aprendizado em Contexto(*In-Context Learning*)

In-Context Learning, refere-se à habilidade do modelo de aprender a executar uma nova tarefa em tempo de execução, baseando-se apenas nas informações, instruções e exemplos fornecidos diretamente no prompt, sem que seus pesos internos sejam atualizados (sem a necessidade de retreinamento) (BROWN et al., 2020). Em vez de um processo de treinamento custoso, o "aprendizado"ocorre guiado pelo conteúdo da solicitação em tempo de execução do modelo.Esta abordagem é análoga a ensinar uma tarefa a um ser humano por meio de demonstrações.

Neste trabalho, duas técnicas principais de In-Context Learning foram utilizadas: Zero-Shot e Few-Shot Learning.

2.4.1.1 Zero-Shot Learning

Na abordagem zero-shot, o modelo recebe apenas a instrução da tarefa, sem nenhum exemplo de como executá-la. O sucesso da tarefa depende inteiramente da capacidade do modelo de compreender a instrução e de generalizar a partir de seu conhecimento pré-treinado. Essa técnica é ideal para tarefas mais simples e diretas, onde o onde o objetivo é bem definido (WANG et al., 2022).

No contexto deste trabalho, a aprendizagem zero-shot foi a base para a Etapa 2: Validação Iterativa dos Dados Extraídos 4.3.3.2. Após a extração inicial, o sistema formula prompts simples e diretos para que o modelo valide a si mesmo, como ilustrado na Listagem 2, que mostra a geração dinâmica do prompt para validar uma data de início de uma licença.

```
prompt = f"""
O texto abaixo menciona a data de {date_type} a ser usufruída para a
licença como '{date_value}' (no formato yyyy-mm-dd)?

Considere também datas no formato brasileiro (dd/mm/yyyy) e a
possibilidade da data estar escrita por extenso.

Texto: {text}

Responda com apenas **Sim** ou **Não**. Não adicione explicações.
"""
resposta = get_completion_only_yes_or_no(prompt)
verification_results[f"data_{date_type}_verificada"] = resposta
verificacoes_realizadas.append(resposta.strip().lower() == "sim")
```

Listing 2: Exemplo de código que gera um prompt zero-shot para validação de data de início de uma licença.

Neste caso, não há exemplos; o modelo deve apenas compreender a pergunta e respondê-la com base no texto fornecido.

2.4.1.2 Few-Shot Learning

Na aprendizagem com poucas amostras, ou few-shot learning, o prompt é enriquecido com um pequeno conjunto de exemplos de alta qualidade que demonstram a tarefa sendo executada corretamente. O número de exemplos, ou "shots", é geralmente baixo (ex: 2-shot, 3-shot ou 5-shot), mas suficiente para guiar o modelo a replicar o padrão desejado (LIU et al., 2021). Cada exemplo consiste em um par de entrada e saída esperada. Esses exemplos funcionam como uma ferramenta de condicionamento, guiando o modelo a gerar uma saída que corresponda precisamente ao formato, estrutura e estilo desejados.

Esta foi a principal técnica utilizada na Etapa 1: Extração Inicial via Few-Shot Learning 4.3.3.1. Conforme detalhado na Metodologia, os prompts para os processos dos tipos Licença para Cursos de Aprimoramento, Licença para Cursos de Pós-Graduação e Afastamento para Pós-Graduação *Stricto Sensu* são todos de few-shot learning, contendo múltiplos exemplos de pares "texto de entrada"e "saída JSON". A complexidade e o detalhamento desses prompts podem ser consultados nos apêndices. Por exemplo, o prompt completo para a extração em processos **do tipo** Licença para Cursos de Aprimoramento está disponível no Apêndice 8.1.

2.4.2 Técnicas de Raciocínio (*Reasoning*)

Para tarefas de extração que exigem mais do que a simples localização de texto, é necessário induzir o modelo a realizar um raciocínio mais complexo. As técnicas de

raciocínio na engenharia de prompts são projetadas para decompor um problema, forçando o LLM a "pensar"de forma sequencial e lógica antes de fornecer uma resposta final (WEI et al., 2025).

2.4.3 Cadeia de Pensamento (*Chain-of-Thought* - CoT)

A Cadeia de Pensamento, ou *Chain-of-Thought (CoT)*, é uma técnica de engenharia de prompts que melhora a capacidade dos LLMs em tarefas que exigem raciocínio de múltiplos passos. A abordagem consiste em instruir o modelo a não apresentar a resposta final diretamente, mas a primeiro decompor o problema e gerar uma série de passos intermediários que simulam uma linha de raciocínio lógico (WEI et al., 2023).

Embora a aplicação clássica da CoT envolva fazer o modelo verbalizar seu raciocínio, os princípios de decomposição e raciocínio sequencial podem ser aplicados de forma estrutural na própria arquitetura dos prompts, como foi feito neste trabalho em duas frentes principais, conforme descrito nas subseções seguintes:

2.4.3.1 Raciocínio Implícito nos Exemplos de Extração (*Few-Shot*)

A CoT foi aplicada de forma implícita na Etapa 1: Extração Inicial via *Few-Shot Learning* 4.3.3.1. Os exemplos de few-shot learning foram deliberadamente construídos para ensinar o modelo a realizar inferências. Por exemplo, em uma das demonstrações, o texto de entrada continha apenas a data de início e a duração da licença, mas a saída JSON correspondente apresentava a data de fim já calculada corretamente, como ilustrado na Listagem 3.

```
{
    "input": """
    O presente processo cujo interessado eh o servidor [nome do servidor], ...
    solicita licenca capacitacao de 75 dias, a partir de 15 de fevereiro de 2.024.
    ... Restante do texto ...
    """,
    "output": """
    {
        "nível": "Mestrado",
        "data_para_realizacao": {
            "inicio": "2024-02-15",
            "fim": "2024-04-29"
        },
        "local": {
            "curso": "Performances Culturais",
            "local": "Universidade Federal de Goiás"
        },
        "esta_matriculado": true
    }
    """
}
```

Listing 3: Exemplo de demonstração (*1-shot*) onde o modelo deve inferir a `data_de_fim`.

Ao ser exposto a esses pares, o modelo não aprende apenas a extrair, mas a executar o passo de raciocínio implícito (o cálculo da data) que é necessário para chegar à resposta completa, aplicando essa lógica a novos processos.

2.4.3.2 Raciocínio Guiado nas Instruções de Verificação

O princípio da CoT também foi fundamental na Etapa 2: Validação Iterativa dos Dados Extraídos 4.3.3.2. Em vez de uma pergunta simples, os prompts de validação foram estruturados como uma "lista de verificação", forçando o modelo a seguir uma cadeia de pensamento guiada antes de dar sua resposta. O modelo precisa avaliar o texto sob múltiplos critérios simultaneamente, como ilustrado na Listagem 4, que apresenta o código que gera o prompt para a verificação de um curso.

```
# Trecho do código da função de verificação
prompt_curso = f"""
Analise o texto abaixo e responda APENAS com 'Sim' ou 'Não':

O texto contém TODAS estas informações sobre o curso?
1. Nome do curso: '{extracted_data["local"]["curso"]}'
2. Que será realizado durante o período da licença capacitação
3. Que está vinculado a uma pós-graduação (mestrado/doutorado)

Critérios de rejeição:
- Se o curso mencionado não for de pós-graduação
- Se não estiver claro que será realizado durante a licença
- Se for apenas mencionado de passagem sem relação com a solicitação

Texto: {text}
"""
resposta_curso = get_completion_only_yes_or_no(prompt_curso)
# ... (restante da lógica de verificação)
```

Listing 4: Exemplo de prompt estruturado para forçar o raciocínio guiado na etapa de verificação.

Como ilustrado, o prompt não faz uma pergunta aberta. Ele apresenta ao modelo uma lista de verificação com critérios positivos e negativos. Essa estrutura força o modelo a executar uma cadeia de pensamento implícita e guiada: para responder "Sim", ele precisa validar sequencialmente se cada um dos critérios positivos é atendido e se nenhum dos critérios de rejeição se aplica. Essa abordagem é muito mais robusta do que uma pergunta simples, pois reduz a ambiguidade e aumenta significativamente a confiabilidade da etapa de validação.

2.5 Chunking

O *Chunking*, ou segmentação, refere-se ao processo de dividir grandes volumes de texto em partes menores e mais gerenciáveis, conhecidas como chunks (RAMSHAW; MARCUS, 1995). Essa técnica é uma etapa de pré-processamento essencial no PLN moderno, especialmente ao lidar com os LLMs.

Os LLMs possuem uma capacidade limitada de processamento de texto por vez, definida por sua janela de contexto, que restringe o número de tokens na entrada (DAI et al., 2019). Embora modelos avançados como o *gpt-4o-mini* possuam janelas de contexto amplas, documentos extensos como os processos administrativos do IFG podem exceder esse limite. A segmentação se torna, portanto, indispensável para permitir que o modelo analise o conteúdo completo do documento, processando-o em partes sequenciais que respeitem

suas restrições de entrada (LIU et al., 2023). A Figura 2 ilustra o fluxo simplificado deste processo.

Figura 2 – Fluxograma simplificado do processo de segmentação (*chunking*).

Uma segmentação inadequada representa um risco crítico para a precisão da Extração de Informação, pois pode separar informações intrinsecamente ligadas e, com isso, prejudicar a capacidade do modelo de realizar extrações complexas. Por exemplo, o nome de um curso pode ficar em um chunk, enquanto sua respectiva instituição fica no chunk seguinte, quebrando a ligação semântica que o modelo precisa para realizar uma extração relacional correta. Para mitigar este risco, a técnica padrão é a sobreposição de chunks (overlapping), na qual parte do final de um fragmento é repetida no início do próximo. Essa redundância deliberada garante a continuidade contextual nas fronteiras dos segmentos, reduzindo significativamente a chance de quebras abruptas de informação (BARNETT et al., 2024).

Abordagens podem variar desde a divisão por delimitadores semânticos, como sentenças, até a segmentação por um tamanho fixo de tokens. A escolha da estratégia impacta diretamente a integridade do contexto fornecido ao modelo, um fator que foi investigado experimentalmente neste trabalho.

As vantagens do chunking para a tarefa de extração de informação são:

- **Processamento de Documentos Extensos:** Permite a análise de documentos de qualquer tamanho, superando a limitação da janela de contexto dos LLMs.
- **Viabilização da Análise Sequencial:** Ao dividir o texto em partes, torna-se possível implementar mecanismos eficientes como o *early stopping*, onde o processamento é interrompido assim que todas as informações necessárias são encontradas.
- **Redução de Ruído:** Ao submeter um fragmento de texto menor e mais focado ao modelo, aumenta-se a chance de que ele se concentre nos detalhes relevantes daquele trecho específico, sem a distração de informações de outras partes do documento.

3 Trabalhos relacionados

A literatura recente explora diversas abordagens para especializar e otimizar LLMS para tarefas de extração em domínios complexos. Esta seção revisa os trabalhos mais relevantes, categorizando-os em três eixos metodológicos principais: (1) especialização de modelos via ajuste fino (fine-tuning), (2) otimização para eficiência e execução local, e (3) o uso de intervenção humana (Human-in-the-Loop) para garantir a qualidade.

3.1 Abordagens Baseadas em Ajuste Fino (*Fine-tuning*)

Esta técnica consiste em continuar o treinamento de um modelo pré-treinado em um conjunto de dados menor e focado na tarefa-alvo (HOWARD; RUDER, 2018). O trabalho de (LI et al., 2024) propõe um framework chamado LLM-UIE, que utiliza o ajuste fino para treinar modelos de forma a aprender simultaneamente múltiplas tarefas de extração, como Reconhecimento de Entidades Nomeadas (NER) e Extração de Relações (RE). De forma similar, (DUNN et al., 2022) exploraram o ajuste fino de um modelo da família GPT para extrair receitas de síntese de artigos de ciência dos materiais. Ao treinar o modelo em 1.500 parágrafos anotados, alcançaram um F1-score de 84%, o valida a eficácia do ajuste fino para tarefas de extração estruturada em domínios complexos.

Embora a abordagem de ajuste fino, exemplificada por esses trabalhos, demonstre a capacidade de produzir modelos altamente especializados e precisos para tarefas bem definidas, sua principal desvantagem reside na alta dependência de datasets de treinamento, que são caros e demorados para criar. Este trabalho se diferencia ao evitar essa limitação, focando exclusivamente em métodos de aprendizagem no contexto, que oferecem maior agilidade e menor custo de desenvolvimento.

3.2 Abordagens Focadas em Eficiência e Execução Local

Uma frente de pesquisa de crescente importância, especialmente para aplicações em setores com dados sensíveis como o governamental, foca em otimizar a execução de LLMs para ambientes com recursos computacionais limitados, permitindo sua operação local. A principal vantagem desta abordagem é a privacidade e a segurança dos dados. Ao processar as informações inteiramente no ambiente local, elimina-se a necessidade de enviar dados institucionais sensíveis — como os dos processos de RH do IFG — para APIs de terceiros na nuvem, mitigando riscos de segurança e facilitando a conformidade com leis de proteção de dados, como a Lei Geral de Proteção de Dados (LGPD) no Brasil.

O trabalho de (CUI; GONG; CHENG, 2024) propõe o DLISC, uma abordagem que utiliza técnicas de otimização para viabilizar a extração de informação em LLMs que rodam localmente. A metodologia emprega o ajuste fino de baixa dimensionalidade (Dual-LoRA), que especializa o modelo para a tarefa com um custo computacional muito baixo, e o cache de esquema incremental, que armazena e reutiliza as estruturas de dados para acelerar extrações repetitivas. O objetivo é permitir que mesmo dispositivos com capacidade de processamento modesta possam executar tarefas de extração de forma eficiente.

Contudo, o foco em otimização para execução local geralmente envolve um custo de oportunidade, onde se utilizam modelos menores, o que pode resultar em um sacrifício da performance bruta em comparação com os modelos de ponta disponíveis. Este trabalho, em contraste, adota um objetivo diferente nesta fase da pesquisa: estabelecer uma linha de base da máxima performance de extração possível, utilizando os melhores modelos disponíveis, independentemente de seu custo ou local de execução. A investigação sobre a otimização e a portabilidade do modelo mais eficaz para um ambiente de produção local, como a proposta por (CUI; GONG; CHENG, 2024), constitui-se, portanto, como um importante trabalho futuro.

3.3 Abordagens com Intervenção Humana (*Human-in-the-Loop*)

Uma terceira vertente metodológica, muito alinhada a este trabalho, utiliza a colaboração com um especialista humano — uma abordagem conhecida como Human-in-the-Loop (HITL) — para garantir a qualidade dos dados. Um exemplo prático dessa abordagem na criação de datasets é o estudo de (JEBLICK et al., 2024), que utilizou LLMs de código aberto para extrair dados clínicos de relatórios médicos. A metodologia de criação do dataset de referência foi facilitada pelo HITL, onde o especialista humano valida e corrige as extrações iniciais do modelo para construir o gabarito. Esta abordagem é diretamente análoga à utilizada neste trabalho, na qual a construção do gabarito de 138 processos foi um processo iterativo. Conforme os dados eram extraídos pelos modelos em desenvolvimento, os resultados com scores baixos eram analisados manualmente para diagnosticar a origem do erro, permitindo corrigir inconsistências tanto na anotação do próprio gabarito quanto problemas no texto-fonte. Este ciclo de "anotar-testar-refinar"foi fundamental para garantir a alta qualidade do dataset final.

Avançando nessa linha de colaboração entra humano e inteligência artifical, o trabalho de (DAS et al., 2024) propõe o STRUCTSENSE, um framework para extração de informação. Ele se destaca por utilizar agentes que se autoavaliam, criando um ciclo de feedback iterativo para refinar a extração e, por fim, incorpora a validação da saída por especialistas humanos para garantir a qualidade final. A implementação de agentes

autoavaliativos em STRUCTSENSE apresenta um forte paralelo com a arquitetura de extração e verificação em duas etapas desenvolvida neste trabalho. Ambas as abordagens reconhecem que uma única extração pode ser falível e implementam um passo subsequente para refinar e aumentar a confiabilidade do resultado.

3.4 Conclusão

A literatura aponta para diferentes caminhos na extração de informação com LLMs: o ajuste fino, para alta especialização; a otimização, para eficiência; e os sistemas com intervenção humana, para máxima qualidade e criação de dados. No entanto, uma lacuna persiste na avaliação sistemática de abordagens de aprendizagem no contexto que comparam, de forma controlada, o impacto de diferentes estratégias de seleção de texto e de diferentes classes de LLMs (comerciais vs. gratuitos) para o domínio de processos administrativos do IFG.

É nesta lacuna que este trabalho se insere. Ele se diferencia ao realizar um estudo comparativo sistemático focado na abordagem de in-context learning, contribuindo com: (1) um benchmark de performance entre quatro estratégias de texto distintas; (2) uma análise de custo-benefício entre múltiplos LLMs; e (3) a validação de uma arquitetura de verificação em duas etapas, oferecendo diretrizes práticas para a aplicação desta tecnologia em um domínio de alta relevância para a gestão institucional.

4 Solução para Extração de Dados

Neste capítulo, descrevemos a metodologia utilizada para a extração de dados nos processos de licença para capacitação e afastamento da Pró-Reitoria de Desenvolvimento Institucional e Recursos Humanos (PRODIRH) do IFG. Para melhor descrever essa extração, seguimos as etapas do processo de Descoberta de Conhecimento em Bases de Dados (Knowledge Discovery in Databases - KDD) (CASTRO; FERRARI, 2017). O objetivo do KDD é converter dados brutos em informações, transformar informações em conhecimento e, por fim, traduzir conhecimento em ações práticas, o que nos parece bastante alinhado com os objetivos por nós definidos.

As etapas do KDD são: seleção, pré-processamento, transformação, mineração de dados e avaliação. Nas seções seguintes, detalhamos como essas etapas e como foram aplicadas em nosso projeto, com o objetivo maior de obter maior acurácia na extração das informações.

4.1 Seleção e Descrição da Base de Dados

A seleção é a primeira etapa do KDD, e consiste em definir o conjunto de dados que servirá como base para todo o processo. Nesta seção, descrevemos a origem, a estrutura e o escopo dos dados utilizados, bem como a criação do base de dados de avaliação.

4.1.1 Origem e Estrutura dos Dados Brutos

A base de dados deste estudo é composta por processos administrativos do Instituto Federal de Goiás (IFG), provenientes do Sistema Unificado de Administração Pública (SUAP). O escopo foi delimitado para processos iniciados a partir do ano de 2024, período em que os processos da PRODIRH passaram a tramitar em formato digital, garantindo a completude dos documentos.

Por meio de uma parceria com a PRODIRH, os dados foram disponibilizados em formato de arquivos `.csv`. Foram definidos dois tipos de processo para a análise: Licença para Capacitação e Afastamento para Pós-Graduação *Stricto Sensu*. Uma característica importante que difere esses dois tipos é que a Licença para Capacitação tem um limite de tempo de 90 dias, não sendo o caso para Afastamento para Pós-Graduação *Stricto Sensu*.

Os dados brutos foram organizados em duas estruturas tabulares principais:

- Tabela de Processos: contém os metadados de cada processo, com as seguintes campos:

  - `numero_protocolo`: Identificador único do processo (chave primária).
  - `tipo_de_processo`: A classificação do processo (ex: "Licença para Capacitação"ou "Afastamento para Pós-Graduação *Stricto Sensu*").
  - `subtipo_do_processo`: Detalhamento exclusivo para processos de "Licença para Capacitação", classificando-os em "Licença para Cursos de Aprimoramento (LCA)"ou "Licença para Cursos de Pós-Graduação (LCPG)"[1]. A classificação é feita com base na presença de termos associados à pós-graduação. Segundo os analistas da PRODIRH, o processo é classificado como LCPG se o texto contiver alguma das seguintes expressões:
    * *'elaboração de dissertação de mestrado', 'dissertação de mestrado', 'nível de doutorado', 'tese de doutorado', 'defesa de tese', 'qualificação de mestrado', 'programa de pós-graduação', 'monografia', 'trabalho acadêmico', 'defesa acadêmica', 'nível de pós-graduação stricto sensu'.*

    Caso contrário, o processo é classificado como LCA.
  - `assunto`: Descrição textual breve do objeto do processo.
  - `setor_criacao_nome`: Unidade que iniciou o processo.
  - `setor_atual_nome`: Unidade onde o processo se encontra atualmente.
  - `data_hora_criacao`: Data e hora de abertura do processo.
  - `ultima_modificacao`: Data e hora da última alteração.

- Tabela de Documentos: Cada processo tem associado diversos documentos, todos esses armazenados em "Tabela de Documentos", que armazena o conteúdo textual de cada parte que compõe um processo. De forma geral, os processos são compostos por documentos como o requerimento inicial, despachos de tramitação e pareceres técnicos. Os atributos desta tabela são:
  - `tipo_documento`: A natureza do documento (ex: "Requerimento", "Parecer", "Despacho").
  - `numero_protocolo`: Chave que referencia a Tabela de Processos, indicando a qual processo o documento pertence.
  - `corpo`: O conteúdo textual completo do documento. Este campo é a fonte primária de dados não estruturados para a etapa de extração de informação.

A relação entre as tabelas é de um-para-muitos, onde um único processo com chave primária `numero_protocolo` pode conter múltiplos documentos, cada um representando

[1] A licença para Cursos de Pós-Graduação como tipo de Licença Capacitação, ocorre quando um servidor deseja uma licença de até 90 dias para alguma atividade no curso de Pós-graduação, considerando que esse sevidor não utilize do Afastamento para Pós-Graduacao *Stricto Sensu*.

um trâmite ou uma interação. A extração de informações ocorre primariamente a partir do campo `corpo` dos documentos. O objetivo é identificar e extrair dados específicos que não estão previamente estruturados, tais como: **nome da instituição de ensino**, **nome do curso** e a **data de fim** do afastamento ou licença. A Listagem 5 apresenta um exemplo da estrutura de dados que se expera extrair de um processo de Afastamento *Stricto Sensu*.

```
{
  "local": {
    "curso": "Administração",
    "local": "Faculdade Alves Farias - Alfa"
  },
  "nível": "Mestrado",
  "esta_matriculado": true,
  "data_para_realizacao": {
    "inicio": "2025-01-02",
    "fim": "2026-12-22"
  }
}
```

Listing 5: Exemplo de dados extraídos para Afastamento *Stricto Sensu*.

4.1.2 Criação do Dataset de Avaliação

Para avaliar a performance das estratégias de extração de forma objetiva, foi imprescindível a criação de uma base de dados denominada gabarito. Neste caso, para cada processo foi necessário extrair manualmente as informações de análise, simulando o reconhecimento das informações por um analista.

Iniciamos essa construção fazendo uma seleção e filtragem das amostras de processos. Assim, foram filtrados e removidos os casos que não eram adequados para o treinamento e a validação dos modelos. Os critérios para remoção foram:

- **Processos Vazios:** Registros que, no conjunto de dados fornecido, constavam como processos, mas não continham nenhum documento textual associado.
- **Processos Parciais:** Por exemplo, casos que continham unicamente a justificativa da solicitação, sem o restante dos despachos e demais documentos do trâmite. Essa ausência de peças processuais impede a criação de um gabarito completo com todos os dados-alvo.

Ao final deste processo de limpeza, consolidou-se um *dataset* composto por **138 processos**. Essa amostra é dividida entre **100** processos de "Licença para Capacitação"e **38** de "Afastamento para Pós-Graduação *Stricto Sensu*".

Na sequência, avaliamos individualmente cada processo e extraímos do campo "corpo" os especificos valores que os modelos deve extrair. Os valores foram organizados em uma estrutura de dados do tipo JSON, definindo atributos e valores.

Ao fim deste processo manual, utilizando um formato JSON, nosso gabarito é composto pelas colunas: `numero_protocolo`, `tipo_de_processo` e `gabarito`.

Abaixo mostramos exemplos dos dados extraídos manualmente e seus valores para os respectivos tipos de processo:

- Afastamento para Pós-Graduação *Stricto Sensu*: o gabarito para este tipo de processo foca nas informações sobre o curso e o período do afastamento. Um exemplo da estrutura de dados é apresentado na Listagem 6.

```
{
  "local": {
    "curso": "Educação para Ciências e Matemática",
    "local": "Instituto Federal de Goiás (IFG)"
  },
  "nível": "Doutorado",
  "esta_matriculado": true,
  "data_para_realizacao": {
    "fim": "2026-12-22",
    "inicio": "2024-10-03"
  }
}
```

Listing 6: Exemplo de gabarito para Afastamento para Pós-Graduação *Stricto Sensu*.

- Licença Capacitação para Cursos de Pós-Graduação: similar ao afastamento, mas dentro da modalidade de licença. A estrutura de dados é mostrada na Listagem 7.

```
{
  "local": {
    "curso": "Performances Culturais",
    "local": "Universidade Federal de Goiás"
  },
  "nível": "Mestrado",
  "esta_matriculado": true,
  "data_para_realizacao": {
    "fim": "2025-04-29",
    "inicio": "2024-02-15"
  }
}
```

Listing 7: Exemplo de gabarito para Licença Capacitação para Cursos de Pós-Graduação.

- Licença para Capacitação para Cursos de Aprimoramento: este subtipo permite que o servidor realize um ou mais cursos. A estrutura de dados foi modelada para refletir essa flexibilidade de variação de cursos, como visto na Listagem 8.

```
{
  "locais": [
    {
      "cursos": [
        {
          "nome": "Tradução e Interpretação de Libras",
          "carga_horaria": 280
        },
        {
          "nome": "Tecnologias assistivas para educação especial",
          "carga_horaria": 280
        }
      ],
      "instituicao": "Centro Educacional de Desenvolvimento Profissional - CEDEP"
    }
  ],
  "modalidade_cursos": "EAD",
  "carga_horaria_total": 560,
  "data_para_realizacao": {
    "fim": "2024-12-31",
    "inicio": "2024-10-04"
  }
}
```

Listing 8: Licença para Capacitação para Cursos de Aprimoramento.

4.2 Pré-processamento de Dados

A etapa de Pré-processamento no KDD é fundamental para tratar inconsistências e ruídos que podem comprometer o desempenho dos modelos na fase de mineração. Como a fonte primária de dados é o campo `corpo`, que armazena o conteúdo de documentos, um *pipeline* de limpeza foi aplicado a todos os dados.

4.2.1 Conversão de HTML para Texto Puro

Inicialmente, devido a origem dos dados, todo o conteúdo HTML foi convertido para texto puro. Para essa tarefa, foi utilizada a biblioteca BeautifulSoup do Python[2]. Essa ferramenta analisa a estrutura do documento HTML e extrai apenas o conteúdo textual visível, descartando toda a marcação de formatação (tags). O resultado desta etapa é um texto sem estruturas HTML.

[2] <https://www.crummy.com/software/BeautifulSoup/>

4.2.2 Filtragem de Conteúdo e Redução de Ruído Semântico

Após a conversão para texto puro, foi realizada uma filtragem de conteúdo com dois objetivos principais: i) reduzir o volume de dados irrelevantes e ii) garantir a integridade da avaliação experimental. Em nosso contexto, é crucial evitar que o modelo tenha acesso a documentos gerados posteriormente pelos **analistas (servidores da PRODIRH responsáveis pela análise do processo)**. Documentos como pareceres e despachos frequentemente já contêm os dados consolidados que desejamos extrair. A presença desses documentos no contexto de entrada geraria um cenário não realístico e uma avaliação artificialmente otimista, onde o modelo apenas recuperaria a informação já processada pelo humano ao invés de extraí-la das fontes primárias.

Para reduzir o texto de entrada e permitir que o modelo foque no conteúdo essencial, foram eliminados: i) longos trechos de fundamentação legal, ii) justificativas padronizadas, iii) cabeçalhos, rodapés, informações de contatos (endereço, telefone) e blocos de assinatura (nomes, cargos e matrículas). Esses blocos de texto, embora necessários para a validade do processo administrativo, não contêm as informações específicas relevantes e representam um ruído para o modelo. Para isso, foram identificados múltiplos padrões de texto recorrentes e uma expressão regular foi compilada para remover todos esses trechos. O padrão foi configurado para ser flexível, abrangendo múltiplas linhas e ignorando a diferença entre maiúsculas e minúsculas, garantindo uma limpeza eficaz e a otimização do texto que seria efetivamente analisado na etapa de mineração.

Para garantir a padronização e a eficiência no processamento, foi realizado ainda um processo de normalização sintática, que consistiu na substituição de quebras de linha múltiplas, tabulações e sequências de espaços por um único caractere de espaço. Ao final, obteve-se um texto puro, contínuo e otimizado para ser submetido às fases subsequentes de transformação e extração de informações.

Complementarmente, para assegurar a integridade da avaliação (evitando a contaminação descrita anteriormente), a primeira ação foi remover os pareceres e despachos finais emitidos pela PRODIRH. Como os processos analisados já estavam concluídos, esses trechos continham um resumo com os próprios dados-alvo da extração. Para automatizar essa exclusão, e a partir do feedback dos servidores da PRODIRH, realizou-se uma análise manual dos documentos que identificou um padrão consistente: os pareceres técnicos geralmente se iniciam com expressões como "No presente processo"ou "Neste processo"e finalizam com os nomes e cargos dos diretores. Com base nesse padrão, foi desenvolvida uma expressão regular para localizar e remover essas seções de todos os documentos.

4.3 Mineração

A etapa de Mineração de Dados é uma parte importante do processo KDD, na qual os padrões e informações úteis são efetivamente descobertos. No nosso contexto, usaremos modelos generativos já existentes. Contudo, avaliaremos as diferentes possibilidades de uso desses modelos, visando maximiziar a qualidade da fase corresponde à extração da informação estruturada a partir do texto não estruturado.

No intuito de encontrar a melhor forma de extração de dados, variamos as nossas análises sobre as senguites perspectivas: i) diferentes estratégias de seleçao dos texto, ii) varição entre LLMs, e iii) variação das estratégias *In-context* de construção dos prompts.

4.3.1 Estratégias de Seleção de Texto

Na prática, o corpo principal de cada processo é bastante longo. Dessa forma, entregar todo o texto do processo para os LLMs não é garantia de efetividade, além de aumentar o custo de análise devido ao volume do texto. Logo, a decisão de qual porção de um processo submeter ao LLM tem impacto direto na qualidade e no custo da extração. Para investigar esses efeitos, quatro estratégias de seleção e fragmentação de texto — processo conhecido na literatura como ***chunking*** — foram implementadas e avaliadas:

- **Corpo Completo do Processo:** Nesta estratégia, o conteúdo textual de todos os documentos de um processo é concatenado para formar um único e longo texto de entrada. Essa estratégia servirá de comparação com as demais, pois mesmo o texto tendo passado por uma limpeza do seu conteúdo, é possível que ela favoreça a ocorrência do problema denominado "perdido no meio"(ou *lost in the middle*), em que a performance do LLM degrada ao lidar com contextos muito extensos (LIU et al., 2023).

- **Extração Direcionada ao Requerimento:** O método consiste em buscar e concatenar exclusivamente o conteúdo de documentos do tipo `REQUERIMENTO` ou `JUSTIFICATIVA`. Esta estratégia se baseia na heurística de que as informações mais cruciais e diretas sobre a solicitação estão contidas nos documentos iniciais redigidos pelo próprio servidor. A hipótese testada aqui é se um texto mais curto e focado, com alta densidade informacional, pode superar o texto completo ao reduzir o ruído de despachos, pareceres e outros trâmites. Esta abordagem avalia o balanço entre a completude da informação e a eficiência do processamento, testando se a fonte primária de dados é suficiente para a extração completa. A implementação da função que realiza essa busca direcionada é detalhada no Apêndice 8.6.

- **Segmentação por Sentenças (*Semantic Chunking*):** Esta estratégia executa uma análise sequencial do documento. O texto completo é primeiro segmentado

em blocos semânticos (*chunks*), utilizando o ponto final como delimitador, com o objetivo de isolar despachos ou pareceres individuais (a implementação detalhada desta função de segmentação se encontra no Apêndice 8.7). O sistema então itera sobre esses blocos, submetendo cada um a uma tentativa de extração. A cada iteração, o resultado é verificado e, se todos os campos-alvo forem preenchidos, o processo é interrompido por um mecanismo de parada antecipada (*early stopping*). Esta abordagem testa a hipótese de que a informação completa para a extração está frequentemente contida em um único bloco de intervenção processual.

- **Segmentação por Tokens de Tamanho Fixo (*Fixed-Size Chunking*):** A quarta estratégia aplica a técnica clássica de *chunking* por tamanho fixo, segmentando o texto em blocos com um máximo de 400 tokens. Para garantir que a divisão seja precisa em relação ao custo e à janela de contexto de cada família de modelos, foram utilizados os tokenizadores específicos de cada modelo.[3] Esta abordagem é um padrão da indústria para lidar com limites de contexto, buscando um meio-termo entre fornecer informação suficiente e manter um tamanho de entrada gerenciável. Assim como na estratégia por sentenças, a extração é feita de forma iterativa, pedaço por pedaço (*chunk by chunk*), utilizando também o mecanismo de *early stopping*. A função que implementa esta segmentação é apresentada no Apêndice 8.8.

4.3.2 Seleção dos Modelos Generativos de Extração

Um objetivo importante deste trabalho é avaliar como diferentes modelos generativos, com suas distintas arquiteturas e custos, executam a tarefa de extração de dados. Para facilitar um comparativo direto e eficiente, foi utilizado o OpenRouter[6], uma interface unificada que permite interagir com múltiplos modelos de diferentes provedores através de uma única API.

A seleção de modelos buscou incluir tanto opções comerciais de ponta quanto alternativas de acesso gratuito, permitindo uma análise de custo-benefício. A Tabela 1 detalha os modelos escolhidos, suas categorias e datas de referência.

A seleção inclui modelos como o `openai/gpt-4o-mini`, conhecido por seu balanço entre performance e custo, e o `deepseek-v3`, um forte concorrente na categoria comercial, com treinamento específico para tarefas de raciocínio ou *reasoning*. Para a análise de custo-benefício, foram escolhidos o `deepseek-r1-0528:free`, que permite uma comparação direta com sua contraparte paga, e o `gpt-oss-20b:free` como representante de um modelo de acesso gratuito.

[3] Para os modelos da OpenAI (como o GPT-4o-mini), foi utilizada a biblioteca tiktoken[4], que replica seu método de tokenização oficial. Para os modelos da DeepSeek, foi empregado o deepseek-tokenizer[5].

[6] <https://openrouter.ai/>

Tabela 1 – Modelos selecionados para o estudo, com suas categorias, datas de treinamento e lançamento.

Categoria	**Modelo**	**Treinamento até**	**Lançamento**
Modelos Comerciais	`gpt-4o-mini`	Outubro de 2023	Julho de 2024
	`gpt-4.1-mini`	Junho de 2024	Abril de 2025
	`gpt-5-mini`	Não disponível	Agosto de 2025
	`deepseek-v3`	Julho de 2024	Abril de 2024
Modelos Gratuitos	`gpt-oss-20b:free`	Junho de 2024	Agosto de 2025
	`deepseek-r1-0528:free`	Abril de 2024	Julho de 2024

A implementação da comunicação com os modelos via OpenRouter foi encapsulada em uma função em Python. Essa abordagem permitiu que a troca do modelo de extração para cada experimento fosse realizada de forma eficiente, alterando-se apenas um parâmetro na chamada da função. A implementação detalhada desta função de acesso à API está disponível para consulta no Apêndice 8.9.

4.3.3 Arquitetura de Extração em Duas Etapas: Geração e Validação

A metodologia de extração de dados não se limitou a uma única chamada ao modelo. Baseando-se na abordagem de proposta por (POLAK; MORGAN, 2024), foi implementado um pipeline sequencial projetado para assegurar a consistência dos dados através de um mecanismo de redundância. O funcionamento desta arquitetura divide-se em duas fases distintas de atuação do LLM:

1. **Fase de Geração (O Extrator):** Na primeira etapa, o LLM recebe o texto do processo (conforme a estratégia de seleção adotada) e um prompt *few-shot* contendo exemplos de pares entrada/saída. O objetivo aqui é puramente gerativo: o modelo deve identificar as entidades no texto não estruturado e mapeá-las para o JSON alvo.

2. **Fase de Validação (O Auditor):** Na segunda etapa, ocorre o processo de *Auto-questionamento*. O sistema utiliza uma nova instância do LLM, mas agora alimentada com um prompt diferente. Em vez de pedir para extrair dados, o sistema fornece o JSON gerado na etapa anterior e instrui o modelo a verificar se aqueles dados realmente existem no texto original. Para isso, o modelo deve responder a perguntas binárias de verificação (ex: "A data de fim '2024-12-01' consta no texto?"). Se o modelo responder "Não", o campo é descartado ou sinalizado para revisão.

Essa abordagem emula o processo humano de "revisão por pares", mas realizada pelo próprio modelo em momentos distintos, aumentando significativamente a confiabilidade

da extração final. Para facilitar a compreensão global do processo, o fluxo completo desta arquitetura de extração e validação está ilustrado na Figura 3.

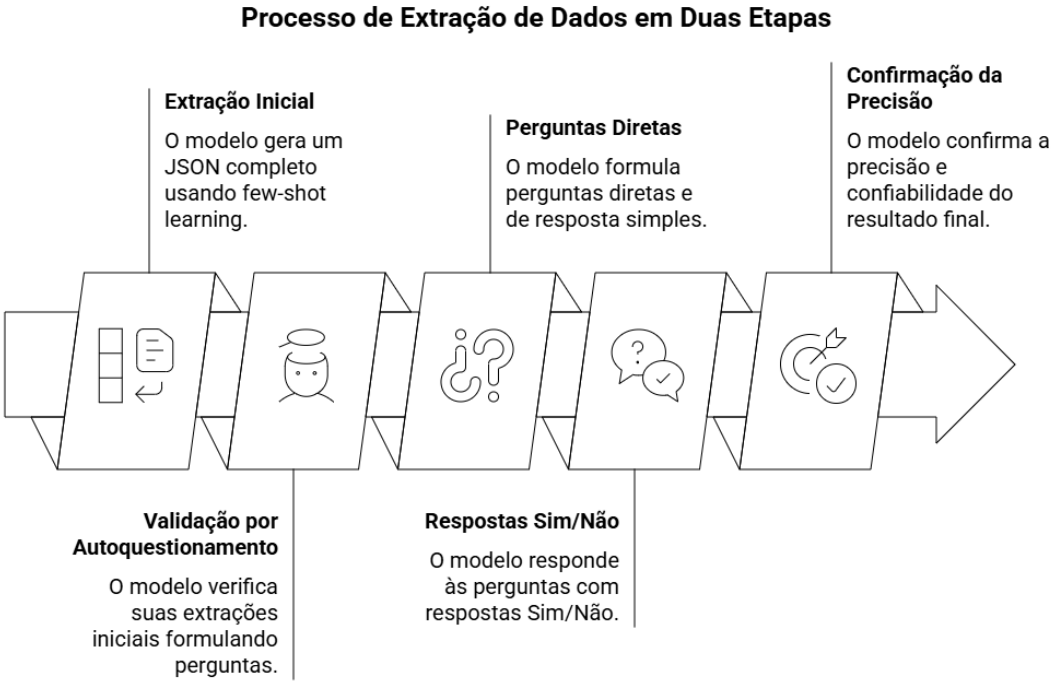

Figura 3 – Fluxograma da arquitetura de extração em duas etapas: Geração e Validação - Fonte: Autor

4.3.3.1 Etapa 1: Extração Inicial via *Few-Shot Learning*

Nesta etapa, o texto selecionado é inserido em um prompt. A arquitetura do prompt se baseia em *few-shot learning*. Foram desenvolvidos três prompts especializados: um para o processo de "Afastamento para Pós-Graduação *Stricto Sensu* (ASS)", e dois para o processo de "Licença para Capacitação", sendo um para o subtipo "Licença para Cursos de Aprimoramento (LCA)"e outro para "Licença para Cursos de Pós-Graduação (LCPG)".

Cada prompt foi estruturado com três componentes principais:

- Uma instrução clara da tarefa e do formato de saída JSON esperado.
- Exemplos de pares "texto de entrada"/ "saída JSON correta"para guiar o modelo.
- Um conjunto de instruções detalhadas sobre como tratar casos específicos (ex: dados ausentes, como no caso quando há a ausência da listagem da instituição onde será realizado os cursos).

Devido à sua extensão e para fins de reprodutibilidade, os templates completos e detalhados de cada um desses três prompts são apresentados, respectivamente, nos Apêndices 8.1, 8.2 e 8.3.

4.3.3.2 Etapa 2: Validação Iterativa dos Dados Extraídos

Para aumentar a confiabilidade dos dados obtidos na Etapa 1, foi implementado um processo de verificação automática da fidelidade de cada campo extraído. Essa abordagem de duas etapas força o modelo a engajar em um processo de raciocínio e autoavaliação. Em vez de confiar cegamente em sua primeira resposta, o modelo é compelido a reexaminar a fonte de dados sob a ótica de uma pergunta restrita e focada. Esse ciclo de "gerar e depois verificar"aumenta a robustez da extração final, simulando um processo de raciocínio mais próximo ao humano e mitigando o risco de alucinações.

A mecânica consiste em gerar um prompt de verificação para cada campo-chave do JSON extraído, cuja resposta esperada é apenas "Sim"ou "Não". Por exemplo, para validar o campo modalidade_cursos, o sistema pergunta: "O texto abaixo menciona que a modalidade do curso é 'EAD'?". A implementação completa desta lógica de verificação está detalhada nos Apêndices 8.4 e 8.5. A função `verify_extraction` é aplicada aos processos de Licença para Cursos de Aprimoramento (LCA), enquanto a função `verify_pos_extraction` é utilizada para os processos de Licença para Cursos de Pós-Graduação (LCPG) e Afastamento para Pós-Graduação Stricto Sensu (ASS).

Ao final deste ciclo de verificação, o objeto JSON é consolidado, contendo apenas os dados que foram positivamente validados a partir do texto original. Este objeto final é usado para verificar o grau de acertividade do modelo, comparado ao nosso gabarito. A Listagem 9 apresenta um exemplo desta estrutura de dados final para um processo de LCA.

```json
{
  "locais": [
    {
      "cursos": [
        {
          "nome": "Especialização em Economia Solidária, Inovação e Gestão Social",
          "carga_horaria": 368
        }
      ],
      "instituicao": "Educamundo"
    }
  ],
  "modalidade_cursos": "EAD",
  "carga_horaria_total": 368,
  "data_para_realizacao": {
    "fim": "2024-11-30",
    "inicio": "2024-08-01"
  }
}
```

Listing 9: Exemplo de objeto JSON final após as etapas de extração e verificação.

4.4 Planejamento da Avaliação

A etapa final do processo KDD, a Avaliação, é fundamental para validar a eficácia da solução proposta. Neste trabalho, optou-se por apresentar o desenho experimental, a definição detalhada das métricas adaptadas (Similaridade e Acurácia) e a análise dos dados integralmente no Capítulo 5 (Resultados Experimentais).

Essa organização tem como objetivo centralizar a discussão quantitativa, permitindo que o leitor tenha acesso às definições técnicas das métricas e aos critérios de avaliação imediatamente antes de observar sua aplicação prática na análise de desempenho das estratégias e modelos.

5 Resultados Experimentais

Este capítulo apresenta e analisa os resultados quantitativos obtidos a partir da aplicação da solução para extração de dados descrita no Capítulo 4. A seguir, definimos as métricas utilizadas para aferição de qualidade, detalhamos os eixos de avaliação e apresentamos a discussão comparativa dos dados.

5.1 Definição das Métricas Utilizadas

Visto que a tarefa consiste na extração de texto não estruturado, métricas de classificação binária(certo e errado) não seriam suficientes para capturar a qualidade parcial das respostas. Por recomendação da análise metodológica, as métricas foram adaptadas para o contexto de comparação textual entre o JSON extraído pelo modelo e o JSON gabarito. Abaixo, detalhamos as métricas de Similaridade e Acurácia utilizadas para compor os resultados apresentados neste capítulo.

5.1.1 Índice de Similaridade (Precisão Textual)

Para avaliar a qualidade da extração de cada campo individualmente, adotou-se um **Índice de Similaridade**. Esta métrica substitui a verificação binária (certo/errado) por uma escala contínua de 0 a 1, permitindo quantificar o quão próximo o texto extraído está do esperado. O cálculo varia conforme o tipo de dado:

- **Para valores numéricos ou booleanos:** A comparação é estrita. Se o valor é idêntico, o índice é 1.0; caso contrário, é 0.0.
- **Para valores textuais (strings):** Utilizou-se o algoritmo ***SequenceMatcher***[1] da biblioteca padrão do Python. Este algoritmo calcula a semelhança entre as strings baseando-se na maior subsequência comum contínua. O resultado é um valor flutuante entre 0.0 e 1.0 que atua como um proxy para a **precisão** da extração, penalizando erros de digitação ou formatação sem descartar totalmente respostas parcialmente corretas.
- **Para valores em lista:** Em campos com múltiplos valores (ex: lista de cursos), o algoritmo primeiro alinha cada item do gabarito com seu correspondente mais similar na lista extraída. O índice final da propriedade é a média das similaridades desses pares.

[1] <https://docs.python.org/3/library/difflib.html>

5.1.2 Similaridade Média por Processo

Para obter uma visão consolidada do desempenho do modelo em um processo inteiro (um objeto JSON completo), calculamos a média aritmética dos Índices de Similaridade de todas as N propriedades avaliadas naquele documento:

$$\text{Similaridade Média} = \frac{1}{N} \sum_{i=1}^{N} \text{Sim}_i$$

Onde Sim_i é o índice de similaridade da i-ésima propriedade.

5.1.3 Acurácia (Limiar de Aceitação)

Para complementar a análise com uma métrica mais rigorosa, calculou-se a **Acurácia**. Diferente da similaridade, a acurácia é binária: considera-se que o modelo "acertou"o campo apenas se a extração for virtualmente perfeita. A regra de decisão aplicada foi:

$$\text{Acurácia}_i = \begin{cases} 1.0 & \text{se } \text{Sim}_i \geq 0.95 \\ 0.0 & \text{caso contrário} \end{cases}$$

O limiar de 0.95 (95% de similaridade) foi estabelecido para tolerar diferenças mínimas de formatação (como espaços extras ou pontuação), mas rejeitar qualquer divergência de conteúdo semântico. A Acurácia Geral do processo é a média das acurácias individuais de seus campos.

5.2 Eixos de Avaliação Experimental

A avaliação experimental é estruturada em torno da análise de duas variáveis principais: a estratégia de seleção de texto e o modelo de linguagem utilizado. A Tabela 2 resume os eixos de avaliação e as respectivas abordagens testadas neste trabalho.

Tabela 2 – Resumo dos eixos, modelos e critérios avaliados no estudo experimental.

Eixo de Análise	**Abordagens / Critérios**
Estratégia de Seleção de Texto	• Corpo Completo do Processo • Extração Direcionada ao Requerimento • Segmentação por Sentenças • Segmentação por Tokens de Tamanho Fixo
Modelo de Linguagem (LLM)	• **Comerciais:** `gpt-4o-mini`, `gpt-4.1-mini`, `gpt-5-mini`, `deepseek-v3`. • **Gratuitos:** `gpt-oss-20b:free`, `deepseek-r1-0528:free`.
Critérios avaliados	• **Métricas Gerais:** Similaridade Média e Acurácia. • **Propriedades avaliadas:** – Instituição – Data de Fim – Nome do curso

5.3 Organização da Análise

Para organizar a discussão e facilitar a compreensão dos múltiplos experimentos realizados, a análise subsequente foi estruturada da seguinte forma:

- A **Seção 5.4** estabelece as bases para a análise de desempenho, apresentando dois modelos representativos (um comercial e um gratuito) e as propriedades-chave que serão o foco da comparação nas seções seguintes.
- A **Seção 5.5** apresenta uma análise aprofundada, avaliando o desempenho dos modelos sob a estratégia de Corpo Completo.
- A **Seção 5.6** investiga a performance com a estratégia de Extração Direcionada ao Requerimento, testando a robustez de cada modelo em um cenário de contexto reduzido.
- As **Seções 5.7** e **5.8** concluem as análises individuais, avaliando o impacto das duas estratégias baseadas em fragmentação de texto.
- Por fim, a **Seção 5.9** consolida os resultados e apresenta as conclusões finais.

5.4 Análise Comparativa de Estratégias e Modelos

Esta seção apresenta a análise comparativa de desempenho entre as quatro estratégias de seleção de texto, avaliadas em dois modelos representativos: o comercial `openai/gpt-4o-mini` e o gratuito `openai/gpt-oss-20b:free`. O objetivo é realizar uma análise que permita não apenas identificar a estratégia mais eficaz, mas também compreender de que forma a escolha do modelo interage com a qualidade do texto de entrada.

Para todas as execuções, a temperatura dos modelos foi configurada como 0.0 para garantir a máxima consistência e determinismo nos resultados.

Para a análise, consideraremos a extração dos seguintes dados:

- **Instituição:** Representa o nome da organização (universidade, centro de treinamento, etc.) onde o curso ou a pós-graduação será realizado.
- **Data de Fim:** Refere-se à data de término do período de licença ou afastamento solicitado pelo servidor.
- **Nome do Curso:** O nome oficial do curso de aprimoramento, mestrado ou doutorado que motiva a solicitação.

Estes três campos foram escolhidos como foco da análise por representarem informações essenciais e universais a todos os tipos de processo avaliados (LCA, LCPG e ASS). Toda solicitação de licença ou afastamento para fins de capacitação possui, necessariamente, uma instituição de destino, um período de realização com data de término, e um ou mais cursos que justificam o pedido.

Para facilitar a descrição, utilizamos as seguintes siglas para os processos:

- **LCA:** Licença para Cursos de Aprimoramento.
- **LCPG:** Licença para Cursos de Pós-Graduação.
- **ASS:** Afastamento para Pós-Graduação Stricto Sensu.

A análise de performance dos modelos é apresentada em duas tabelas distintas. A Tabela 3 detalha os índices de Acurácia, métrica na qual uma extração só é classificada como um "acerto"se seu Índice de Similaridade em relação ao gabarito for igual ou superior ao limiar de 0.95. Em complemento, a Tabela 4 apresenta os valores de Similaridade Média, que mensuram a fidelidade textual da extração. Em ambas, os resultados são agrupados pelos tipos de processo (LCA, LCPG, ASS) e a métrica "Geral"é calculada pela média aritmética das três propriedades avaliadas.

Tabela 3 – Resultados comparativos de **Acurácia** entre os modelos `GPT-4o-mini` e `GPT-OSS 20b`.

	Corpo Completo		**Requerimento**		**Sentenças**		**Tokens**	
Processo - Propriedade	**4o-mini**	**Oss**	**4o-mini**	**Oss**	**4o-mini**	**Oss**	**4o-mini**	**Oss**
LCA - Instituição	**0.745**	0.697	0.590	0.660	0.400	0.316	0.316	0.318
LCA - Data de Fim	**0.839**	0.782	0.798	0.692	0.609	0.417	0.529	0.333
LCA - Nome do Curso	0.865	**0.916**	0.837	0.884	0.762	0.531	0.664	0.613
LCA - Média	**0.816**	0.798	0.742	0.745	0.590	0.421	0.503	0.421
LCPG - Instituição	0.917	0.667	**1.000**	0.333	0.667	0.405	0.500	0.200
LCPG - Data de Fim	0.615	**0.692**	0.455	0.500	0.385	0.357	0.231	0.167
LCPG - Nome do Curso	**0.917**	0.417	0.900	0.667	0.500	0.400	0.417	0.400
LCPG - Média	**0.816**	0.592	0.785	0.500	0.517	0.387	0.383	0.256
ASS - Instituição	0.632	0.658	0.500	0.447	0.421	0.421	0.368	0.368
ASS - Data de Fim	0.816	**0.842**	0.605	0.579	0.368	0.342	0.338	0.132
ASS - Nome do Curso	0.684	0.711	**0.737**	0.579	0.500	0.500	0.421	**0.447**
ASS - Média	0.711	**0.737**	0.614	0.535	0.430	0.421	0.376	0.316

Tabela 4 – Resultados comparativos de **Similaridade** entre os modelos `GPT-4o-mini` e `GPT-OSS 20b`.

	Corpo Completo		**Requerimento**		**Sentenças**		**Tokens**	
Processo - Propriedade	**4o-mini**	**Oss**	**4o-mini**	**Oss**	**4o-mini**	**Oss**	**4o-mini**	**Oss**
LCA - Instituição	**0.905**	0.877	0.750	0.852	0.591	0.522	0.507	0.538
LCA - Data de Fim	**0.975**	0.870	0.952	0.815	0.755	0.454	0.716	0.367
LCA - Nome do Curso	0.922	**0.957**	0.913	0.900	0.861	0.674	0.824	0.802
LCA - Média	**0.934**	0.901	0.872	0.856	0.736	0.550	0.682	0.569
LCPG - Instituição	0.975	0.830	**0.997**	0.678	0.730	0.610	0.593	0.482
LCPG - Data de Fim	**0.898**	0.844	0.852	0.825	0.700	0.617	0.354	0.300
LCPG - Nome do Curso	**0.992**	0.717	0.991	0.910	0.628	0.599	0.590	0.687
LCPG - Média	**0.955**	0.797	0.947	**0.804**	0.686	0.609	0.512	0.490
ASS - Instituição	**0.820**	0.809	0.756	0.723	0.660	0.637	0.576	0.573
ASS - Data de Fim	0.897	**0.900**	0.689	0.639	0.434	0.387	0.371	0.145
ASS - Nome do Curso	0.853	**0.863**	0.862	0.756	0.663	0.638	0.538	0.604
ASS - Média	0.857	**0.857**	0.769	0.706	0.586	0.554	0.495	0.441

5.4.1 Interpretação dos Resultados

A primeira análise descreve a comparação geral entre as estratégias que preservam um contexto amplo (Corpo Completo e Requerimento) versus aquelas que o fragmentam (Sentenças e Tokens) o texto original. Para ambos os modelos avaliados, o GPT-4o-mini e o GPT-OSS 20B, observa-se uma superioridade das abordagens de contexto amplo em todos os tipos de processo. Por exemplo, na Tabela 4, o processo de Afastamento (ASS) mostra uma Similaridade Média do GPT-4o-mini na estratégia Corpo Completo de **0.857**, superior ao da estratégia por Tokens (**0.495**). Essa mesma tendência se repete com o GPT-OSS 20b, cujos scores caem de **0.857** para **0.441** nos mesmos cenários. A hipótese

central para essa disparidade, reforçada pelo fato de ocorrer em modelos de diferentes capacidades, é que a tarefa de extração neste domínio é fundamentalmente relacional. Ou seja, o modelo não precisa apenas encontrar entidades, mas compreender a relação entre elas (um curso e sua instituição, uma data de início e sua duração). A fragmentação do texto quebra essas ligações semânticas, tornando a tarefa mais difícil e explicando a queda de performance das estratégias de segmentação.

Aprofundando a análise, comparamos as duas melhores estratégias, Corpo Completo e Requerimento, para entender o balanço entre fornecer o máximo de informação e a informação mais relevante. Para os processos de LCA e ASS, a estratégia Corpo Completo se mostrou a mais robusta. Por exemplo, na Tabela 4, com o GPT-4o-mini, a Similaridade Média foi de **0.934** (LCA) e **0.857** (ASS), valores superiores aos obtidos apenas com o Requerimento. A hipótese para este resultado, validada pela análise qualitativa dos dados, é que a estratégia Requerimento pode sofrer de um problema de dados faltantes textuais. Em muitos processos de LCA, por exemplo, o servidor anexa a lista detalhada de cursos em um documento PDF, e o texto do requerimento inicial contém apenas uma referência a esse anexo (ex: "conforme lista de cursos anexa"). Como este estudo analisa apenas o conteúdo textual extraído do SUAP, a informação contida no PDF não está presente no documento de requerimento. No entanto, em despachos ou pareceres de setores intermediários, os analistas frequentemente transcrevem essa lista para o corpo do texto a fim de realizar sua análise. Dessa forma, a estratégia Corpo Completo é a única que captura essa informação em etapas posteriores, explicando sua performance superior.

Em contraste, para o LCPG, a estratégia Requerimento demonstrou competitividade e eficiência. Na Tabela 4, os resultados entre Corpo Completo e Requerimento foram virtualmente equivalentes: o GPT-4o-mini alcançou **0.955** com texto completo e **0.947** com requerimento, enquanto o GPT-OSS 20b apresentou uma ligeira preferência pelo requerimento (**0.804** vs **0.797**). Isso consolida a hipótese de que a natureza deste tipo de processo torna o documento de requerimento informacionalmente autossuficiente. Por se tratar de uma solicitação para um único e bem definido objetivo, todas as informações essenciais são consolidadas de forma clara no pedido inicial. Nesse cenário, o texto do Requerimento provou-se suficiente, evitando o processamento de tokens administrativos adicionais sem perda de qualidade na extração.

Para entender a razão da queda de performance nas estratégias de segmentação, a análise aprofunda-se em propriedades vulneráveis à fragmentação do texto, comparando o impacto em ambos os modelos. O campo Data de Fim no processo de Afastamento (ASS) ilustra essa falha. Seguindo a Tabela 4, com a estratégia Corpo Completo, tanto o GPT-4o-mini quanto o GPT-OSS 20b alcançaram excelentes Índices de Similaridade (0.897 e 0.900, respectivamente). Contudo, ao usar a estratégia por Tokens, a performance do GPT-4o-mini despencou para 0.371, e a do GPT-OSS 20b sofreu uma queda ainda maior,

caindo para um índice de apenas 0.145. A hipótese é que a Data de Fim frequentemente requer inferência (cálculo a partir de dados que podem agora estar em chunks diferentes), uma capacidade que depende de um contexto amplo. A queda mais acentuada do GPT-OSS 20b sugere que sua capacidade de realizar inferências com informação fragmentada é mais limitada.

Nota-se que os resultados da Tabela 3 e da Tabela 4 são amplamente consistentes entre as duas métricas. As estratégias e modelos que se destacam na Similaridade, que mede a qualidade geral da extração, são, na grande maioria dos casos, os mesmos que lideram em Acurácia, que mede a proporção de extrações quase perfeitas. Essa consistência é significativa, pois valida a robustez das conclusões apresentadas. Ela indica que as estratégias com maior similaridade não estão apenas "chegando perto"da resposta correta com mais frequência, mas também estão atingindo o alto grau de precisão (Similaridade ≥ 0.95) necessário para serem consideradas um acerto.

Em suma, os resultados demonstram que a performance de ambos os modelos está ligada à capacidade da estratégia de fornecer um contexto coeso e amplo. As altas performances alcançadas com as estratégias Corpo Completo e Requerimento atestam a robustez do GPT-4o-mini para tarefas de raciocínio, mesmo com informação parcial. O GPT-OSS 20b, por sua vez, mostrou-se competente, mas mais dependente do contexto completo para atingir seu pico de performance. A principal conclusão desta análise é dupla: primeiro, que abordagens de contexto amplo são um pré-requisito para esta tarefa; e segundo, que embora o modelo comercial ofereça maior robustez, a alternativa gratuita representa uma opção viável e de alto desempenho quando combinada com a estratégia de entrada de dados mais adequada.

5.5 Análise Comparativa dos Modelos na Estratégia 'Corpo Completo'

Após determinar o impacto das diferentes estratégias de seleção de texto, a segunda fase da avaliação experimental foca em comparar o desempenho dos diferentes LLMs. Para simplificar a análise, todos os testes desta seção foram conduzidos utilizando uma única estratégia de entrada: o Corpo Completo do Processo. Esta estratégia foi escolhida por ser uma abordagem robusta que fornece o máximo de contexto disponível para a análise.

As Tabelas 5 e 6 apresentam os resultados comparativos de Acurácia e Similaridade, respectivamente, para os seis modelos avaliados.

Tabela 5 – Resultados de **Acurácia** por modelo, utilizando a estratégia "Corpo Completo".

		Modelos Comerciais				Modelos Gratuitos	
	Propriedade	**GPT-5 Mini**	**GPT-4.1 Mini**	**GPT-4o-mini**	**DeepSeek-v3**	**r1-0528-free**	**GPT-OSS 20b**
LCA	Acc Instituição	0.694	0.732	**0.745**	0.722	0.716	0.697
	Acc Data de Fim	**0.908**	0.851	0.839	0.885	0.780	0.782
	Acc Nome do Curso	0.916	**0.920**	0.865	0.912	0.910	0.916
	Acc (Média)	0.839	0.834	0.816	**0.840**	0.802	0.798
LCPG	Acc Instituição	0.750	0.833	**0.917**	0.833	0.875	0.667
	Acc Data de Fim	**0.846**	0.692	0.615	0.692	0.667	0.692
	Acc Nome do Curso	0.417	0.750	**0.917**	0.667	0.750	0.417
	Acc (Média)	0.671	0.758	**0.816**	0.731	0.764	0.592
ASS	Acc Instituição	0.658	0.658	0.632	**0.737**	0.711	0.658
	Acc Data de Fim	0.842	0.816	0.816	**0.868**	0.842	0.842
	Acc Nome do Curso	0.711	0.711	0.684	**0.711**	0.684	0.711
	Acc (Média)	0.737	0.728	0.711	**0.772**	0.746	0.737

Tabela 6 – Resultados de **Similaridade** por modelo, utilizando a estratégia "Corpo Completo".

		Modelos Comerciais				Modelos Gratuitos	
	Propriedade	**GPT-5 Mini**	**GPT-4.1 Mini**	**GPT-4o-mini**	**DeepSeek-v3**	**r1-0528-free**	**GPT-OSS 20b**
LCA	Sim Instituição	0.862	0.888	**0.905**	0.874	0.890	0.877
	Sim Data de Fim	0.983	0.926	**0.975**	0.954	0.839	0.870
	Sim Nome do Curso	0.960	0.960	0.922	0.958	0.950	**0.957**
	Similaridade Média	**0.935**	0.925	0.934	0.929	0.893	0.901
LCPG	Sim Instituição	**0.949**	0.967	0.975	0.967	0.965	0.830
	Sim Data de Fim	0.885	0.890	**0.898**	0.913	0.822	0.844
	Sim Nome do Curso	0.731	0.921	**0.992**	0.844	0.921	0.717
	Similaridade Média	0.855	0.926	**0.955**	0.908	0.903	0.797
ASS	Sim Instituição	0.879	0.865	0.820	0.863	**0.905**	0.809
	Sim Data de Fim	0.924	0.916	0.897	**0.926**	0.902	0.900
	Sim Nome do Curso	**0.891**	0.876	0.853	0.883	0.863	0.863
	Similaridade Média	**0.898**	0.886	0.857	0.891	0.890	0.857

5.5.1 Interpretação dos Resultados

A primeira análise foca na performance geral dos modelos, avaliada pela Similaridade Média consolidada para cada tipo de processo. Os modelos comerciais (GPT-5 Mini, GPT-4.1 Mini, GPT-4o-mini e DeepSeek-v3) consistentemente ocupam o topo da performance, com médias frequentemente acima de 0.890. O GPT-5 Mini desponta como o modelo de melhor desempenho agregado, especialmente nos processos de LCA (0.935) e ASS (0.898). A hipótese para este resultado é que a performance geral está diretamente correlacionada com o volume de dados no conjunto pré-treinado e nas estratégias de *reasoning* usadas no treinamento desses modelos. Modelos mais avançados e recentes possuem uma capacidade superior de seguir instruções complexas e de raciocinar sobre contextos longos e ruidosos, como os da estratégia Corpo Completo.

5.5.1.1 Custo-Benefício: Modelos Pagos vs. Gratuitos

As métricas revelam que, embora os modelos gratuitos sejam inferiores no geral, o DeepSeek r1-0528-free demonstra uma competitividade notável. No processo de ASS, por exemplo, ele se destaca ao obter o maior Índice de Similaridade na extração de Instituição (0.905), superando todos os modelos comerciais nesse cenário específico. Embora a análise detalhada mostre que o DeepSeek r1-0528-free não obteve o maior índice para a extração de Instituição no processo LCA, seu resultado (0.890) demonstra uma alta competitividade, permanecendo muito próximo ao do líder da métrica, o GPT-4o-mini (0.905). A hipótese é que a arquitetura do DeepSeek pode ser particularmente otimizada para certos tipos de reconhecimento de entidades. Em contraste, o GPT-OSS 20b se posiciona como uma opção mais básica, com um desempenho geral ligeiramente inferior em cenários complexos.

No entanto, é na análise de propriedades que exigem raciocínio mais complexo que os modelos comerciais justificam seu custo e demonstram suas especializações. A extração da Data de Fim, uma tarefa que frequentemente exige inferência, é um ponto forte claro da família DeepSeek, com a versão paga DeepSeek-v3 atingindo a Similaridade mais alta de 0.926 no processo ASS. Por outro lado, a extração de "Nome do Curso"no processo LCPG, uma tarefa que demanda precisão no reconhecimento de uma entidade específica em meio a um texto denso, teve como campeão o GPT-4o-mini, com um índice quase perfeito de 0.992. Essa especialização sugere que, embora os modelos gratuitos sejam extremamente competentes, os modelos comerciais de ponta mantêm uma vantagem em tarefas de raciocínio com maior demanda.

5.6 Análise Comparativa dos Modelos na Estratégia 'Extração Direcionada ao Requerimento'

O propósito desta análise é avaliar a robustez e a capacidade de raciocínio de cada um dos seis LLMs quando confrontados com um texto de entrada mais curto e potencialmente incompleto. Ao focar exclusivamente nos documentos de solicitação inicial, esta estratégia testa a habilidade dos modelos de extrair dados de forma precisa sem o auxílio de informações complementares ou esclarecimentos encontrados em outras partes do processo (despachos e pareceres posteriores).

As Tabelas 7 e 8 apresentam os resultados comparativos de Acurácia e Similaridade para esta estratégia.

Tabela 7 – Resultados de **Acurácia** por modelo, utilizando a estratégia "Extração Direcionada".

		Modelos Comerciais				Modelos Gratuitos	
	Propriedade	**GPT-5 Mini**	**GPT-4.1 Mini**	**GPT-4o-mini**	**DeepSeek-v3**	**r1-0528-free**	**GPT-OSS 20b**
LCA	Acc Instituição	0.518	0.571	0.590	0.610	0.667	**0.760**
	Acc Data de Fim	**0.848**	0.787	0.798	0.809	0.673	0.692
	Acc Nome do Curso	0.814	**0.864**	0.837	0.841	0.827	0.844
	Acc (Média)	0.727	0.741	0.742	0.753	0.722	**0.765**
LCPG	Acc Instituição	0.690	0.900	**1.000**	0.900	0.460	0.333
	Acc Data de Fim	0.500	0.545	0.455	**0.636**	0.400	0.500
	Acc Nome do Curso	0.800	**0.900**	0.900	0.800	0.550	0.667
	Acc (Média)	0.663	0.782	**0.785**	0.779	0.470	0.500
ASS	Acc Instituição	0.526	0.474	0.500	0.526	**0.579**	0.447
	Acc Data de Fim	**0.605**	0.579	0.495	0.605	0.605	0.579
	Acc Nome do Curso	0.500	**0.737**	0.737	0.684	0.713	0.579
	Acc (Média)	0.544	0.597	0.577	0.605	**0.632**	0.535

Tabela 8 – Resultados de **Similaridade** por modelo, utilizando a estratégia "Extração Direcionada ao Requerimento".

		Modelos Comerciais				Modelos Gratuitos	
	Propriedade	**GPT-5 Mini**	**GPT-4.1 Mini**	**GPT-4o-mini**	**DeepSeek-v3**	**r1-0528-free**	**GPT-OSS 20b**
LCA	Sim Instituição	0.679	0.743	0.750	0.752	0.762	**0.852**
	Sim Data de Fim	0.921	0.860	**0.952**	0.908	0.750	0.815
	Sim Nome do Curso	0.877	**0.928**	0.913	0.918	0.902	0.900
	Similaridade Média	0.826	0.844	**0.872**	0.859	0.805	0.856
LCPG	Sim Instituição	0.770	0.989	**0.997**	0.988	0.720	0.678
	Sim Data de Fim	0.850	0.779	0.852	**0.879**	0.630	0.825
	Sim Nome do Curso	0.920	0.951	**0.991**	0.874	0.760	0.910
	Similaridade Média	0.847	0.906	**0.947**	0.914	0.703	0.804
ASS	Sim Instituição	0.793	0.765	0.756	**0.784**	0.764	0.723
	Sim Data de Fim	0.689	**0.682**	0.689	0.689	0.666	0.639
	Sim Nome do Curso	0.781	**0.863**	0.862	0.845	0.842	0.756
	Similaridade Média	0.754	0.770	0.769	**0.773**	0.757	0.706

5.6.1 Interpretação dos Resultados

A primeira análise foca na performance geral, medida pela Similaridade Média. Os resultados da Tabela 8 mostram que, neste cenário de contexto restrito, os modelos da família OpenAI (GPT-5 Mini, GPT-4.1 Mini, GPT-4o-mini) se destacam consistentemente, ocupando as primeiras posições na maioria dos casos, especialmente nos processos de licença (LCA e LCPG). O GPT-4o-mini, em particular, demonstrou a maior robustez, alcançando a Similaridade Média mais alta para LCA (0.872) e LCPG (0.947). A hipótese para essa liderança é que esses modelos possuem uma capacidade de raciocínio mais apurada, conseguindo inferir e extrair informações corretamente mesmo quando o contexto é limitado.

5.6.1.1 Análise por Tipo de Processo

A transição da estratégia Corpo Completo para Requerimento resulta em uma queda de performance, particularmente no processo de Afastamento (ASS), onde a melhor Similaridade Média cai de 0.898 (com GPT-5 Mini na seção anterior) para 0.773 (com DeepSeek-v3 nesta estratégia). A hipótese central para essa queda vai além da simples perda de um contexto confirmatório; ela reside na própria natureza evolutiva do processo administrativo. O documento de requerimento representa a intenção inicial do servidor, mas as informações-chave, como as datas de afastamento, podem ser alteradas durante a tramitação. Um setor pode ajustar o período solicitado para melhor se adequar às necessidades do departamento ou para respeitar limites regulatórios, como o percentual máximo de servidores afastados simultaneamente. Consequentemente, o Corpo Completo é a única estratégia que garante o acesso à "versão final"do processo, contida nos despachos finais. Ao analisar apenas o requerimento, o modelo pode extrair perfeitamente a data solicitada, mas falhar na avaliação por esta não ser a data aprovada e registrada no gabarito, explicando a queda de performance.

5.6.1.2 Análise por Propriedade

O campo Data de Fim, por exemplo, se confirmou como um grande desafio. Mesmo para os melhores modelos, o Índice de Similaridade raramente superou 0.90 nos processos de Afastamento (ficando na casa de 0.68), reforçando a hipótese de que a ausência de despachos confirmatórios torna a extração desta informação particularmente ambígua para todos.

O resultado mais notável, no entanto, veio da performance dos modelos gratuitos. O GPT-OSS 20b apresentou um desempenho excepcional no processo LCA, obtendo o maior Índice de Similaridade entre todos os modelos para a extração de Instituição (0.852) e atingindo a maior Acurácia Média (0.765), superando até os modelos comerciais. A hipótese para este fenômeno contraintuitivo é que a simplicidade do texto na estratégia Requerimento favorece a sua arquitetura. Sem o "ruído"de múltiplos documentos e trâmites, o modelo consegue focar em padrões de extração mais diretos. Modelos mais complexos, por outro lado, podem ser mais suscetíveis a ambiguidades quando o contexto confirmatório que esperam encontrar está ausente.

Em suma, a estratégia de Extração Direcionada ao Requerimento atua como um teste de robustez que, ao limitar o contexto, penaliza a performance de todos os modelos, mas revela diferenças cruciais em suas arquiteturas e capacidades de raciocínio. Os modelos comerciais da OpenAI, em particular o GPT-4o-mini, demonstraram a maior resiliência ao contexto parcial, consolidando sua posição como as ferramentas de melhor desempenho geral, especialmente em cenários que exigem inferências mais complexas, como o LCPG. Contudo, a análise também expõe uma notável especialização entre os modelos gratuitos:

o GPT-OSS 20b se destacou em extrair entidades diretas de um texto limpo e focado (LCA), sugerindo que pode ser uma excelente opção para tarefas mais simples.

5.7 Análise Comparativa dos Modelos na Estratégia 'Segmentação por Tokens'

Esta seção analisa a performance dos modelos sob a estratégia de Segmentação por Tokens de Tamanho Fixo, na qual os documentos são divididos em chunks de 400 tokens. A análise busca determinar se o processamento do texto em partes gerenciáveis é eficaz para a extração de dados complexos e como a capacidade de cada modelo é impactada por essa fragmentação controlada do contexto.

As Tabelas 9 e 10 apresentam os resultados comparativos de Acurácia e Similaridade para esta estratégia.

Tabela 9 – Resultados de **Acurácia** por modelo, utilizando a estratégia "Segmentação por Tokens".

		Modelos Comerciais				Modelos Gratuitos	
	Propriedade	**GPT-5 Mini**	**GPT-4.1 Mini**	**GPT-4o-mini**	**DeepSeek-v3**	**r1-0528**	**GPT-OSS 20b**
LCA	Acc Instituição	0.260	0.280	0.316	0.256	0.280	**0.318**
	Acc Data de Fim	**0.594**	0.471	0.529	0.517	0.341	0.333
	Acc Nome do Curso	0.622	0.651	0.664	0.647	**0.667**	0.613
	Acc (Média)	0.492	0.467	**0.503**	0.473	0.429	0.421
LCPG	Acc Instituição	0.500	0.500	0.500	0.417	**0.500**	0.200
	Acc Data de Fim	0.273	0.231	0.231	**0.231**	0.000	0.167
	Acc Nome do Curso	0.200	0.333	**0.417**	0.333	0.375	0.400
	Acc (Média)	0.324	0.355	**0.383**	0.327	0.292	0.256
ASS	Acc Instituição	**0.395**	0.368	0.368	0.395	0.342	0.368
	Acc Data de Fim	0.132	0.132	**0.158**	0.132	0.105	0.132
	Acc Nome do Curso	**0.474**	0.447	0.421	0.447	0.447	0.447
	Acc (Média)	**0.334**	0.316	0.316	0.325	0.298	0.316

Tabela 10 – Resultados de **Similaridade** por modelo, utilizando a estratégia "Segmentação por Tokens".

		Modelos Comerciais				Modelos Gratuitos	
Processo	**Propriedade**	**GPT-5 Mini**	**GPT-4.1 Mini**	**GPT-4o-mini**	**DeepSeek-v3**	**r1-0528**	**GPT-OSS 20b**
LCA	Sim Instituição	0.462	0.495	0.507	0.468	0.514	**0.538**
	Sim Data de Fim	0.671	0.548	**0.716**	0.586	0.383	0.367
	Sim Nome do Curso	0.782	0.795	**0.824**	0.807	0.809	0.802
	Similaridade Média	0.638	0.613	**0.682**	0.620	0.569	0.569
LCPG	Sim Instituição	0.584	**0.593**	0.593	0.525	0.557	0.482
	Sim Data de Fim	0.345	0.354	**0.354**	0.292	0.089	0.300
	Sim Nome do Curso	0.513	0.581	0.590	0.415	0.454	**0.627**
	Similaridade Média	0.481	0.509	**0.512**	0.411	0.367	0.470
ASS	Sim Instituição	0.577	0.569	0.576	**0.582**	0.529	0.573
	Sim Data de Fim	0.145	0.145	**0.371**	0.145	0.105	0.145
	Sim Nome do Curso	**0.614**	0.587	0.538	0.564	0.565	0.604
	Similaridade Média	0.445	0.434	**0.495**	0.430	0.400	0.441

5.7.1 Interpretação dos Resultados

A primeira análise foca na performance geral dos modelos, avaliada pela Similaridade Média (Tabela 10). O resultado mais imediato é a queda drástica de performance para todos os modelos, sem exceção. O melhor desempenho geral foi alcançado pelo GPT-4o-mini no processo LCA, com uma Similaridade Média de apenas 0.682, um valor muito inferior aos scores acima de 0.90 obtidos pelo mesmo modelo nas estratégias de contexto amplo. Para os processos LCPG e ASS, os resultados são ainda piores, com scores gerais frequentemente abaixo de 0.50.

A hipótese central para essa falha generalizada é a fragmentação do contexto relacional. A divisão do texto em *chunks* arbitrários, mesmo com sobreposição, frequentemente separa informações que precisam ser analisadas em conjunto. Por exemplo, um chunk pode conter o nome de um curso, enquanto o chunk seguinte contém a instituição e a carga horária. Ao processar cada fragmento isoladamente, o LLM perde a capacidade de conectar essas entidades, resultando em extrações incompletas ou incorretas.

5.7.1.1 Análise por Propriedade

A extração de Nome do Curso, uma entidade textual geralmente bem definida, ainda apresenta uma performance razoável (Índice de Similaridade frequentemente acima de 0.80 para LCA). Isso ocorre porque a identificação de um nome próprio é uma tarefa de reconhecimento de entidade que depende menos de um contexto amplo.

Em contraste, a extração de Data de Fim e Instituição sofre uma queda acentuada. O Índice de Similaridade para Data de Fim no processo ASS, por exemplo, atinge valores extremamente baixos, como 0.371 para o GPT-4o-mini e até 0.105 para modelos gratuitos. A hipótese, já discutida anteriormente, é que essas propriedades exigem raciocínio inferencial e relacional. A Data de Fim precisa do contexto da data de início e da duração, enquanto a Instituição precisa do contexto do curso.

5.7.1.2 Comparativo entre Modelos

Sob a condição de alta degradação contextual imposta por esta estratégia, o GPT-4o-mini manteve uma resiliência superior, liderando nas métricas gerais de Similaridade. Contudo, a disparidade de performance entre os modelos foi visivelmente menor do que nos cenários de contexto amplo. A hipótese é que a fragmentação do texto atuou como o principal gargalo de desempenho, sobrepondo-se às capacidades individuais de cada modelo. Uma vez que as ligações semânticas essenciais são quebradas na entrada, a capacidade superior de raciocínio dos modelos mais avançados se torna ineficaz, pois lhes falta a informação necessária para operar.

Em suma, os resultados demonstram que, para a tarefa de extração de dados

relacionais e inferenciais de processos administrativos, a estratégia de Segmentação por Tokens de Tamanho Fixo é ineficaz. A perda de contexto de longo alcance que ela acarreta impede que os modelos realizem o raciocínio necessário, validando a conclusão de que estratégias que preservam a integridade de contextos amplos (Corpo Completo e Requerimento) são essenciais para o sucesso desta aplicação.

5.8 Análise Comparativa dos Modelos na Estratégia 'Segmentação por Sentenças'

A última análise experimental investiga o desempenho dos seis LLMs sob a estratégia de Segmentação por Sentenças. Esta abordagem busca um meio-termo entre as estratégias de contexto amplo e as de fragmentação por tokens, ao dividir o documento em blocos semanticamente coesos que visam corresponder a despachos ou pareceres por completo. O objetivo é avaliar se processar o documento em unidades de significado completas é uma abordagem viável.

As Tabelas 11 e 12 apresentam os resultados comparativos de Acurácia e Similaridade para esta estratégia.

Tabela 11 – Resultados de **Acurácia** por modelo, utilizando a estratégia "Segmentação por Sentenças".

		Modelos Comerciais				**Modelos Gratuitos**	
	Propriedade	**GPT-5 Mini**	**GPT-4.1 Mini**	**GPT-4o-mini**	**DeepSeek-v3**	**r1-0528**	**GPT-OSS 20b**
LCA	Acc Instituição	0.268	0.286	**0.400**	0.310	0.320	0.316
	Acc Data de Fim	**0.623**	0.575	0.609	0.586	0.341	0.417
	Acc Nome do Curso	0.761	**0.761**	0.762	0.671	0.655	0.531
	Acc (Média)	0.551	0.541	**0.590**	0.522	0.439	0.421
LCPG	Acc Instituição	0.700	0.667	0.667	0.667	**0.625**	0.405
	Acc Data de Fim	0.545	0.385	0.385	**0.538**	0.444	0.357
	Acc Nome do Curso	0.400	**0.500**	0.500	0.500	0.500	0.400
	Acc (Média)	0.548	0.517	0.517	**0.568**	0.523	0.387
ASS	Acc Instituição	**0.447**	0.421	0.421	0.447	0.474	0.421
	Acc Data de Fim	0.316	0.368	0.368	**0.395**	0.368	0.342
	Acc Nome do Curso	0.474	**0.500**	0.500	0.500	0.500	0.500
	Acc (Média)	0.412	0.430	0.430	**0.447**	**0.447**	0.421

Tabela 12 – Resultados de **Similaridade** por modelo, utilizando a estratégia "Segmentação por Sentenças".

		Modelos Comerciais				Modelos Gratuitos	
	Propriedade	**GPT-5 Mini**	**GPT-4.1 Mini**	**GPT-4o-mini**	**DeepSeek-v3**	**r1-0528**	**GPT-OSS 20b**
LCA	Sim Instituição	0.477	0.504	**0.591**	0.526	0.510	0.522
	Sim Data de Fim	0.684	0.664	**0.755**	0.663	0.363	0.454
	Sim Nome do Curso	**0.865**	0.841	0.861	0.773	0.779	0.674
	Similaridade Média	0.675	0.670	**0.736**	0.654	0.551	0.550
LCPG	Sim Instituição	0.734	0.730	0.730	0.730	0.685	0.610
	Sim Data de Fim	0.782	0.577	0.700	**0.738**	0.733	0.617
	Sim Nome do Curso	**0.603**	0.595	0.628	0.572	0.579	0.599
	Similaridade Média	**0.706**	0.634	0.686	0.680	0.666	0.609
ASS	Sim Instituição	0.668	0.640	**0.660**	0.642	0.652	0.637
	Sim Data de Fim	0.405	0.413	**0.434**	0.421	0.421	0.387
	Sim Nome do Curso	0.666	0.651	**0.663**	0.651	0.651	0.638
	Similaridade Média	0.580	0.568	**0.586**	0.571	0.575	0.554

5.8.1 Interpretação dos Resultados

A avaliação geral dos resultados, medida pela Similaridade Média na Tabela 12, posiciona esta estratégia em um patamar intermediário: consistentemente superior à fragmentação por tokens, mas ainda muito inferior às abordagens de contexto amplo. O GPT-4o-mini novamente se destaca como o modelo de melhor desempenho geral na maioria dos cenários (LCA e ASS), atingindo uma Similaridade Média máxima de 0.736 para o processo LCA. No entanto, este valor ainda representa uma queda de performance significativa em relação ao índice de 0.934 do mesmo modelo na estratégia Corpo Completo, indicando que a abordagem ainda é subótima.

A hipótese para este resultado intermediário é que, embora a segmentação por sentenças preserve a coerência de despachos individuais, ela ainda sofre de fragmentação de contexto de longo prazo. Muitas vezes, um parecer em uma parte do documento faz referência a informações apresentadas em um requerimento muito anterior e, ao analisar cada bloco isoladamente, o LLM perde essa conexão. Soma-se a isso o risco de quebras indevidas em nível local; de forma similar ao que ocorre na divisão por tokens, uma única informação complexa (como um curso e sua instituição) pode estar gramaticalmente distribuída em mais de uma sentença, resultando em uma extração incompleta quando os blocos são processados separadamente.

5.8.1.1 Análise por Propriedade

Fica evidente que mesmo as tarefas mais simples são prejudicadas: a extração de Nome do Curso, embora seja a de maior sucesso relativo nesta estratégia com índices que chegam a 0.865, representa uma queda de performance considerável em comparação com os resultados próximos a 0.990 obtidos nas estratégias de contexto amplo.

O problema se agrava em tarefas que exigem raciocínio mais complexo. Campos como Instituição (relacional) e Data de Fim (inferencial) apresentam resultados consistentemente baixos em todos os modelos. A extração de Data de Fim no processo ASS, por exemplo, ilustra a falha da estratégia: a Similaridade do GPT-4o-mini despenca de 0.897 na abordagem Corpo Completo para apenas 0.434 ao usar Sentenças, provando que a conexão entre informações distantes no texto é perdida. A hipótese, portanto, se consolida: a estrutura semântica dos processos não é contida em sentenças ou parágrafos isolados, mas distribuída ao longo de todo o documento, tornando as estratégias de segmentação fundamentalmente falhas para a extração de informações.

Em conclusão, a estratégia de Segmentação por Sentenças representa uma melhoria marginal sobre a fragmentação por tokens, mas seu desempenho ainda se mostra insatisfatório para a complexidade da tarefa. A performance consistentemente subótima em todos os modelos testados, incluindo os mais avançados, leva a uma conclusão importante: a falha é inerente à estratégia, e não aos modelos. A perda de contexto de longo prazo, como demonstrado pelos resultados, é um obstáculo intransponível para a extração de dados relacionais. Desta forma, a análise comparativa consolida a tese de que a preservação do contexto global, oferecida pelas abordagens de Corpo Completo e Requerimento, é um pré-requisito indispensável para se alcançar a alta fidelidade necessária nesta ferramenta de extração de informações.

5.9 Conclusões dos Resultados

A conclusão mais contundente deste estudo é que a estratégia de seleção de texto tem um impacto mais significativo na qualidade da extração do que a própria escolha do modelo de linguagem. A diferença de performance entre uma estratégia de contexto amplo e uma de fragmentação é muito maior do que a diferença entre um modelo comercial e um gratuito de ponta. Para ilustrar, o modelo gratuito GPT-OSS 20b com a estratégia Corpo Completo alcançou uma Similaridade Média de 0.901 no processo LCA, superando com folga o modelo comercial GPT-4o-mini quando este foi forçado a usar a estratégia de Tokens (0.682). Isso demonstra que investir em uma preparação de dados e em uma estratégia de contexto robusta é mais crucial do que simplesmente optar pelo modelo mais avançado.

Com base nos resultados, as estratégias de contexto amplo, Corpo Completo e Requerimento, provaram ser as únicas abordagens consistentemente viáveis para esta tarefa, enquanto as estratégias de segmentação (Tokens e Sentenças) se mostraram fundamentalmente inadequadas devido à perda de contexto relacional. A análise sugere que a Corpo Completo é a estratégia mais segura e robusta, especialmente para processos informacionalmente complexos e distribuídos como o LCA. A Extração Direcionada ao

Requerimento, por sua vez, se mostrou uma alternativa altamente eficaz para processos mais padronizados e autocontidos, como o LCPG, onde obteve resultados de alta precisão com menor custo computacional.

Na análise entre modelos, estabeleceu-se uma clara hierarquia. Os modelos comerciais, liderados pelo GPT-4o-mini, ofereceram a maior consistência e o melhor desempenho agregado, consolidando-se como a escolha ideal para aplicações que exigem máxima confiabilidade. No entanto, o estudo também demonstrou de forma conclusiva a viabilidade das alternativas gratuitas. Modelos como o DeepSeek-r1-0528-free e o GPT-OSS 20b, quando pareados com a estratégia Corpo Completo, entregaram um desempenho competitivo.

A conclusão principal é que a estratégia Corpo Completo se revelou a mais robusta para a extração de dados em processos complexos. Adicionalmente, a análise validou que, embora modelos comerciais como o GPT-4o-mini ofereçam a maior consistência, alternativas gratuitas de ponta entregam uma performance altamente competitiva. Fica demonstrado, portanto, que a combinação de uma estratégia de contexto amplo com um modelo gratuito de alto desempenho constitui uma solução técnica validada, acessível e eficaz para o módulo de extração de dados do projeto Aurora.

Por fim, é importante destacar que nossos achados se alinham de forma significativa ao trabalho (STUHLER; TON; OLLION, 2025). Assim como evidenciado em nossa análise, os autores também observaram que modelos pagos mantêm uma vantagem consistente em termos de confiabilidade e performance, mas que alternativas gratuitas, quando bem aplicadas, alcançam resultados próximos e satisfatórios. Esse paralelo reforça que, ainda que os modelos comerciais liderem em consistência, os modelos open-source já atingem níveis de desempenho suficientemente bons para viabilizar aplicações práticas em cenários de extração de informação.

6 Limitações do Trabalho

Para a correta interpretação dos resultados e conclusões apresentadas, é fundamental reconhecer as fronteiras e as limitações inerentes a este estudo. Estas limitações não invalidam os achados, mas contextualizam seu escopo de aplicabilidade e apontam caminhos para futuras investigações. As principais limitações são:

- **Escopo e Tamanho do Conjunto de Dados:** A análise foi realizada sobre um *dataset* de 138 processos, um número que, embora suficiente para os experimentos deste trabalho, é considerado baixo. Esta limitação se manifesta de duas formas: primeiro, a generalização dos resultados para outros tipos de processos administrativos não contemplados no estudo deve ser feita com cautela. Segundo, mesmo para os tipos de processo analisados (LCA, LCPG e ASS), o tamanho da amostra pode não ter capturado a totalidade de casos excepcionais e variações de redação encontradas nos pareceres e documentos que tramitam até a PRODIRH.
- **Classificação dos subtipos de Licença para Capacitação:** A classificação dos subtipos de Licença para Capacitação (LCA vs. LCPG), por exemplo, depende de uma lista fixa de palavras-chave. Um processo de pós-graduação com uma redação atípica, que não contenha os termos esperados, pode ser classificado incorretamente, levando à aplicação de um *prompt* de extração inadequado.

7 Conclusão

A complexidade e o volume crescentes dos processos administrativos representam um desafio contínuo para a eficiência da gestão pública. No contexto do Instituto Federal de Goiás (IFG), a sobrecarga de trabalho na análise de documentos pela Pró-Reitoria de Desenvolvimento Institucional e Recursos Humanos (PRODIRH) motivou este trabalho, que teve como objetivo central desenvolver e avaliar sistematicamente metodologias baseadas em Modelos de Linguagem de Grande Escala (LLMs) para a extração de informações estruturadas. A meta foi construir uma base metodológica, robusta e validada por meio de experimentação, para o módulo de extração do projeto Aurora.

A análise experimental, que comparou quatro estratégias de seleção de texto e seis diferentes LLMs, revelou que a escolha da estratégia de contexto é o fator mais crítico para o sucesso da extração, superando até mesmo a diferença de capacidade entre os modelos. Ficou demonstrado que abordagens que preservam um contexto amplo, como Corpo Completo e Requerimento, são vastamente superiores às que fragmentam o texto. A investigação também concluiu que não há uma estratégia universalmente ótima, sendo a Corpo Completo mais indicada para processos informacionalmente complexos (LCA e ASS) e a Requerimento para processos mais autocontidos (LCPG). Na comparação entre modelos, os LLMs comerciais de ponta confirmaram sua superioridade em consistência e robustez, mas o estudo validou de forma contundente a viabilidade de alternativas gratuitas de alto desempenho, como o eepSeek-R1-0528, que apresentaram resultados altamente competitivos.

Considera-se, portanto, que os objetivos propostos foram plenamente alcançados. O trabalho cumpriu sua meta ao avaliar sistematicamente diferentes métodos, resultando em contribuições significativas. A principal delas é a própria validação de uma metodologia baseada em engenharia de prompts como uma solução eficaz para a extração de dados no contexto da PRODIRH. Adicionalmente, a pesquisa gerou uma análise comparativa inédita que estabelece um benchmark de performance, não apenas entre diferentes LLMs, mas também entre as estratégias de texto investigadas. Por fim, a criação de um dataset de avaliação com 138 processos reais serve como um ativo fundamental que permitiu a condução deste estudo.

Apesar dos resultados positivos, este trabalho possui limitações que abrem caminhos promissores para pesquisas futuras. Uma direção natural seria investigar o ajuste fino (*fine-tuning*) de modelos no dataset aqui gerado, o que poderia criar um modelo mais especializado, com potencial aumento de precisão. A exploração de modelos inerentemente menores e mais eficientes também é uma via importante, visando a implementação

da solução em ambientes com recursos computacionais mais limitados e garantindo a escalabilidade da ferramenta.

Adicionalmente, a arquitetura poderia ser expandida com a implementação de um sistema de Geração Aumentada via Recuperação (RAG). No contexto específico de Extração de Informação, o RAG poderia ser utilizado para recuperar dinamicamente normas regulatórias específicas ou exemplos de documentos similares (técnica conhecida como *dynamic few-shot*), enriquecendo o contexto do prompt para resolver ambiguidades e aumentar a assertividade da extração.

Em suma, este trabalho não apenas validou a aplicação de LLMs para um problema prático e de alto impacto na administração pública, mas também produziu um *pipeline* de análise e um conjunto de dados que ficam disponíveis institucionalmente, servindo como base sólida para o avanço contínuo da automação inteligente dos processos de gestão de pessoas no Instituto Federal de Goiás.

8 Apêndice

8.1 Prompt para Licença para Cursos de Aprimoramento (LCA)

O código Python completo que implementa a montagem do prompt para os processos de Licença para Cursos de Aprimoramento (LCA) está disponível para consulta no seguinte endereço eletrônico:

<https://gist.github.com/danielVFS/2ea218ccdddc4a00e80592581aa2fe88>

8.2 Prompt para Licença para Cursos de Pós-Graduação (LCPG)

O código Python completo que implementa a montagem do prompt para os processos de Licença para Cursos de Pós-Graduação (LCPG) está disponível para consulta no seguinte endereço eletrônico:

<https://gist.github.com/danielVFS/fd18301b65e212edea4a7d3ed99929ff>

8.3 Prompt para Afastamento para Pós-Graduação Stricto Sensu (ASS)

O código Python completo que implementa a montagem do prompt para os processos de Licença para Afastamento para Pós-Graduação Stricto Sensu (ASS) está disponível para consulta no seguinte endereço eletrônico:

<https://gist.github.com/danielVFS/83417364be54543b7ce4bde329e0a7f4>

8.4 Código de Verificação para Licença para Cursos de Aprimoramento (LCA)

O código Python completo que implementa a função de verificação com perguntas de acompanhamento (verify_extraction), utilizada para validar os dados extraídos de processos do tipo Licença para Cursos de Aprimoramento (LCA), está disponível para consulta no seguinte endereço eletrônico:

<https://gist.github.com/danielVFS/096b2d262fd94495f82803c1faef551b>

8.5 Código de Verificação para Extração (Pós-Graduação)

O código Python completo que implementa a função de verificação com perguntas de acompanhamento (verify_pos_extraction), utilizada para validar os dados extraídos tanto de processos do tipo Licença para Cursos de Pós-Graduação (LCPG) quanto de Afastamento para Pós-Graduação Stricto Sensu (ASS), está disponível para consulta no seguinte endereço eletrônico:

<https://gist.github.com/danielVFS/b51a6366b7b4cc625966ffc344e1e956>

8.6 Função de Busca por Requerimento

O código Python completo que implementa a função get_documentos_requerimento_ou_justifi responsável pela estratégia de "Extração Direcionada ao Requerimento", está disponível para consulta no seguinte endereço eletrônico:

<https://gist.github.com/danielVFS/f0ae856d05c0c3111e854a09db9f4e49>

8.7 Função de Segmentação por Sentenças

O código Python completo que implementa a função split_into_sentences, responsável pela segmentação inteligente do texto em sentenças completas e pela tratativa de abreviaturas, está disponível para consulta no seguinte endereço eletrônico:

<https://gist.github.com/danielVFS/0b1b1e099f86e64799b951db2c44c330>

8.8 Função de Segmentação por Tokens

O código Python completo que implementa a função split_document_by_tokens, responsável pela estratégia de "Segmentação por Tokens de Tamanho Fixo", está disponível para consulta no seguinte endereço eletrônico:

<https://gist.github.com/danielVFS/e2176d7e55e69aa6b3885e3cb4997a4a>

8.9 Função de Comunicação com a API OpenRouter

A listagem a seguir apresenta o código em Python utilizado para encapsular as chamadas para a API do OpenRouter, permitindo a comunicação modular com os diferentes LLMs avaliados neste trabalho.

```python
import requests
import os

OPENROUTER_API_KEY = os.getenv("OPENROUTER_API_KEY")
BASE_URL = "https://openrouter.ai/api/v1/chat/completions"

def get_completion(prompt: str, model: str = 'deepseek/deepseek-chat:free',
                   temperature: float = 0):
    """Obtém a completude de um prompt usando um modelo específico via OpenRouter."""
    headers = {
        "Authorization": f"Bearer {OPENROUTER_API_KEY}",
    }
    payload = {
        "model": model,
        "messages": [
            {"role": "system", "content": "Você é um assistente especializado em extrair ..."},
            {"role": "user", "content": prompt}
        ],
        "temperature": temperature
    }
    try:
        response = requests.post(BASE_URL, headers=headers, json=payload)
        response.raise_for_status()
        data = response.json()
        return data["choices"][0]["message"]["content"].strip()
    except Exception as e:
        print(f"Ocorreu um erro na chamada da API: {e}")
        return None
```

Listing 10: Função para comunicação com os LLMs via API do OpenRouter.

Referências

BARNETT, S. et al. *Seven Failure Points When Engineering a Retrieval Augmented Generation System*. 2024. Disponível em: <https://arxiv.org/abs/2401.05856>. Citado na página 22.

BHAMBHORIA, R. et al. *Evaluating AI for Law: Bridging the Gap with Open-Source Solutions*. 2024. Disponível em: <https://arxiv.org/abs/2404.12349>. Citado na página 8.

BROWN, T. B. et al. *Language Models are Few-Shot Learners*. 2020. Disponível em: <https://arxiv.org/abs/2005.14165>. Citado 2 vezes nas páginas 14 e 17.

CASTRO, L. D.; FERRARI, D. Introdução à mineração de dados: Conceitos básicos, algoritmos e aplicações. *ISBN*, 2017. Citado na página 26.

CUI, Y.; GONG, Y.; CHENG, P.-J. *Effective and Efficient Schema-aware Information Extraction Using On-Device Large Language Models*. 2024. Citado na página 24.

DAI, Z. et al. *Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context*. 2019. Disponível em: <https://arxiv.org/abs/1901.02860>. Citado na página 21.

DAS, S. et al. *STRUCTSENSE: A TASK-AGNOSTOC AGENTIC FRAMEWORK FOR STRUCTURED INFORMATION EXTRACTION WITH HUMAN-IN-THE-LOOP EVALUATION AND BENCHMARKING*. 2024. Citado na página 24.

DUNN, A. et al. *Structured information extraction from complex scientific text with fine-tuned large language models*. 2022. Disponível em: <https://arxiv.org/abs/2212.05238>. Citado na página 23.

HOWARD, J.; RUDER, S. *Universal Language Model Fine-tuning for Text Classification*. 2018. Disponível em: <https://arxiv.org/abs/1801.06146>. Citado na página 23.

HUANG, L. et al. *A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions*. 2023. Citado na página 13.

JEBLICK, K. et al. *Evaluating local open-source large language models for data extraction from unstructured reports on mechanical thrombectomy in patients with ischemic stroke*. 2024. Citado na página 24.

JURAFSKY, D.; MARTIN, J. H. *Speech and language processing*. 3rd ed. draft. ed. [S.l.]: Prentice Hall, 2023. Citado na página 11.

LI, Y. et al. *Efficient Unified Information Extraction Model based on Large Language Models*. 2024. Citado na página 23.

LIU, N. F. et al. *Lost in the Middle: How Language Models Use Long Contexts*. 2023. Disponível em: <https://arxiv.org/abs/2307.03172>. Citado 2 vezes nas páginas 22 e 32.

LIU, P. et al. *Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing*. 2021. Disponível em: <https://arxiv.org/abs/2107.13586>. Citado na página 18.

MIN, S. et al. *Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?* 2022. Disponível em: <https://arxiv.org/abs/2202.12837>. Citado na página 15.

NADEAU, D.; SEKINE, S. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, John Benjamins, v. 30, n. 1, p. 3–26, 2007. Citado na página 14.

NG, A.; JORDAN, M. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In: DIETTERICH, T.; BECKER, S.; GHAHRAMANI, Z. (Ed.). *Advances in Neural Information Processing Systems*. MIT Press, 2001. v. 14. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2001/file/7b7a53e239400a13bd6be6c91c4f6c4e-Paper.pdf>. Citado na página 8.

POLAK, M. P.; MORGAN, D. Extracting accurate materials data from research papers with conversational language models and prompt engineering. *Nature Communications*, Springer Science and Business Media LLC, v. 15, n. 1, fev. 2024. ISSN 2041-1723. Disponível em: <http://dx.doi.org/10.1038/s41467-024-45914-8>. Citado 2 vezes nas páginas 15 e 34.

RAMSHAW, L. A.; MARCUS, M. P. *Text Chunking using Transformation-Based Learning*. 1995. Disponível em: <https://arxiv.org/abs/cmp-lg/9505040>. Citado na página 21.

REIS, J.; SANTO, P. E.; MELãO, N. Impacts of artificial intelligence on public administration: A systematic literature review. In: *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*. [S.l.: s.n.], 2019. p. 1–7. Citado na página 9.

SAHOO, P. et al. *A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications*. 2024. Disponível em: <https://arxiv.org/abs/2402.07927>. Citado na página 15.

SARAWAGI, S. Information extraction. *Found. Trends databases*, v. 1, n. 3, p. 261–377, 2008. Disponível em: <https://doi.org/10.1561/1900000003>. Citado na página 14.

SCHULHOFF, S. et al. *The Prompt Report: A Systematic Survey of Prompting Techniques*. 2024. Disponível em: <https://arxiv.org/abs/2406.06608>. Citado na página 15.

SHAKI, J.; KRAUS, S.; WOOLDRIDGE, M. Cognitive effects in large language models. In: _____. *ECAI 2023*. IOS Press, 2023. ISBN 9781643684376. Disponível em: <http://dx.doi.org/10.3233/FAIA230505>. Citado na página 12.

STUHLER, O.; TON, C. D.; OLLION, E. From codebooks to promptbooks: Extracting information from text with generative large language models. *Sociological Methods & Research*, v. 54, n. 3, p. 794–848, 2025. Disponível em: <https://doi.org/10.1177/00491241251336794>. Citado na página 54.

VASWANI, A. et al. *Attention Is All You Need*. 2017. Citado 2 vezes nas páginas 11 e 13.

WANG, Y. et al. *Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks*. 2022. Disponível em: <https://arxiv.org/abs/2204.07705>. Citado na página 17.

WEI, J. et al. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. 2023. Disponível em: <https://arxiv.org/abs/2201.11903>. Citado na página 19.

WEI, T.-R. et al. *A Survey on Feedback-based Multi-step Reasoning for Large Language Models on Mathematics*. 2025. Disponível em: <https://arxiv.org/abs/2502.14333>. Citado na página 19.

ZHAO, W. X. et al. *A Survey of Large Language Models*. 2023. Citado 2 vezes nas páginas 12 e 15.