Extração de Esquemas em Documentos Legais Não Estruturados Utilizando LLMs

Luan Alecxander Krzyzaniak[1], **Denio Duarte**[1],**Geomar A. Schreiner**[1], **Giancarlo Salton**[1]

[1]Universidade Federal do Rio Grande do Sul (UFFS) - Campus Chapecó
Caixa Postal 181 – 89815-899 – Chapecó – SC – Brazil

luan.krzyzaniak@estudante.uffs.edu.br, {duarte,gschreiner,gian}@uffs.edu.br

Resumo. *Este trabalho explora o uso de Large Language Models (LLMs) na extração de esquemas a partir de documentos legais (atas de registro de preços) não estruturados. O estudo propõe um pipeline para aplicação de LLMs na identificação e organização eficiente de informações jurídicas. Para avaliação, foi utilizado o modelo Mistral 7B Instruct v0.2 em 471 documenos, segmentadas em blocos de 2.000 tokens. Os resultados da análise por documento indicam alto desempenho na classificação de tipos de dados (Type Accuracy: 0,946; Type Precision: 0,972), mas desempenho semântico moderado (Semantic Accuracy: 0,412; Semantic Coverage: 0,714), revelando consistência na tipagem, porém limitações na correspondência semântica.*

Abstract. *This study investigates the use of Large Language Models (LLMs) for schema extraction from unstructured legal documents, specifically Price Registration Minutes (Atas de Registro de Preços). The research proposes a pipeline to analyze the efficacy of LLMs in identifying and systematically organizing legal information. For evaluation purposes, the Mistral 7B Instruct v0.2 model was deployed across 471 documents, segmented into blocks of 2,000 tokens. Per-document analysis results indicate high performance in data type classification (Type Accuracy: 0.946; Type Precision: 0.972). Conversely, the results demonstrate moderate semantic performance (Semantic Accuracy: 0.412; Semantic Coverage: 0.714), indicating consistency in data typing but highlighting limitations in semantic correspondence.*

1. Introdução

O sistema jurídico brasileiro gera diariamente um volume massivo de documentos legais, como petições, leis e contratos, que são armazenados em formatos predominantemente não estruturados [Aquino et al. 2024]. A análise desses documentos requer mão de obra significativa, tanto para a identificação de informações específicas quanto para a sua organização, o que frequentemente resulta em atrasos na tramitação e conclusão de processos judiciais [Aquino et al. 2024]. Embora técnicas de NLP já tenham sido empregadas em tarefas similares, ainda existem limitações quando se trata de lidar com documentos despadronizados e fortemente contextuais [Dagdelen et al. 2024]. Assim, a automatização desses processos se mostra um desafio aberto e de importante resolução.

Este trabalho tem como objetivo avaliar o desempenho de *Large Language Models* (LLMs) na extração de esquemas a partir de documentos não estruturados, com ênfase em

documentos legais, elaborando uma *pipeline* capaz de extrair a estrutura desses documentos[1]. Busca-se, com isso, reduzir o tempo de análise e organização de informações, aumentar a eficiência em processos jurídicos e demonstrar a eficácia da aplicação de LLMs em tarefas de processamento de linguagem natural voltadas a textos não estruturados.

Os testes conduzidos com uma coleção de Atas de Registros de Preços (ARPs) utilizando o modelo Mistral 7B Instruct v0.2 demonstraram que a abordagem proposta apresenta desempenho satisfatório na extração de esquemas. De modo geral, o modelo foi capaz de identificar a maior parte dos campos relevantes e manter a consistência na atribuição de tipos de dados, alcançando índices elevados de acurácia e precisão tipológica. Embora as métricas de similaridade semântica tenham evidenciado variação significativa entre os documentos, a cobertura média obtida indica que a estratégia adotada ainda é capaz de recuperar, de forma satisfatória, uma parcela substancial das informações dadas por um *golden schema*.

O restante do trabalho está organizado da seguinte forma: A Seção 2 apresenta os trabalhos relacionados e a Seção 3 apresenta o referencial teórico. Na Seção 4 apresenta a metodologia utilizada para a realização do trabalho e a Seção 5 apresenta os resultados obtidos. Por fim a Seção 6 apresenta as conclusões e trabalhos futuros são apontadas.

2. Trabalhos Relacionados

Nesta seção são apresentados trabalhos cujo tema, ferramentas, técnicas ou conceitos foram aplicáveis ou relevantes para a realização deste trabalho. Os artigos englobam temas relacionados a textos não estruturados, técnicas de extração de esquemas, aplicações de técnicas de melhoramento de LLMs como *fine-tuning* e *Retrieval-Augmented Generation* (RAG), além de estudos a respeito de documentos legais.

O trabalho de [Dagdelen et al. 2024] trata da automação da extração de informações em textos científicos com foco em compostos químicos e seus atributos. O maior interesse é expandir sobre as técnicas de NLP para o reconhecimento de entidades nomeadas (NER) e extração de relações (RE) que podem ser aplicadas para esta tarefa, mas que não apresentam resultados satisfatórios isoladamente. A utilização de LLMs como GPT-3 e Llama-2 também é explorada, mas apresentam resultados insatisfatórios devido ao menor treinamento em campos científicos menos populares, segundo o autor, resultado da baixa quantidade de conteúdo computacionalmente acessível disponível sobre estes temas. Para resolver o problema, os autores propõem a integração da extração da técnica NER com relações de entidade (apelidado NERRE) ao uso de LLMs, nomeado LLM-NERRE. Os autores utilizaram a técnica com os modelos GPT-3 e Llama-2 para realizar testes em três tarefas de extração de compostos químicos. As métricas de precisão para qualquer uma das três atividades atingiram 0.87 no pior caso, resultados satisfatórios que demonstram a usabilidade do LLM-NERRE, segundo o autor.

Já no trabalho de [Zhang and Soh 2024], é proposta uma solução para automatizar a criação de *knowledge graphs* para grandes volumes de textos sem esquemas estabelecidos (isto é, dados não estruturados) utilizando LLMs. Para solucionar esse problema, é proposto um *framework* que implementa a LLM em uma *pipeline* de três passos nomeada *Extract-Define-Canonicalize* ou *EDC*. O *framework* foi aplicado em três *datasets*

[1]Projeto disponível em github.com/LuanKrzyzaniak/tcc

distintos e, segundo o autor, demonstrou resultados superiores ou em par com técnicas modernas. Os testes foram realizados nas LLMs GPT-4, Mistral-7b e GPT-3.5-Turbo, sendo a aplicação com GPT-4 aquela com melhores resultados.

Os autores [Lewis et al. 2020] exploram LLM com *Retrieval-Augmented Generation* (RAG) para área médica de duas formas: o RAG-*sequence*, que realiza a pesquisa externa uma vez para cada sequência completa de *tokens*, e o RAG-*token* realiza uma pesquisa única para cada *token* analisado. Os modelos foram testados por uma bateria de perguntas classificadas como *knowledge-intensive*, ou seja, que exigem a análise de grande quantidade de informações para serem respondidas. Segundo os autores, os resultados de ambas as arquiteturas RAG nestes testes superaram tanto modelos que utilizam apenas LLMs quanto implementações baseadas em recuperar e extrair, demonstrando sua aplicabilidade.

Em sequência, o trabalho de [Aquino et al. 2024] apresenta o uso do *RAG* em documentos legais brasileiros. Os autores exploram uma alternativa para a extração de dados específicos de documentos legais utilizando LLMs auxiliadas pelo método *RAG*. A extração foi realizada em dois conjuntos de dados específicos propostos, e os resultados atingidos se mostraram satisfatórios, segundo os autores. A *pipeline* demonstrou um resultado de 90% de acurácia, superando modelos que utilizam expressões regulares, que atingiram média de 59% de acerto.

A abordagem proposta neste trabalho se distingue das apresentadas por extrair a estrutura de documentos textuais sem algum pré-processando em LLM (e.g., RAG e *fine-tuning*) para mitigar o custo computacional.

3. Referencial Teórico

Esta seção apresenta brevemente os conceitos necessários para a compreensão da proposta deste trabalho. Inicialmente, é apresentado o conceito de documentos legais, seguido do conceito de esquemas e finalizando com uma discussão sobre LLMs.

Documentos legais englobam todos os documentos utilizados por operadores do direito, como advogados, juízes e tribunais, além de incluir legislação, doutrina e jurisprudência. Englobam-se a Constituição Federal, leis, mandatos, contratos, petições, procurações e inúmeros outros modelos que possuam valor legal [Lima and Flores 2016].

O enorme volume de procurações públicas, cerca de 200 mil entre 2020 e 2023, justifica a necessidade de padronização de documentos legais [Aquino et al. 2024]. A facilidade na leitura e compreensão agiliza o andamento de processos no judiciário e, além disso, a padronização da formatação das leis reduz a necessidade de interpretações subjetivas, garantindo que sejam "redigidas com clareza, precisão e ordem lógica", conforme determina a Lei Complementar nº 95/1998, que reforça a Constituição Federal de 1988.

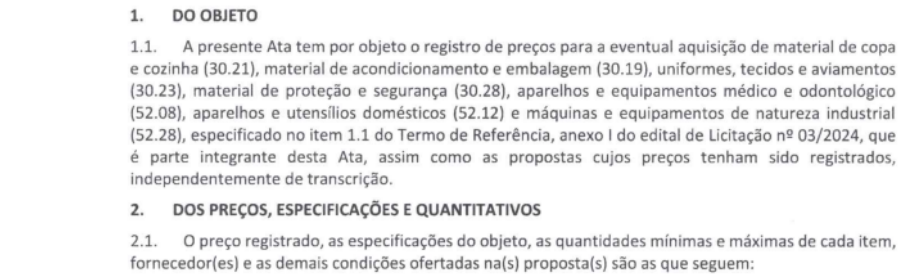
O conhecimento da estrutura de cada documento legal é necessário para a compreensão, análise e identificação de informações importantes e relevantes aos processos da União. Sua importância se estende ao âmbito digital, onde uma estrutura formal e respeitada facilita a extração e o tratamento desses textos, uma atividade demorada que pode ser automatizada, dado um documento bem estruturado.

A identificação da estrutura (esquema) em dados estruturados, como tabelas relacionais, tende a ser direta devido ao formato rigidamente padronizado. Entretanto, essa ta-

refa se torna mais complexa em dados semi-estruturados e ainda mais desafiadora em dados não estruturados [Christopher et al. 2022]. Em dados semi-estruturados, como JSON e XML, os metadados estão embutidos nos próprios valores. Essa característica proporciona flexibilidade, mas também aumenta a variabilidade estrutural. Um mesmo arquivo JSON pode conter objetos completamente distintos, dificultando a inferência automática de um esquema global [Banhara et al. 2024].

Dados não estruturados, principalmente textuais, são amplamente predominantes na internet — artigos, contratos, jornais, postagens e documentos administrativos. A extração de informações estruturais nesses cenários requer interpretação contextual, pois os metadados podem aparecer de formas variadas ou mesmo sem rótulo explícito. Além disso, diferentes documentos podem expressar a mesma categoria de informação de maneiras heterogêneas (por exemplo, ocupações como `Representante`, `Estudante` ou `Escritor`).

A Figura 1 ilustra um trecho de uma ARP que evidencia a dificuldade de identificar esquemas em texto corrido. O Item 1.1, por exemplo, descreve os itens cujos preços estão registrados na ata. Assim, é necessário, dentro do texto, encontrar a referência dos itens que estão sendo registrados.



1. DO OBJETO

1.1. A presente Ata tem por objeto o registro de preços para a eventual aquisição de material de copa e cozinha (30.21), material de acondicionamento e embalagem (30.19), uniformes, tecidos e aviamentos (30.23), material de proteção e segurança (30.28), aparelhos e equipamentos médico e odontológico (52.08), aparelhos e utensílios domésticos (52.12) e máquinas e equipamentos de natureza industrial (52.28), especificado no item 1.1 do Termo de Referência, anexo I do edital de Licitação nº 03/2024, que é parte integrante desta Ata, assim como as propostas cujos preços tenham sido registrados, independentemente de transcrição.

2. DOS PREÇOS, ESPECIFICAÇÕES E QUANTITATIVOS

2.1. O preço registrado, as especificações do objeto, as quantidades mínimas e máximas de cada item, fornecedor(es) e as demais condições ofertadas na(s) proposta(s) são as que seguem:

Figura 1. Exemplo da cláusula de um contrato de compra de uma casa com dados fictícios.

Modelos de Linguagem de Grande Escala são modelos computacionais baseados em técnicas avançadas de aprendizado profundo, desenvolvidos para interpretar e gerar textos de maneira humana ou estruturada. Estes modelos surgiram como evolução dos primeiros modelos estatísticos de linguagem, como os modelos n-gramas, e posteriormente dos modelos baseados em redes neurais recorrentes (RNNs) e LSTMs[2]. Com o avanço da arquitetura *Transformer* introduzida por [Vaswani et al. 2017], tornou-se possível escalar o treinamento de modelos de linguagem para níveis antes impraticáveis, tanto em termos de quantidade de dados quanto de complexidade de tarefas.

Com o crescimento no tamanho dos modelos, observou-se o surgimento de capacidades emergentes, como raciocínio em múltiplas etapas, compreensão de instruções, geração de código e até mesmo interações multimodais. Essas capacidades, não explicitamente programadas, desafiam a intuição tradicional sobre o funcionamento de sistemas computacionais e trazem novas perspectivas sobre a interface entre linguagem e inteligência artificial [Wei et al. 2022]. Apesar de seu bom desempenho em uma ampla gama

[2]LSTM, ou *Long Short-Term Memory*, é um tipo de arquitetura de rede neural recorrente (RNN) projetada para lidar com o problema da dependência de longo prazo em dados sequenciais, como séries temporais

de tarefas, os LLMs ainda enfrentam limitações relevantes, como alucinação de fatos, viés nos dados de treinamento e elevado custo computacional [Bender et al. 2021].

4. Metodologia

A metodologia adotada neste trabalho inicia com as etapas de coleta, pré-processamento e preparação do conjunto de dados, além da escolha e preparação do modelo.

A construção do conjunto de dados teve início com a coleta de documentos legais disponibilizados em portais oficiais do governo, particularmente Atas de Registro de Preço (ARP) do Portal de Transparência[3]. As ARPs possuem esquemas semelhantes, mas diferem em estrutura e representação de dados, o que as torna adequadas para o estudo da tarefa de extração de esquemas e, sobretudo, permite a comparação via *embedding* utilizando um *golden schema* como base (ambos detalhados a seguir).

A pesquisa e a coleta inicial ocorreram de forma manual, com o objetivo de compreender a estrutura e a distribuição dos documentos nos diferentes portais. Após essa etapa exploratória, a automação foi implementada para ampliar significativamente o volume de dados e garantir maior controle sobre os critérios de seleção. Para isso, foi utilizado o *framework Selenium*[4], escolhido por sua capacidade de simular interações humanas com navegadores e lidar com páginas que dependem fortemente de *JavaScript* ou carregamento dinâmico de conteúdo.

Após concluída a coleta automatizada, os documentos passaram por um processo de pré-seleção para garantir sua usabilidade. Arquivos duplicados ou compostos majoritariamente por imagens digitalizadas foram excluídos, uma vez que não estão contidos no escopo do trabalho, resultando em um conjunto de 471 documentos. Os arquivos resultantes passaram posteriormente por etapas de limpeza, normalização e preparação textual, detalhadas na próxima seção. O pré-tratamento textual buscou tratar a heterogeneidade dos documentos coletados via *web scraping*. As ARPs apresentam formatos variados, problemas de digitalização, padrões inconsistentes e textos com ruídos típicos de PDFs governamentais. Para garantir uma qualidade uniforme nos dados enviados ao modelo de linguagem, foi utilizada uma *pipeline* que limpa, padroniza e organiza o texto sem perda de dados relevantes, implementada por meio de um script elaborado na linguagem Python.

Uma parte importante do processo foi a tentativa de normalização de valores monetários. Inicialmente, buscou-se padronizar preços e valores financeiros, já que são elementos centrais nas ARPs. No entanto, essa abordagem mostrou-se inviável. Muitos números presentes nos documentos possuem um padrão de escrita similar a valores monetários, como CPFs, CNPJs, telefones, códigos de itens e percentuais. A ausência frequente do prefixo *R$* impediu diferenciar, de forma segura, valores financeiros de outras entidades numéricas. Mesmo tentando restringir a normalização pelo contexto, os documentos apresentaram muitas variações para uma detecção confiável. Para evitar modificações incorretas em dados sensíveis, especialmente identificadores pessoais e referências administrativas, esta etapa do processamento foi abandonada e esses valores foram preservados em suas formas originais.

[3]https://contratos.sistema.gov.br/transparencia

[4]https://www.selenium.dev/

Depois da limpeza e normalização, o texto das páginas foi reunido em um único bloco contínuo, garantindo que a LLM tenha acesso sequencial completo ao conteúdo do documento. Com essa pipeline, foi possível padronizar os documentos sem comprometer informações críticas, reduzindo ruídos e mantendo a consistência necessária para a análise e para a avaliação do desempenho da LLM.

Para os experimentos, foi escolhido o *Mistral 7B Instruct v0.2*, um modelo de código aberto projetado para ser eficiente em tarefas de *finetuning* [Jiang et al. 2023]. Durante testes empíricos o modelo apresentou desempenho satisfatório na identificação e extração dos esquemas propostos no *dataset* elaborado, superando outras alternativas testadas, como o *T5-base* da *Google*.

O Mistral 7B emprega um tokenizador baseado em *SentencePiece* com *Byte-Pair Encoding* (BPE). Esse tipo de tokenização, semelhante à adotada pela família de modelos LLaMA, opera no nível de subpalavras, permitindo que o modelo represente eficientemente tanto termos jurídicos específicos quanto estruturas numéricas e padrões textuais. Também, a escolha do Mistral 7B também se justifica por sua boa relação entre desempenho e custo computacional [Jiang et al. 2023]. Diferentemente de modelos de maior porte, como o Llama 3 70B ou o GPT-4, o Mistral 7B oferece boa precisão com requisitos computacionais mais modestos, tornando viável sua utilização no contexto desta pesquisa, que opera sob limitações de hardware que impedem a utilização de modelos computacionalmente custosos.

O próximo passo foi criar um esquema que servisse como base para verificar a corretude das estruturas extraídas. Este esquema foi chamado de *golden schema*. O *golden schema* consiste em uma estrutura JSON que identifica os campos de interesse presentes nas Atas de Registro de Preços (ARPs) e define os tipos de dados esperados para cada campo. Sua elaboração foi realizada manualmente, com base na análise de quinze documentos representativos coletados aleatoriamente da mesma fonte do dataset. A análise permitiu identificar informações recorrentes e relevantes, como número da ata, datas, órgão gerenciador, fornecedores e itens registrados. Campos complexos, como listas de fornecedores e itens, foram modelados como listas de objetos, enquanto informações compostas, como dados do órgão gerenciador ou períodos de vigência, foram representadas como objetos aninhados. Como resultado, o *golden schema* ficou composto por 25 campos. A construção do *golden schema* permitiu criar uma referência consistente para avaliação do modelo, fornecendo tanto a identificação dos campos quanto os tipos de dados esperados, essenciais para o cálculo de métricas como *Type Accuracy*, *Type Precision* e similaridade semântica. A Figura 2 apresenta uma versão reduzida do *golden schema* utilizado neste trabalho, destacando apenas os diferentes tipos estruturais presentes no conjunto completo. O exemplo inclui: chaves simples, representando campos simples como `string` (Linha 2); objetos, que unem propriedades internas organizadas em uma hierarquia (Linhas 4-9); e listas (*arrays* - Linha 12), utilizadas para representar coleções de elementos estruturados, como conjuntos de fornecedores ou itens registrados.

O próximo passo foi a definição de um *prompt* para o modelo (Figura 3). Ele foi desenvolvido para guiar a LLM na identificação e estruturação das informações mais relevantes presentes nos documentos, respeitando a forma definida pelo *golden schema*. A elaboração do *prompt* focou em garantir consistência nos resultados produzidos pelo modelo, tornando possível a comparação direta com o *golden schema* e permitindo uma

```
{
  "numero_ata": {"type": "string"},
  ...
  "vigencia": {
    "type": "object",
    "properties": {"data_inicio": { "type": "date" },
                   "data_fim": { "type": "date" }}},
...
"fornecedores_registrados": {
  "type": "array",
  "items": { "type": "object",
             "properties": {"razao_social": {"type": "string"},
                            "cnpj": { "type": "string" }}}}
...}
```

Figura 2. Exemplo simplificado do *golden schema*

```
Analise o texto abaixo e extraia um esquema JSON que descreva os dados mais importantes encontrados no documento.
O esquema deve seguir o formato chave-valor, com o nome do campo e o tipo de dado (inteiro, string, lista, etc.).
Regras:
- Identifique apenas informações relevantes e recorrentes.
- Use tipos de dado genéricos: "string", "integer", "float", "date", "boolean", "percent".
- Não preencha valores, apenas defina o tipo esperado.
- Se o texto sugerir listas ou tabelas, represente-as com "array" e especifique o tipo dos elementos.
- Use o formato JSON Schema (com "type" e "properties" quando aplicável).
- Não repita este prompt nem adicione comentários, explicações ou texto além do JSON.
- Retorne apenas o JSON puro e válido.

Exemplo de estruturas esperadas:
{ "documento": { "type": "string" },
  "fornecedor": {"type": "object",
                 "properties": {"nome": { "type": "string" },
                  "valor": { "type": "integer" }}},
  "itens": {"type": "array",
            "items": {"type": "object",
                      "properties": {
                      "descricao": { "type": "string" },
                      "quantia": { "type": "integer" }}}}
}
```

Figura 3. Prompt utilizado nos testes

avaliação objetiva da capacidade da LLM em extrair informações estruturadas de documentos legais complexos.

Finalmente, foram realizados os experimentos para avaliar o desempenho do Mistral 7B na extração de esquemas a partir das ARPs previamente coletadas e submetidas ao processo de pré-tratamento textual.

Para viabilizar a execução da LLM em ambiente local, os testes foram realizados em uma máquina equipada com uma GPU NVIDIA RTX 4070 Ti, com 8 GB de memória dedicada. Devido ao tamanho do modelo e às limitações de hardware, foi empregada quantização em 4 bits durante o carregamento da LLM, permitindo reduzir o custo de memória do modelo. Essa técnica tornou possível o processamento dos documentos em tempo hábil e dentro da capacidade computacional disponível.

O Mistral 7B possui um limite de contexto aproximado de 8.000 *tokens*, o que inviabiliza o envio de documentos extensos em uma única chamada ao modelo. Para contornar essa restrição, o conteúdo de cada documento foi segmentado em blocos de 2.000 *tokens*, gerados por meio do tokenizador nativo do próprio Mistral. Essa abordagem garantiu que cada parte do documento fosse processada integralmente, ao mesmo tempo em que evitou truncamentos automáticos decorrentes de alto consumo do contexto da LLM pela junção do *prompt* e do texto do documento, o que poderia comprometer a extração das informações.

```
1 {"orgao_gerenciador": {
2   "type": "object",
3   "properties": {
4     "email": { "type": "string" },
5     "telefone": { "type": "string" },
6     "endereco": { "type": "object",
7                   "properties": {
8                   "bairro":   { "type": "string" },
9                   "cep":      { "type": "string" }}}},
10   :
11  "fornecedores": {
12    "type": "array",
13    "items": {
14      "type": "object",
15      "properties": {
16       "registro_de_precos": {
17         "type": "object",
18         "properties": {"cancelado": { "type": "boolean" },
19                        "precos": {
20                        "type": "array",
21                        "items": {"type": "object",
22                                  "properties": {"item": { "type": "string"},
23                                                 "preco": { "type": "float"}}}}}}}
24  },
25  :
26 "cancelamento": {"type": "object", "properties": {"motivo": { "type": "string" }}},
27 ...}
```

Figura 4. Exemplo simplificado do esquema unificado gerado pelo modelo

Após a execução completa do processo de extração, cada documento teve seu esquema reconstruído a partir dos *chunks* processados pela LLM, resultando em um único JSON por documento. Esse JSON preserva as chaves e estruturas de todos os chunks, integrando todas as informações extraídas ao longo do documento, garantindo uma representação completa e unificada do esquema produzido pela LLM.

Com os JSONs individuais prontos, foi realizada a primeira etapa de avaliação, na qual cada documento foi comparado diretamente com o *golden schema*. Nessa fase, foram calculadas métricas de acurácia semântica, cobertura semântica, F1 semântico, acurácia de tipos e precisão de tipos. Essas métricas permitem analisar, em uma média por documento, o quão próximo o modelo chegou da estrutura esperada.

A Figura 4 apresenta um trecho do esquema resultante em que foram extraídos, por exemplo, três objetos complexos: (*i*) `orgao_gerenciador` (linhas 1-9), (*ii*) `fornecedores` (linhas 11-26), que também é um *array* , e (*iii*) `cancelamento` (linhas 29-31). Apesar do esquema apresentado na Figura 4 parecer estruturalmente correto, é necessário verificar se a semântica representa a coleção dos documentos de entrada. Na próxima seção, são analisados os resultados.

5. Resultados

A análise qualitativa do desempenho do modelo foi realizada comparando os JSONs extraídos com o *golden schema* de referência, utilizando métricas de similaridade semântica e acurácia de tipos. Para isso, aplicou-se um modelo de *embeddings* local (*all-MiniLM-L6-v2*) para calcular as similaridades entre pares de campos esperados e extraídos, utilizando a métrica de similaridade do cosseno. Essa abordagem permitiu identificar correspondências exatas (*semantic_good*) e parciais (*semantic_partial*), além de avaliar a correção dos tipos de dados previstos pelo modelo. Os resultados gerais que englobam a média resultante da comparação individual do JSON de cada documento encontram-se resumidos na Tabela 1.

A *Semantic Accuracy* de $0,412$ indica que aproximadamente 41% dos campos

Tabela 1. Métricas gerais de desempenho do Mistral 7B Instruct v0.2 na extração de esquemas de ARPs por documento.

Métrica	Valor médio
Semantic Accuracy	$0,412$
Semantic Coverage	$0,714$
Semantic F1	$0,519$
Type Accuracy	$0,946$
Type Precision	$0,972$
Campos previstos (média)	$30,9$

previstos pelo modelo correspondiam semanticamente aos campos do *golden schema*, considerando apenas correspondências com similaridade igual ou superior a $0,75$. Já a *Semantic Coverage* de $0,714$ revela que cerca de 71% dos campos do esquema de referência foram cobertos, incluindo correspondências parciais com similaridade entre $0,60$ e $0,75$. O valor intermediário da *Semantic F1* $(0,519)$ evidencia um equilíbrio entre precisão semântica e cobertura.

Em termos de identificação dos tipos de dados, o modelo apresentou um bom desempenho. A *Type Accuracy* média de $0,946$ indica que $94,6\%$ dos campos extraídos pelo modelo possuíam tipos que correspondiam aos do *golden schema*. A *Type Precision* de $0,972$ mostra que, entre os campos semanticamente bem mapeados (*semantic_good*), $97,2\%$ apresentavam o tipo correto, evidenciando a consistência do modelo na atribuição de tipos. Adicionalmente, o modelo gerou em média $30,9$ campos por documento, valor próximo ao número de campos do *golden schema*, de 25. Isso indica que a LLM conseguiu extrair grande parte das informações relevantes, com um valor aceitável de ruído, embora algumas correspondências semânticas não tenham atingido o limiar de similaridade máxima.

Os experimentos realizados mostram que o Mistral 7B desempenhou um papel satisfatório na extração de estruturas de ARPs. Salienta-se, também, que o modelo é de tamanho médio em relação aos parâmetros (sete bilhões) e, portanto, possui um menor custo computacional, exigindo hardware convencional para a execução.

6. Conclusão

O presente trabalho avaliou a capacidade do modelo Mistral 7B Instruct v0.2 na extração de esquemas estruturados a partir de documentos legais não estruturados, especificamente, Atas de Registro de Preços (ARPs).

Os testes mostraram que o modelo é capaz de identificar e organizar a maior parte das informações relevantes, produzindo JSONs consistentes e com uma tipagem correta entre $94,6\%$ e $97,2\%$ dos campos, dependendo da métrica considerada. A *Type Accuracy* e a *Type Precision*, elevadas em ambas as etapas do teste, apontam a robustez do modelo na classificação dos tipos de campos, mesmo quando o conteúdo dos documentos é complexo ou variado.

Por outro lado, a avaliação semântica, representada pela *Semantic Accuracy* e pela *Semantic Coverage*, demonstrou limitações na correspondência entre os campos extraídos e o *golden schema*, mesmo utilizando técnicas como *embedding*. Isso indica que, embora

o modelo seja capaz de estruturar dados de forma coerente, a compreensão semântica completa ainda não é satisfatória para a automação sem supervisão humana.

Dessa forma, este estudo conclui que o Mistral 7B pode agilizar a preparação de dados e reduzir parte do esforço manual na extração de esquemas de documentos não estruturados, especialmente no âmbito legal, atuando como ferramenta auxiliar. Entretanto, sua aplicação prática exige estratégias de verificação, limpeza e correção semântica, garantindo que as informações extraídas sejam não apenas estruturadas, mas também semanticamente corretas.

Como direção de trabalhos futuros, propõe-se a exploração de aprimoramentos no modelo, por meio de RAG e *fine-tuning*, bem como o aprimoramento do pós-processamento dos esquemas, visando aumentar a correspondência de chaves e reduzir a necessidade de intervenção manual.

Referências

Aquino, I., dos Santos, M. M., Dorneles, C., and Carvalho, J. T. (2024). Extracting information from brazilian legal documents with retrieval augmented generation. In *Anais Estendidos do XXXIX SBBD*, pages 280–287, Porto Alegre, RS, Brasil. SBC.

Banhara, N., Schreiner, G. A., da Silva Feitosa, S., and Duarte, D. (2024). Enumeration, tagged unions, tuples, and collections: A novel approach to extracting json schema. In *Simpósio Brasileiro de Banco de Dados (SBBD)*, pages 234–246. SBC.

Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of ACM FAccT*, pages 610–623.

Christopher, C., Moore, K., and Liebowitz, D. (2022). *SchemaDB: A Dataset for Structures in Relational Data*, page 233–243. Springer Nature Singapore.

Dagdelen, J. et al. (2024). Structured information extraction from scientific text with large language models. *Nature communications*, 15(1):1418.

Jiang, A. Q. et al. (2023). Mistral 7b.

Lewis, P. et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In *34th International Conference on Neural Information Processing Systems*, NIPS '20.

Lima, E. d. S. and Flores, D. (2016). A evolução da legislação relacionada à digitalização e aos documentos digitais no âmbito da administração pública federal. *Revista Sociais e Humanas*, 29(1):75–91.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Wei, J. et al. (2022). Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Zhang, B. and Soh, H. (2024). Extract, define, canonicalize: An LLM-based framework for knowledge graph construction. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.