1. Introdução

Modelos de linguagem em larga escala (*Large Language Models* LLMs), como GPT, LLaMA e T5, têm revolucionado o campo do Processamento de Linguagem Natural (NLP) ao alcançar desempenhos impressionantes em tarefas variadas, como geração de texto, sumarização e perguntas e respostas (*Question Answering* QA). Contudo, sua aplicação a dados estruturados, como tabelas, permanece um desafio devido à natureza compacta e sem contexto dessas informações. Diferentemente de textos narrativos, que possuem conectivos e semântica explícita, tabelas dependem de relações implícitas entre colunas e linhas para transmitir significado.

Dada essa lacuna, várias abordagens têm sido propostas para integrar dados tabulares com LLMs. Métodos tradicionais frequentemente dependem de arquiteturas especializadas para dados tabulares ou do uso direto de formatos estruturados, como CSV, que, embora eficazes em alguns cenários, podem limitar a capacidade do modelo de interpretar nuances e contextos semânticos mais ricos. Um exemplo disso é o modelo OLLAMA 3 8B, que foi projetado para processar diretamente dados tabulares em formato CSV, mas sem transformar as tabelas em texto narrativo.

Neste trabalho, propomos uma metodologia alternativa que utiliza o poder do *Prompt Engineering* para transformar dados tabulares em entradas textuais contextualizadas. Denominada **TabletoContext,** nossa abordagem gera automaticamente templates narrativos a partir da descrição das colunas e do conteúdo das tabelas. Esses templates enriquecem o contexto semântico das tabelas, utilizando conectivos e construções linguísticas que tornam os dados mais compreensíveis para LLMs, permitindo que eles processem informações tabulares como se fossem texto narrativo.

Como contribuição principal, este artigo apresenta:

- Uma metodologia para converter tabelas em texto narrativo de forma automatizada e contextualizada, utilizando descrições semânticas das colunas e linhas.
- Análise detalhada dos ganhos e limitações de ambas as abordagens, explorando como a contextualização textual pode superar formatos estruturados em cenários específicos.

Este trabalho não apenas explora os desafios de integrar tabelas com LLMs, mas também demonstra como a geração de contexto narrativo pode expandir o alcance de modelos generalistas, oferecendo novas perspectivas para o uso de LLMs em dados estruturados.

Para esse estudo usamos 5 datasets do DataBench, que é um benchmark apresentado no artigo [1].

DataBench é um grande benchmark para a tarefa de resposta a perguntas tabulares em dados estruturados ou tabulares. No artigo acima foi proposto o DataBench com o objetivo de fornecer um benchmark para avaliar e comparar LLMs como raciocinadores tabulares, mas flexível para comparar qualquer outro tipo de modelo de resposta a perguntas. Consequentemente, o DataBench é composto por 65 conjuntos de dados de diferentes domínios, números amplamente diferentes de linhas e colunas e tipos de dados heterogêneos. Além disso, o DataBench possui 20 questões feitas à mão por conjunto de dados, com um número total de 1.300

questões. As perguntas são divididas em diferentes tipos, dependendo do tipo de resposta (ou seja, verdadeiro/falso, categorias do conjunto de dados, números ou listas), e cada pergunta é acompanhada pela resposta padrão-ouro correspondente.

Dentre as bases de dados do benchmark supracitado escolhemos cinco conjuntos de dados para aplicação em nosso trabalho. Essas bases de dados estão dentro do domínio "Travel and Locations". O critério foi selecionar os 5 datasets de menor tamanho (linhas x colunas):

- **Titanic (002_Titanic):**
  Esta tabela contém dados relacionados aos passageiros do Titanic, incluindo características como idade, sexo, classe de passagem, tarifa paga, e se sobreviveram ou não ao naufrágio. Os dados são comumente usados para análises de sobrevivência e exploração de padrões como relação entre classe social e taxa de sobrevivência

- **Holiday Package Sales (016_Holiday):**
  Essa tabela representa informações sobre vendas de pacotes de férias, incluindo atributos como destinos, preços, categorias de pacote, e avaliações de clientes. O foco está na análise de vendas e preferências dos consumidores

- **Disneyland Customer Reviews (061_Disneyland):**
  Esta tabela reúne análises de clientes sobre a experiência na Disneyland, com campos como comentários, pontuações de satisfação, categorias de avaliação (como limpeza e acessibilidade) e datas das visitas.

- **Central Park (009_Central):**
  Este conjunto de dados detalha informações sobre atividades, eventos e medições ambientais no Central Park, como tipos de eventos, número de visitantes e dados meteorológicos coletados durante o período de análise.

Finalmente, usamos o DataBench para avaliar o OLLAMA 3 com 8B de parâmetros, sobre dados tabulares, incluindo a nossa abordagem tabletoContex. Durante a análise dos resultados, percebemos que existiam algumas inconsistências nas respostas do Benchmark, portanto separamos uma seção para falar desses casos

2. Motivação

A motivação para este trabalho partiu do **SemEval** uma competição anual que nos propomos a participar esse ano. O **SemEval** (Semantic Evaluation) é uma série anual de competições focadas em desafios no campo do processamento de linguagem natural (PLN). Ele busca promover o avanço técnico na área ao propor tarefas relacionadas à análise semântica, como interpretação de significado, análise de sentimento, detecção de ironia e compreensão de contexto em linguagem natural.

É amplamente reconhecido por oferecer benchmarks padronizados, permitindo a comparação de modelos e algoritmos em tarefas específicas.

Cada edição apresenta um conjunto diversificado de tarefas que desafiam equipes a solucionar problemas com datasets disponibilizados pela organização. Esses dados são geralmente extraídos de fontes reais, como redes sociais, notícias ou bancos de dados temáticos, e exigem abordagens inovadoras para lidar com o contexto, ambiguidade e complexidade semântica.

Nossa equipe selecionou a tarefa ”**Task 8: Question Answering on Tabular Data**” da edição de 2025, uma das principais tarefas será resolver problemas de **Question Answering** sobre dados tabulares, utilizando o benchmark **DataBench**. Essa tarefa exige que os participantes criem sistemas capazes de responder perguntas de forma precisa, realizando análises que combinam processamento textual e manipulação de tabelas. A dificuldade aumenta pela necessidade de lidar com diferentes formatos de perguntas e exigências semânticas específicas. Essa tarefa está dividida em duas subtarefas e decidimos trabalhar nessas duas subtarefas.

Subtarefa I: Os participantes receberão um conjunto de dados (de qualquer tamanho) e uma pergunta sobre ele. A pergunta deve ser respondida utilizando apenas os dados do conjunto fornecido.

Subtarefa II: A tarefa é essencialmente a mesma da subtarefa anterior, mas envolve o uso de uma versão amostrada de cada conjunto de dados, com no máximo 20 linhas por conjunto (consulte a explicação sobre o DataBench Lite). A pergunta deve ser respondida utilizando apenas os dados do conjunto amostrado. Para o conjunto de teste, também forneceremos uma versão reduzida de cada conjunto de dados para esta subtarefa. Esta tarefa é especialmente relevante para testar modelos com uma janela de tamanho reduzido.

O SemEval é reconhecido tanto por seu impacto acadêmico quanto por sua capacidade de fomentar a colaboração global, reunindo pesquisadores de diversas instituições. Suas competições frequentemente resultam em publicações científicas de destaque e no surgimento de novas técnicas em PLN. Participar do SemEval é uma oportunidade de se conectar com a comunidade internacional e contribuir para o progresso da área.

O DataBench é o primeiro benchmark composto por conjuntos de dados tabulares do mundo real, provenientes de diferentes domínios e com um grande número de linhas e colunas, bem como uma ampla variedade de tipos de dados que permitem avaliar distintos tipos de questões relacionadas a cada tipo de dado.

Propomos uma tarefa para incentivar os participantes a desenvolverem um sistema que responda a perguntas do tipo presente no DataBench em conjuntos de dados do dia a dia, onde a resposta pode ser um número, um valor categórico, um valor booleano ou listas de vários tipos. O DataBench pode ser usado como conjunto de treinamento e validação, enquanto liberaremos outro conjunto de teste explicitamente compilado para a competição da tarefa.

O sistema desenvolvido pelos participantes receberá uma série de pares (conjunto de dados, pergunta) e precisará fornecer uma resposta, que será então comparada com um padrão de referência (gold standard).

A resposta pode ser alcançada por meio de uma variedade de métodos. Em nosso artigo, ilustramos duas abordagens diferentes usando InContext Learning (Aprendizagem em Contexto).

3. Engenharia de Prompt

A metodologia via Engenharia de Prompt adotou a abordagem TabletoContext, que visa transformar dados tabulares em entradas textuais contextualizadas, facilitando sua integração com Modelos de Linguagem de Grande Escala nas tarefas QA Tabular. A abordagem utiliza descrições de colunas e valores das tabelas para criar narrativas coerentes que enriquecem semanticamente os dados tabulares.

A metodologia consistiu nas seguintes etapas principais:

- **Coleta e Análise de Dados**
  Análise da estrutura da tabela e semântica das colunas
  Identificação de relações entre colunas
  Compreensão dos tipos de dados e intervalos de valores

- **Geração de Templates**
  Manual Humana: Analistas desenvolvem templates baseados em conhecimento do domínio
  Geração Automatizada por LLM: Usando técnicas de aprendizado one-shot com modelos tipo GPT
  Abordagem Híbrida: Combinando expertise humana com geração automatizada

- **Enriquecimento de Contexto**
  Transformação de dados brutos em declarações em linguagem natural
  Manutenção de relações semânticas entre campos
  Adição de conectivos e construções linguísticas apropriadas

3.1 Engenharia de Prompt

Neste trabalho, nos baseamos na *Engenharia de Prompt* para propor uma metodologia alternativa que utiliza o poder do *Engenharia de Prompt* para transformar dados tabulares em entradas textuais contextualizadas. Denominamos por **TabletoContext,** essa abordagem gera automaticamente templates narrativos a partir da descrição das colunas e do conteúdo das tabelas. Esses templates enriquecem o contexto semântico das tabelas, utilizando conectivos e construções linguísticas que tornam os dados mais compreensíveis para LLMs, permitindo que eles processem informações tabulares como se fossem texto narrativo.

Como contribuição principal, este artigo apresenta:

- Uma metodologia para converter tabelas em texto narrativo de forma automatizada e contextualizada, utilizando descrições semânticas das colunas e linhas.
- Análise detalhada dos ganhos e limitações dessa abordagen, explorando como a contextualização textual pode superar formatos estruturados em cenários específicos.

A metodologia do TabletoContext visa transformar dados tabulares em entradas textuais contextualizadas, facilitando sua integração com Modelos de Linguagem em Larga Escala (LLMs) em tarefas como perguntas e respostas (Question Answering QA). A abordagem utiliza descrições das colunas e dos valores das tabelas para criar narrativas coerentes que enriquecem semanticamente os dados tabulares.

Tabelas apresentam informações de forma condensada, com relações semânticas implícitas entre as colunas (atributos) e as linhas (instâncias). No entanto, a maioria dos LLMs é treinada para processar texto narrativo, não compreendendo facilmente o formato tabular. O TabletoContext resolve esse problema criando uma representação textual rica das tabelas, permitindo que os LLMs interpretem os dados tabulares de maneira mais eficaz.

3.1. Etapas da Metodologia

A entrada deste modelo é a descrição da tabela e dos dados de suas colunas. No exemplo a seguir podemos ver a descrição uma das tabelas que foram utilizadas nesse trabalho, a Tabela Titanic. A tabela Titanic contém informações sobre os passageiros que estavam a bordo do RMS Titanic durante o famoso naufrágio em 1912. Cada linha da tabela representa um passageiro, enquanto as colunas fornecem detalhes específicos sobre ele. Abaixo estão descrições detalhadas de cada coluna:

Survived: É um dado binário que indica se o passageiro sobreviveu ao naufrágio (0: Não sobreviveu,1: Sobreviveu).

Pclass:É um dado categórico que indica a classe da passagem adquirida pelo passageiro, representando uma medida de status socioeconômico (1: Primeira classe (mais alta), 2: Segunda classe, 3: Terceira classe (mais baixa)).

Name: É um dado do tipo texto que descreve o nome completo do passageiro.

Sex: É um dado categórico que indica o gênero do passageiro (male: Masculino, female: Feminino).

Age:É um dado numérico que indica a idade do passageiro em anos (pode haver valores ausentes).

Siblings_Spouses Aboard: É um dado do tipo Numérico, descreve o Número de irmãos ou cônjuges que o passageiro tinha a bordo, os possíveis valores são Números inteiros.

Parents_Children Aboard: É um dado numérico que indica o número de pais ou filhos que o passageiro tinha a bordo. Os possíveis valores são Números inteiros.

Fare: É um dado do tipo Numérico, descreve o o valor da tarifa, os possíveis valores são valores numéricos representando a tarifa em libras esterlinas

3.1.1 Coleta e Análise das Tabelas

Esta etapa tem como objetivo principal compreender a estrutura e o significado das tabelas que serão usadas na metodologia TabletoContext. Aqui, é realizada uma análise aprofundada das colunas, seus tipos de dados, e o contexto semântico que elas representam. Essa etapa é fundamental para garantir que as transformações aplicadas à tabela resultem em um texto contextualizado, claro e adequado para o modelo de linguagem.

A entrada para o modelo é uma tabela com cabeçalhos (colunas) e registros (linhas). Exemplo: dados do Titanic com colunas como `Survived`, `Pclass`, `Name`, etc. É importante que se faça a análise semântica, onde é possível Identificar o significado de cada coluna com base em descrições fornecidas ou inferidas pelo contexto dos dados. Por exemplo:

`Survived`: Indica se o passageiro sobreviveu ao acidente (0 = Não, 1 = Sim).

`Pclass`: Classe social da passagem adquirida.

3.1.2 Geração de Templates Narrativos

Na metodologia **TabletoContext**, a criação dos templates narrativos desempenha um papel crucial ao transformar dados tabulares em entradas textuais contextualizadas e compreensíveis para modelos de linguagem. Essa etapa pode ser realizada de duas formas, uma forma é a criação manual humana, e a outra forma é a criação automática por modelos de linguagem:

Criação Manual por Humanos

Um analista humano com conhecimento sobre os dados tabulares analisa as descrições das colunas e utiliza sua interpretação para elaborar templates textuais ricos e coerentes. Este processo envolve identificar o significado de cada coluna e relacioná-la com as outras, além de decidir a melhor forma de expressar os dados em texto natural, com o uso de conectores linguísticos e palavras que aumentem a clareza.

Para a tabela Titanic, o seguinte tamplate foi desenvolvido por humano:

Na lista dos passageiros que estavam a bordo do Titanic no dia do famoso acidente estão:

*Obs.:(Daqui em diante o template se repete para cada passageiro

[Name] que estava na classe [Pclass] pagou [Fare] pela passagem, [She/He] tinha [Age] anos e [survived/did not survive] e tinha [Siblings_Spouses Aboard] irmãos ou cônjuge no total, essa pessoa também tinha [Parents_Children Aboard] pais ou filhos no total.

Criação Automática por Grandes Modelos de Linguagem

Os templates também podem ser gerados automaticamente utilizando um modelo de linguagem, como o GPT, configurado com técnicas de aprendizado **One Shot**. Nesse caso, o modelo recebe um único exemplo de como os dados tabulares podem ser descritos textualmente e aprende a generalizar esse formato para outras tabelas.

Processo consiste em entrar com o input do exemplo, onde o modelo recebe uma descrição detalhada de uma tabela (colunas e valores) e um exemplo de

template. A generalização ocorre quando o modelo aprende a extrapolar o formato do exemplo para criar templates para tabelas com diferentes estruturas e descrições.

One-shot learning:
Descrição da Tarefa:
Use a descrição da tabela dada, e das suas colunas, para criar um Template do método TabletoContex.

Exemplo:
Descrição da tabela: A tabela Titanic contém informações sobre os passageiros que estavam a bordo do RMS Titanic durante o famoso naufrágio em 1912. Cada linha da tabela representa um passageiro, enquanto as colunas fornecem detalhes específicos sobre ele. Abaixo estão descrições detalhadas de cada coluna.
Descrição das colunas: Survived: É um dado binário que indica se o passageiro sobreviveu ao naufrágio (0: Não sobreviveu,1: Sobreviveu). Pclass:É um dado categórico que indica a classe da passagem adquirida pelo passageiro, representando uma medida de status socioeconômico (1: Primeira classe (mais alta), 2: Segunda classe, 3: Terceira classe (mais baixa)). Name: É um dado do tipo texto que descreve o nome completo do passageiro. Sex: É um dado categórico que indica o gênero do passageiro (male: Masculino, female: Feminino). Age:É um dado numérico que indica a idade do passageiro em anos (pode haver valores ausentes). Siblings_Spouses Aboard: É um dado do tipo Numérico, descreve o Número de irmãos ou cônjuges que o passageiro tinha a bordo, os possíveis valores são Números inteiros. Parents_Children Aboard: É um dado numérico que indica o número de pais ou filhos que o passageiro tinha a bordo. Os possíveis valores são Números inteiros. Fare: É um dado do tipo Numérico, descreve o o valor da tarifa, os possíveis valores são valores numéricos representando a tarifa em libras esterlinas Template gerado: Na lista dos passageiro que estava a bordo Titanic no dia do famoso acidente estão: [Name], na classe [Pclass], pagou [Fare] pela passagem, tinha [Age] anos e [Survived/did not survive]. Estava acompanhado de [Siblings_Spouses Aboard] e [Parents_Children Aboard].

Prompt:
Agora gere um template a partir da seguinte descrição de tabela e colunas:
A tabela Holiday Package Sales apresenta informações sobre vendas de pacotes de férias, cada linha da tabela representa um cliente, enquanto as colunas fornecem detalhes específicos sobre ele. Abaixo estão descrições detalhadas de cada coluna. Descrição das colunas: CustomerID: Identificador único de cada cliente. ProdTaken: Indica se o cliente comprou o produto (0: Não, 1: Sim). Age: Idade do cliente. TypeofContact: Tipo de contato com o cliente (Ex.: Convite da empresa ou Iniciativa própria). CityTier: Classificação da cidade com base em desenvolvimento, população e qualidade de vida. DurationOfPitch: Duração da apresentação de vendas ao cliente. Occupation: Profissão do cliente. Gender: Gênero do cliente. NumberOfPersonVisiting: Número total de pessoas planejando viajar com o cliente. NumberOfFollowups: Número total de follow-ups realizados após a apresentação. ProductPitched: Tipo de produto apresentado ao cliente. PreferredPropertyStar:

Classificação de hotel preferida pelo cliente. MaritalStatus: Estado civil do cliente. NumberOfTrips: Média anual de viagens realizadas pelo cliente. Passport: Indica se o cliente possui passaporte (0: Não, 1: Sim). PitchSatisfactionScore: Nível de satisfação com a apresentação de vendas. OwnCar: Indica se o cliente possui carro (0: Não, 1: Sim). NumberOfChildrenVisiting: Número de crianças menores de 5 anos planejando viajar com o cliente. Designation: Cargo do cliente na organização atual. MonthlyIncome: Renda mensal bruta do cliente.

A Abordagem Automática promove a escalabilidade pois permite a criação rápida de templates para tabelas grandes ou complexas, é uma abordagem flexível pois gera templates adaptados a diferentes domínios e contextos, ainda propicia a redução de esforço humano minimizando a necessidade de intervenção manual, especialmente em cenários com múltiplas tabelas ou tarefas.

Essa abordagem híbrida, onde humanos e modelos de linguagem podem colaborar na criação dos templates, aumenta a robustez e aplicabilidade da metodologia **TabletoContext** para resolver tarefas complexas de perguntas e respostas baseadas em dados tabulares.

3.1.3 Entrada ao Código

O texto gerado é alimentado diretamente no código do Python, permitindo que ele utilize seu conhecimento linguístico para realizar a tarefa solicitada, como responder perguntas relacionadas aos dados tabulares. A seguir podemos ver a parte do código encontrado em https://github.com/prisdatascience/Modelos-de-linguagem/tree/main, onde utilizamos o template gerado pelo LLM para gerar o texto completo a partir da tabela csv, transformando assim essa tabela csv em um texto fluido e contextualizado.

3.1.4. Benefícios da Metodologia

O método TabletoContex introduzido neste trabalho, proporciona um enriquecimento semântico, pois ao transformar tabelas em texto narrativo, a metodologia fornece contexto adicional para o LLM interpretar os dados. Possui uma flexibilidade por ser compatível com qualquer LLM, sem a necessidade de arquiteturas especializadas ou modelos treinados para dados tabulares. Proporciona automatização, haja vista que a geração de templates é automatizável, baseada na descrição das colunas. Além disso ainda é possível notar que essa metodologia possui domínio independente, onde irá funcionar para tabelas de diferentes domínios, desde que as descrições das colunas sejam fornecidas. A seguir podemos ver a função que gera o template:

```python
#Funcao que gera a base de dados com base no template da base '002_Titanic'
def GerarBaseTabularTitanic(dataFrame, tipoBase):

    baseDadosTabular = "In the list of passengers aboard the Titanic
    "on the day of the famous accident are:\n"

    # Itera sobre cada linha do DataFrame
    for index, row in dataFrame.iterrows():

        #Proposta mudança prompt Priscilla

        if (row['Survived'] == "0"):
            textSurvive = "did not survive"
        else:
            textSurvive = "survived"

        if (row[3] == "male"):
            textPromon = "He"
        else:
            textPromon = "She"


        if (tipoBase == 'Full'):
            template = "{2}, in class {1}, paid {7} for the ticket,was {4} years old, and {0}.
            "{3} was accompanied by {5} sibling(s)/spouse(s) and {6} parent(s)/child(ren).\n"
            baseDadosTabular += template.format(textSurvive,row['Pclass'],row['Name'],textPromon,
              row['Age'],row['Siblings_Spouses Aboard'],row['Parents_Children Aboard'],row['Fare'])
        else:
            template = "{2}, in class {1}, paid {6} for the ticket, was {4} years old, and {0}.
            "{3} was accompanied by {5} sibling(s)/spouse(s).\n"
            baseDadosTabular += template.format(textSurvive,row['Pclass'],row['Name'],
               textPromon,row['Age'],row['Siblings_Spouses Aboard'],row['Fare'])

        break


    return baseDadosTabular
```

3.1.4. Benefícios da Metodologia

O método TabletoContex introduzido neste trabalho, proporciona um enriquecimento semântico, pois ao transformar tabelas em texto narrativo, a metodologia fornece contexto adicional para o LLM interpretar os dados. Possui uma flexibilidade por ser compatível com qualquer LLM, sem a necessidade de arquiteturas especializadas ou modelos treinados para dados tabulares. Proporciona automatização, haja vista que a geração de templates é automatizável, baseada na descrição das colunas. Além disso ainda é possível notar que essa metodologia possui domínio independente, onde irá funcionar para tabelas de diferentes domínios, desde que as descrições das colunas sejam fornecidas.

4. Resultados e Discurssões

Inconsistências nas Bases de Dados

O DataBench desempenha um papel fundamental como benchmark para avaliação de modelos de linguagem em tarefas de Question Answering (QA) sobre dados tabulares. Sua estrutura sistemática e o conjunto de tarefas diversas permitem que pesquisadores identifiquem forças e limitações dos modelos em um cenário controlado. Em particular, benchmarks como o DataBench promovem a comparação justa entre modelos, incentivam melhorias em arquitetura e treinamento e fornecem insights sobre aplicações práticas.

No entanto, há desafios a serem considerados. O presente estudo identificou inconsistências nas bases de dados utilizadas, como erros de anotação e formatações inconsistentes. Tais inconsistências impactam a confiabilidade dos resultados e destacam a importância de processos rigorosos de verificação e limpeza dos dados. A seguir podemos ver a tabela que mostra as inconsistências da base.

Base	Tipo de base	Questão	Resposta Ouro	Inconsistências da base de Dados
002_Titanic	Full	Among those who survived, which fare range was the most common: (0-50, 50-100, 100-150, 150+)?	0-50	Os intervalos apresentados na questão não são disjuntos.
002_Titanic	Full	What's the most common age range among passengers: (0-18, 18-30, 30-50, 50+)?	18-30	Os intervalos apresentados na questão não são disjuntos.

002_Titanic	Full	Could you list the lower 3 fare ranges by number of survivors: (0-50, 50-100, 100-150, 150+)?	['50-100', '150+', '100-150']	Os intervalos apresentados na questão não são disjuntos.
002_Titanic	Full	What are the top 4 age ranges('30-50', '18-30', '0-18', '50+') with the highest number of survivors?	['30-50', '18-30', '0-18', '50+']	Os intervalos apresentados na questão não são disjuntos.
009_Central	Full	On which date was the highest precipitation recorded?	1882-09-23	As datas esperadas não possuem um formato entendível pelos modelos
009_Central	Full	On which date was the lowest minimum temperature recorded?	1934-02-09	As datas esperadas não possuem um formato entendível pelos modelos
009_Central	Full	On which date was the highest maximum temperature recorded?	1936-07-09	As datas esperadas não possuem um formato entendível pelos modelos
009_Central	Full	On which date was the deepest snow depth recorded?	1947-12-27	As datas esperadas não possuem um formato entendível

				pelos modelos
009_Central	Full	What are the dates of the top 5 highest recorded precipitation events?	[1882-09-23, 2007-04-15, 1977-11-08, 1903-10-09, 2021-09-01]	As datas esperadas não possuem um formato entendível pelos modelos
009_Central	Full	What are the dates of the top 3 lowest minimum temperatures recorded?	[1934-02-09, 1917-12-30, 1943-02-15]	As datas esperadas não possuem um formato entendível pelos modelos
009_Central	Full	What are the dates of the top 4 highest maximum temperatures recorded?	[1936-07-09, 1918-08-07, 1977-07-21, 2011-07-22]	As datas esperadas não possuem um formato entendível pelos modelos
009_Central	Full	What are the dates of the top 2 deepest snow depth recorded?	[1947-12-27, 1947-12-28]	As datas esperadas não possuem um formato intendível pelos modelos
016_Holiday	Full	What is the designation of the customer with ID 200002?	Executive	O modelo confunde palavras com sentidos semelhantes como 'designation' e 'occupation'
016_Holiday	Full	What are the 3 most common occupations?	['Salaried', 'Small Business', 'Large Business']	O modelo confunde palavras com sentidos semelhantes como

				'designation' e 'occupation'
016_Holiday	Sample	What is the occupation of the customer with ID 200000?		Pergunta sem resposta ouro
016_Holiday	Sample	What is the product pitched to the customer with ID 200001?		Pergunta sem resposta ouro
016_Holiday	Sample	What is the designation of the customer with ID 200002?		Pergunta sem resposta ouro
016_Holiday	Sample	What is the marital status of the customer with ID 200003?		Pergunta sem resposta ouro

5. Metodologia de Avaliação do DataBench

A metodologia de avaliação utilizada pelo DataBench é composta por um processo sistemático para medir a capacidade dos LLMs (Large Language Models) em responder perguntas formuladas a partir de dados tabulares. Este processo considera tanto a tipologia das perguntas (booleanas, categóricas, numéricas, listas de categorias e listas de números) quanto a complexidade das consultas, que podem requerer informações de uma única coluna ou de múltiplas colunas de uma tabela.

Uma característica central dessa avaliação é o uso da **avaliação relaxada**, uma abordagem que visa lidar com a imprevisibilidade na formatação das respostas dos modelos. Em vez de exigir uma correspondência exata de caracteres, a avaliação relaxada aceita pequenas variações na formatação das respostas, desde que o significado essencial seja preservado.

5.1 A Avaliação Relaxada do DataBench

A avaliação relaxada desempenha um papel crucial ao permitir maior flexibilidade no formato das respostas, facilitando a automação do processo de validação. Por exemplo, respostas booleanas como "sim", "verdadeiro", "True" e "1" são aceitas como equivalentes, desde que transmitam o mesmo significado.

Esse método é particularmente útil na análise de respostas numéricas. Para uma resposta correta como "12", a avaliação relaxada consideraria válidas as variantes "12.0" ou "doze". Isso reduz a necessidade de intervenção manual e aumenta a eficiência da avaliação.

Contudo, essa abordagem apresenta limitações significativas. A flexibilidade na definição de pequenas variações abre margem para subjetividades, potencialmente resultando em inconsistências. Além disso, a avaliação relaxada pode inflar os resultados, sugerindo uma precisão dos modelos maior do que realmente é. Isso ocorre porque respostas tecnicamente incorretas, mas semanticamente semelhantes, podem ser indevidamente aceitas.

5.1.1 Crítica à Avaliação Relaxada

Embora a avaliação relaxada seja essencial para lidar com as nuances das respostas geradas por LLMs, ela também introduz desafios. A subjetividade inerente à definição do que constitui uma "pequena variação" pode levar a discrepâncias nos resultados, pois não sabemos o quão pequena de fato essa variação está sendo, essa subjetividade e falta de definição clara nesta avaliação relaxada, no que diz respeito ao cálculo da acurácia torna esta métrica impossível de ser replicada, especialmente no caso onde temos diferentes categorias de perguntas. Além disso, essa flexibilidade pode mascarar deficiências reais dos modelos, dificultando a identificação de áreas específicas que requerem melhorias.

Por outro lado, sem a avaliação relaxada, o processo de validação seria altamente dependente de formatações rígidas, o que penalizaria os modelos por questões não relacionadas à precisão semântica. Assim, a aplicação desse método no DataBench equilibra eficiência e representatividade, mas exige cautela na interpretação dos resultados e no uso desses dados para orientar desenvolvimentos futuros. Precisaríamos ter uma descrição completa de como foi feito esse "relaxamento" para que fosse possível replicar no presente trabalho.

O artigo reconhece essas limitações e sugere que pesquisas futuras devem focar no desenvolvimento de métricas mais robustas, que combinem a flexibilidade da avaliação relaxada com maior rigor na identificação de respostas verdadeiramente corretas. Dessa forma, o DataBench continuará a ser uma ferramenta valiosa para a avaliação de LLMs em QA sobre dados tabulares, enquanto incentiva avanços na área.

Todavia, por todas as impossibilidades de replicar a acurácia, mediante às especificidades da “avaliação relaxada” que não foi apresentada de forma sistemática e detalhada, mostrando uma subjetividade da métrica adotada, no presente trabalho escolhemos quatro métricas.

5.2 Resultados a partir do método Tableto Context

Os resultados do método **TabletoContext** devem ser analisados considerando as inconsistências presentes nas bases de dados. Essas inconsistências impactam diretamente o desempenho do modelo, pois dificultam a geração de respostas precisas e a avaliação adequada. Través da tabela abaixo podemos explorar como cada tipo de inconsistência influencia os resultados.

Base de dados	Acurácia (Avaliação Flex)	rougeScore	stringPresence	BertScoreP	BertScoreR	BertScoreF1
Titanic Full	0,41	0,24	0,20	0,52	0,63	0,56
Titanic Sample	0,49	0,41	0,15	0,60	0,69	0,63
Central Park Full	0,11	0,13	0,00	0,45	0,50	0,47
Central Park Sample	0,40	0,18	0,20	0,40	0,61	0,47
Holliday Full	0,23	0,11	0,15	0,42	0,58	0,48
Holiday Sample	0,26	0,14	0,05	0,49	0,62	0,54
DisneyFull	0,10	0,10	0,00	0,46	0,54	0,49
Disney Sample	0,36	0,14	0,15	0,49	0,60	0,53

Os resultados do método **TabletoContext** refletem os desafios impostos pelas inconsistências nas bases de dados utilizadas. Essas inconsistências dificultam a geração de respostas precisas e comprometem a avaliação das respostas geradas pelo modelo. Na base **Titanic Full**, por exemplo, as perguntas apresentam intervalos de valores sobrepostos, como "0-50, 50-100, 100-150, 150+", o que gera ambiguidades e confunde o modelo. Isso impacta negativamente métricas como acurácia e stringPresence, já que o modelo tem dificuldade em identificar respostas corretas e localizar palavras-chave esperadas. Métricas como BertScore, que avaliam a semântica, conseguem capturar parcialmente a relação entre as respostas geradas e as esperadas, mas ainda assim são prejudicadas pela falta de clareza nos intervalos.

Na base **Central Park Full**, o principal problema está relacionado ao formato das datas esperadas nas respostas, que não são interpretáveis pelo modelo. Perguntas como "On which date was the highest precipitation recorded?" exigem respostas em formatos específicos, como "1882-09-23", que podem ser difíceis de processar. Isso afeta severamente métricas como rougeScore e stringPresence, que dependem da correspondência textual, enquanto a acurácia também é baixa devido à dificuldade do modelo em coincidir as respostas geradas com as esperadas. Apesar disso, o BertScore apresenta resultados ligeiramente melhores, pois avalia aspectos semânticos que não dependem exclusivamente da exatidão textual.

Na base **Holiday Full**, a confusão entre termos como "designation" (cargo) e "occupation" (profissão) prejudica o desempenho do modelo. Essa dificuldade semântica é evidente em perguntas como "What are the 3 most common occupations?", onde o modelo frequentemente gera respostas que não coincidem com as esperadas. Na base **Holiday Sample**, a ausência de respostas ouro em várias perguntas agrava o problema, tornando a avaliação das métricas inconsistente e menos confiável. Isso é refletido em valores baixos de stringPresence e rougeScore, enquanto o BertScore e a acurácia apresentam um desempenho um pouco melhor devido à menor complexidade das perguntas.

Ao comparar os resultados entre as bases completas e as amostras, observa-se que as bases completas apresentam inconsistências mais frequentes e severas, o que explica os valores consistentemente baixos das métricas. Nas amostras, as inconsistências são menos prevalentes, permitindo um desempenho ligeiramente melhor. No entanto, mesmo nas amostras, problemas como a ausência de respostas ouro e ambiguidades semânticas ainda comprometem os resultados.

Essas dificuldades destacam a necessidade de melhorias no pré-processamento dos dados e no treinamento do modelo. Ajustes nos intervalos de valores para torná-los disjuntos, a padronização do formato de datas e a inclusão de exemplos que explorem termos sinônimos podem ajudar a reduzir ambiguidades e melhorar o desempenho. Além disso, a revisão das respostas ouro, garantindo que todas as perguntas tenham respostas bem definidas, é essencial para uma avaliação mais confiável.

5.3 Limitações do Método

A eficácia do template depende de descrições precisas das colunas, além disso essas descrições devem estar associadas às terminologias usadas nas perguntas para se referir aos intens da tabela.

Tabelas grandes podem gerar textos longos, desafiando os limites de entrada do modelo. A janela de contexto do Ollama é de 2048 tokens, é muito provável que algum template gerado para algumas bases de dados denham ultrapassado esse limite.

6 REFERÊNCIA:

[1] GRIJALBA, Jorge Osés et al. Question Answering over Tabular Data with DataBench: A Large-Scale Empirical Evaluation of LLMs. In: **Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)**. 2024. p. 13471-13488.