UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Análise Comparativa de Modelos Derivados do BERT e LLMs para Extração de Dados em Documentos de Imposto de Renda

Jean Lourenço da Silva

Monografia - MBA em Inteligência Artificial e Big Data

ICMC USP SÃO CARLOS

Jean Lourenço da Silva

Análise Comparativa de Modelos Derivados do BERT e LLMs para Extração de Dados em Documentos de Imposto de Renda

Monografia apresentada ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientador: Prof. Dr. Ricardo Cerri

Versão original

São Carlos

2024

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi, ICMC/USP, com os dados fornecidos pelo(a) autor(a)

S856m

Silva, Jean Lourenço
Análise Comparativa de Modelos Derivados do BERT e LLMs para Extração de Dados em Documentos de Imposto de Renda / Jean Lourenço da Silva ; orientador Ricardo Cerri. – São Carlos, 2024.
50 p. : il. (algumas color.) ; 30 cm.

Monografia (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2024.

1. LaTeX. 2. abnTeX. 3. Classe USPSC. 4. Editoração de texto. 5. Normalização da documentação. 6. Tese. 7. Dissertação. 8. Documentos (elaboração). 9. Documentos eletrônicos. I. Cerri, Ricardo, orient. II. Título.

Jean Lourenço da Silva

Análise Comparativa de Modelos Derivados do BERT e LLMs para Extração de Dados em Documentos de Imposto de Renda

Monograph presented to the Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, as part of the requirements for obtaining the title of Specialist in Artificial Intelligence and Big Data.

Concentration area: Artificial Intelligence

Advisor: Prof. Dr. Ricardo Cerri

Original version

São Carlos

2024

RESUMO

SILVA, J. L. **Análise Comparativa de Modelos Derivados do BERT e LLMs para Extração de Dados em Documentos de Imposto de Renda**. 2024. 50 p. Monografia (MBA em Inteligência Artificial e Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2024.

Com a crescente demanda por extração e cruzamento de informações em documentos financeiros, as empresas enfrentam desafios significativos devido à complexidade dessas tarefas. Modelos avançados de PLN, como deBERTa, RoBERTa, DistillBERT, e LLMs como Llama2, GPT-4, e Google Gemini, têm sido utilizados para automatizar essas tarefas, melhorando a precisão e a eficiência, além de reduzir o tempo e o esforço necessários para a análise manual. A vasta gama de modelos BERT e LLMs, com diferentes tamanhos e requisitos computacionais, torna a otimização dessas tarefas de PLN um desafio. Este trabalho se propõe a realizar uma análise comparativa dos principais modelos encoder-only, como os derivados do BERT, e compará-los com modelos decoder-only, como o GPT, na extração de dados de documentos financeiros, especificamente do imposto de renda. Para isso, foram geradas bases de dados sintéticas com 1.000, 10.000 e 100.000 amostras. Após o fine-tuning em diversos modelos e a realização de 30 testes com dados reais, foram analisadas métricas como Precisão, Revocação, F1-Score e Acurácia, além de aspectos técnicos, como complexidade de implementação e hospedagem. Nos testes realizados, os LLMs Llama2 7B, GPT-4 e Gemini atingiram um resultado perfeito de extração de dados. Entre os modelos derivados do BERT, as versões multilíngue ou em português brasileiro apresentaram uma acurácia de 98% e F1-Score de 96,5% para o deBERTa Large, 98% e 96,5% para o deBERTa Base Multilíngue, 98% e 96,2% para o RoBERTa Base, 70,5% e 59,4% para o DistillBERT Base, 88,4% e 86,1% para o BERT Base Multilíngue, e 97% e 93,3% para o BERT Large. Concluindo assim, que o maior impacto no desempenho dos modelos se dá através da quantidade de amostras de treino. Com aproximadamente 10.000 amostras, os modelos BERT tornam-se competitivos em nível comercial ao comparado com as LLMs, oferecendo, contudo, com a vantagem em termos de custo de implementação e hospedagem.

Palavras-chave: Extração de Dados, Processamento de Linguagem Natural, BERTimbau, DeBERTa, DeBERTina, RoBERTa, DistilBERT, Imposto de Renda, Dados sintéticos

ABSTRACT

SILVA, J. L. **Análise Comparativa de Modelos Derivados do BERT e LLMs para Extração de Dados em Documentos de Imposto de Renda**. 2024. 50 p. Monograph (MBA in Artificial Intelligence and Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2024.

With the increasing demand for information extraction and cross-referencing in financial documents, companies face significant challenges due to the complexity of these tasks. Advanced NLP models, such as deBERTa, RoBERTa, DistillBERT, and LLMs like Llama2, GPT-4, and Google Gemini, have been used to automate these tasks, improving accuracy and efficiency while reducing the time and effort required for manual analysis. The wide range of BERT and LLM models, with different sizes and computational requirements, makes optimizing these NLP tasks a challenge. This study aims to perform a comparative analysis of the main encoder-only models, such as those derived from BERT, and compare them with decoder-only models, like GPT, in data extraction from financial documents, specifically income tax forms. For this, synthetic datasets with 1,000, 10,000, and 100,000 samples were generated. After fine-tuning various models and conducting 30 tests with real data, metrics such as Precision, Recall, F1-Score, and Accuracy were analyzed, along with technical aspects like implementation and hosting complexity. In the tests performed, LLMs Llama2 7B, GPT-4, and Gemini achieved perfect data extraction results. Among the models derived from BERT, the multilingual or Brazilian Portuguese versions showed an accuracy of 98% and an F1-Score of 96.5% for deBERTa Large, 98% and 96.5% for deBERTa Multilingual Base, 98% and 96.2% for RoBERTa Base, 70.5% and 59.4% for DistillBERT Base, 88.4% and 86.1% for BERT Multilingual Base, and 97% and 93.3% for BERT Large. Thus, concluding that the most significant impact on model performance comes from the amount of training data. With approximately 10,000 samples, BERT models become commercially competitive compared to LLMs, offering, however, an advantage in terms of implementation and hosting costs.

Keywords: Data Extraction, Natural Language Processing, BERTimbau, DeBERTa, DeBERTina, RoBERTa, DistilBERT, Income Tax, Synthetic Data

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Com o grande avançado da inteligência artificial e o aumento exponencial de dados gerados pelo planeta, surgem demandas para inovações de como trabalhar estes dados, no contexto financeiro extrair conhecimentos de documentos complexos porem de forma autônoma pode significar o sucesso da uma empresa.

Este trabalho visa abordar esses desafios comparando modelos derivados do BERT (Devlin *et al.*, 2019), como RoBERTa (Liu *et al.*, 2019), com Large Language Models (LLMs) em tarefas de extração de informações em documentos financeiros. O BERT (*Bidirectional Encoder Representations From Transformers*) tem sido o estado da arte durante anos devido à sua grande capacidade de compreensão contextual.

Por outro lado, LMMs representam um avanço significativo em PLN (Processamento de Linguagem Natural), oferecendo melhorias em termos de precisão e flexibilidade. Além disso, este estudo explorará técnicas de *fine-tuning* como também a criação de dados sintéticos de documentos do imposto de renda brasileiro para treinar e avaliar os modelos.

1.0.1 Contextualização e Motivação

As empresas se deparam no seu cotidiano repleto de documentos onde é necessário extrair e cruzar informações, a demanda por automatização destas tarefas exaustivas para pessoas tem sido crescente. A extração de informações em documentos financeiros são tarefas cruciais que enfrentam vários desafios, como a diversidade e a complexidade dos documentos. A aplicação de modelos avançados de PLN, como RoBERTa e os variados LLMs, promete aumentar a precisão e eficiência dessas tarefas, permitindo a automação de processos complexos e a redução do tempo e esforço necessários para a análise manual de documentos.

1.0.2 Desafios e Contribuições

Apesar dos avanços na área de processamento de linguagem natural, a extração de dados de documentos ainda apresenta desafios, documentos financeiros como o imposto de renda que será trabalhado neste projeto, é um documento robusto, de padrões variados, com mudanças anuais e preenchido por pessoas diversas, o que ocasiona em divergências mesmo em informações iguais. A extração destes dados de forma comercial é complexa, visto que a alocação dos recursos é de difícil escolha. Um LLM como o GPT-4 extrairia de forma satisfatória estes dados, porém o seu custo é elevado. Do outro lado um modelo BERT customizado especialmente para o seu problema, tem custo de implementação e manutenção baixos, porem um custo técnico elevado, devido à necessidade de se contratar um profissional capacitado para desenvolvê-lo, pois há muito modelos derivados do BERT

no mercado para análise, de tamanhos diferentes, o profissional precisa manipular os hyper parâmetros do modelo, assim como preparar manualmente uma base de dados para se treinar o modelo inicial.

Para abordar esses desafios, este trabalho propõe a criação de dados sintéticos simulando o imposto de renda para realizar o fine-tuning dos modelos derivados do BERT. A geração de dados sintéticos permitirá a criação de conjuntos de dados robustos e variados, necessários para testar a eficácia dos modelos. Nesse contexto, a comparação entre modelos derivados do BERT, como RoBERTa, e Large Language Models (LLMs), como o LLaMA, pode oferecer percepções valiosas. As contribuições esperadas deste trabalho incluem:

- **Comparação de Modelos**: Avaliar e comparar o desempenho de modelos em destaque derivados BERT, como RoBERTa, deBERTa, distillBERT com LLMs em destaque, como o LLaMA, nas métricas padrões como precisão, revogação, F1, acurácia como também em complexidade de treinamento e hospedagem.
- **Testes de modelos derivados do BERT**: Aplicar técnicas de fine-tuning em diversos modelos, com diferentes conjuntos de dados para testar o desempenho de diversos modelos.
- **Usabilidade em Contexto Real**: Testar e validar a usabilidade dos modelos em contextos reais, fornecendo evidências práticas de sua aplicabilidade em documentos financeiros em português.

Com essas contribuições, o trabalho visa proporcionar analises e comparações de modelos derivados do BERT como também uma metodologia de extração de informações no documento do imposto de renda brasileiro, promovendo uma excelente compreensão contextual, mantendo uma classificação precisa, além de considerar a hospedagem e implementação dos modelos em ambientes de produção.

1.0.3 Objetivos do Trabalho

Este trabalho tem como objetivos principais a criação e análise de dados sintéticos, assim como a avaliação e comparação de modelos de linguagem encoder e decoder only. Especificamente, buscamos:

- Criar dados sintéticos idênticos ao imposto de renda para treinar e avaliar os modelos.
- Avaliar e comparar o desempenho dos modelos mais promissores derivados do BERT (Encoder only) e as LLMs (Decoder only) mais utilizadas como LLaMA em tarefas de extração de dados.

- Investigar as relações entre tamanho, precisão, revogação, F1, acurácia, recursos, computacionais, complexidade para treinamento, complexidade para finetuning de modelos.
- Analisar a viabilidade de hospedagem e implementação de modelos derivados do BERT e de LLMs em ambientes de produção.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Introdução ao Processamento de Linguagem Natural

De forma simplória, como discutido por Diksha Khurana (Khurana *et al.*, 2022), processamento de linguagem natural pode ser dividido em duas partes: Compreensão de Linguagem Natural (Natural Language Understanding - NLU) e Geração de Linguagem Natural (Natural Language Generation - NLG).

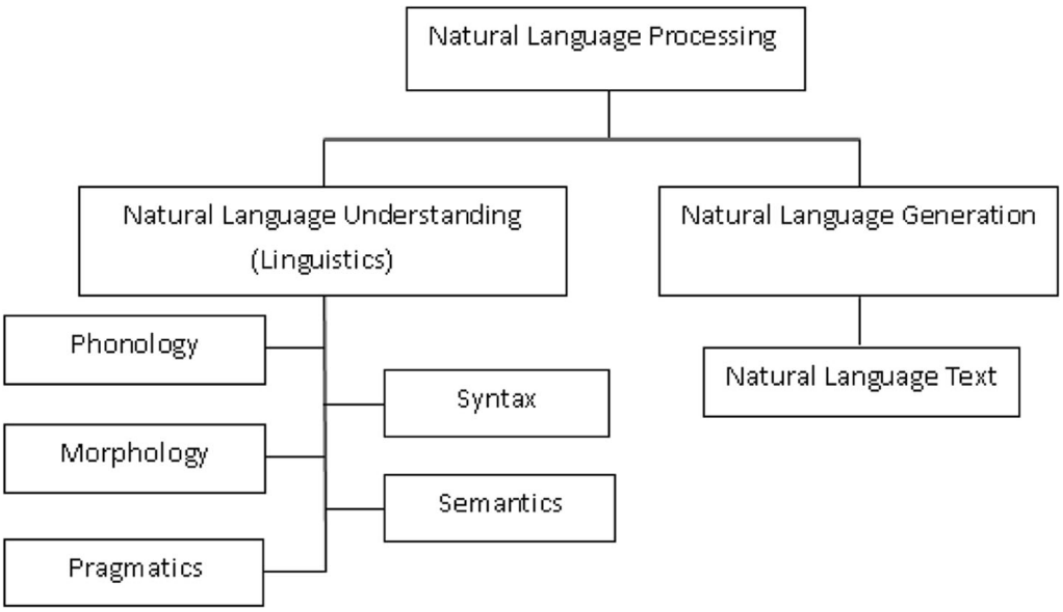

Figura 1 – Classificação do Processamento de Linguagem Natural. Imagem extraída de (Khurana *et al.*, 2022)

NLU capacita máquinas a entender a linguagem natural e analisá-la, extraindo conceitos, entidades, emoções, palavras-chave, etc. É utilizada em demandas como atendimento ao cliente para compreender os relatados e desejos, seja verbalmente ou por escrito, de pessoas. Abaixo discutiremos algumas terminologias:

- **Fonologia**: Consiste na parte da Linguística que se refere à organização sistemática dos som. A fonologia inclui o uso semântico do som para compreender o significado das línguas humanas.

- **Morfologia**: A morfologia trata das partes das palavras denominadas de morfemas, que são as unidades menores de significado.

- **Léxico**: Tanto os seres humanos quanto os sistemas de PNL interpretam o significado das palavras individuais. A parte léxica da NLU fornece a compreensão das palavras em um texto, incluindo seu significado, variações e grafia em contexto para a máquina.
- **Sintático**: O objetivo do nível sintático é produzir uma sentença que revele as dependências. Estruturais entre as palavras. Este processo revela as frases que transmitem mais significado em comparação com o significado das palavras individuais.
- **Semântico**: No nível semântico, a máquina determina os possíveis significados de uma frase processando sua estrutura, para assim reconhecer as palavras mais relevantes e entender as interações entre elas.
- **Discurso**: O nível de discurso lida com mais de uma frase, analisando a estrutura lógica e fazendo conexões entre palavras e sentenças para garantir a coerência do texto.
- **Pragmático**: Este nível analisa as sentenças que não são diretamente faladas, utilizando conhecimento do mundo real para entender sobre o que se está falando no texto. Resumidamente, é descobrir o significado pretendido da frase.

A Geração de Linguagem Natural (NLG) é o processo de produzir frases, sentenças e parágrafos que sejam significativos a partir de uma representação interna.

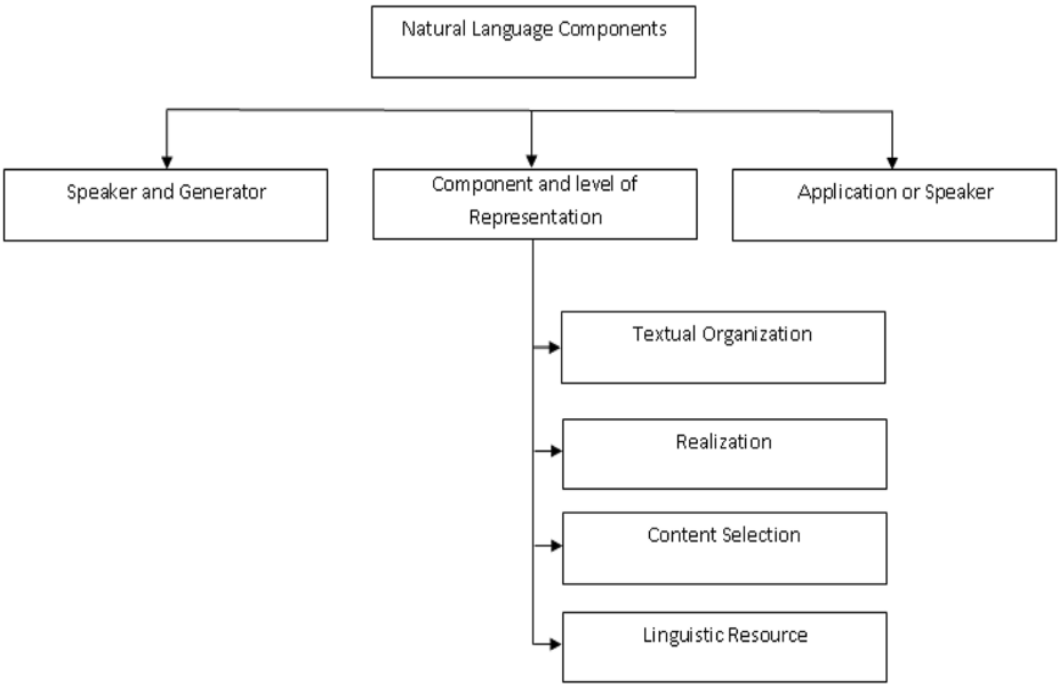

Figura 2 – Componentes de Geração de Linguagem Natural. Imagem extraída de (Khurana *et al.*, 2022)

- **Locutor**: Para gerar um texto, é necessária uma pessoa ou máquina capaz de se comunicar para se ter uma situação.

- **Componentes e Leveis de representação**: O processo de geração de linguagem envolve as seguintes tarefas entrelaçadas:

  - **Organização textual**: A informação deve ser organizada textualmente conforme a gramática, deve ser ordenada tanto sequencialmente quanto em termos de relações linguísticas como modificações.
  - **Realização**: Os recursos selecionados e organizados devem ser realizados como uma saída de texto ou voz real.
  - **Seleção de conteúdo**: A informação deve ser selecionada e incluída no conjunto. Dependendo de como essa informação é analisada, partes das unidades podem precisar ser adicionadas ou removidas.
  - **Recursos linguísticos**: Para apoiar a realização da informação, devem ser escolhidos recursos linguísticos. No final, esses recursos se resumirão a escolhas de palavras específicas, expressões idiomáticas, construções sintáticas, etc.

- **Aplicação ou o orador**: Serve apenas para manter o modelo da situação. Nesta etapa, a aplicação inicia o processo, mas não participa da geração de linguagem. Ele armazena o histórico, estrutura o conteúdo potencialmente relevante.

O processamento de linguagem natural (NLP) tem várias aplicações, como tradução automática, detecção de *spam* em e-mails, extração de informações, sumarização e resposta a perguntas; abaixo discutiremos sobre algumas das principais aplicações:

- **Tradução automática**: A tradução automática já tem implícito em seu nome para que serve, porém, ela não consiste apenas em traduzir algumas palavras, mas também como a preservação do significado das frases, gramática e tempos verbais. O Google Translate utiliza métodos estatísticos e de aprendizado de máquina, buscando por paralelos entre os idiomas e avaliam a probabilidade de correspondência.

- **Sistemas de categorização**: Consiste em sistemas automáticos de categorização de dados, normalmente envolvendo PLN. Podem ser dados de filmes, financeiros, de futebol, médicos, etc.

- **Filtro de spam**: O filtro de *spam*,como aquele utilizado em e-mails, normalmente se utiliza de machine learning para separar emails com propósito de spam dos importantes.

- **Extração de informações**: Envolve em identificar partes de interesse em dados textuais, como nomes, lugares, eventos, datas, horários e preços. Essa extração é útil para resumir informações relevantes para as necessidades do usuário.
- **Sumarização**: Consiste na habilidade de resumir textos dos mais variados, mantendo seu significado é essencial.
- **Sistemas de dialogo**: Desde chatbots empresariais até jogos eletrônicos. São sistemas de apoio que conseguem, a certo nível, manter um diálogo com o usuário.
- **Medicina**: Projetos que visam extrair e resumir informações médicas, como sinais, sintomas e dados de resposta a medicamentos, para identificar possíveis efeitos colaterais.

2.2 Modelos de Aprendizado Profundo em PLN

A estrutura básica de uma rede neural consiste em inúmeros neurônios artificiais, ou nós, que estão interligados. Cada nó na rede recebe entrada de nós anteriores, realiza cálculos e passa sua saída para nós subsequentes. Existem alguns modelos mais comumente trabalhados em Processamento de linguagem natural (Arkhangelskaya, 2021):

- **Redes Neurais Convolucionais (CNNs)**: As CNNs são aplicadas em tarefas como análise de sentimento ou identificação de entidades nomeadas. Elas são eficazes na identificação de padrões locais ou recursos em dados.
- **Redes Neurais Recorrentes (RNNs) e suas variantes (LSTM, GRU)**: Esses modelos são capazes de lidar com sequências de dados, o que é essencial para tarefas de PLN, como tradução automática ou geração de texto.
- **Modelos de Transformadores**: Modelos de transformadores, como o BERT (Devlin *et al.*, 2019), GPT (Radford *et al.*, 2018) (Brown *et al.*, 2020) e XLNet (Yang *et al.*, 2020), revolucionaram muitas tarefas de PLN devido à sua capacidade de capturar informações contextuais em grandes conjuntos de texto.

Apesar de suas vantagens, a aplicação de modelos de aprendizado profundo em PLN apresenta vários desafios, em destaque a necessidade de grandes conjuntos de dados, treinamento intensivo, exigindo alto poder computacional e complexidade de desenvolvimento.

2.3 Introdução as Large language models (LLMs)

Para compreender os *Large language models* (LLM), podemos dividir em conjuntos: modelos codificador-decodificador, apenas codificador e modelos apenas decodificadores

(Yang *et al.*, 2023). Na Figura abaixo, é apresentado o processo detalhado de evolução dos modelos de linguagem.

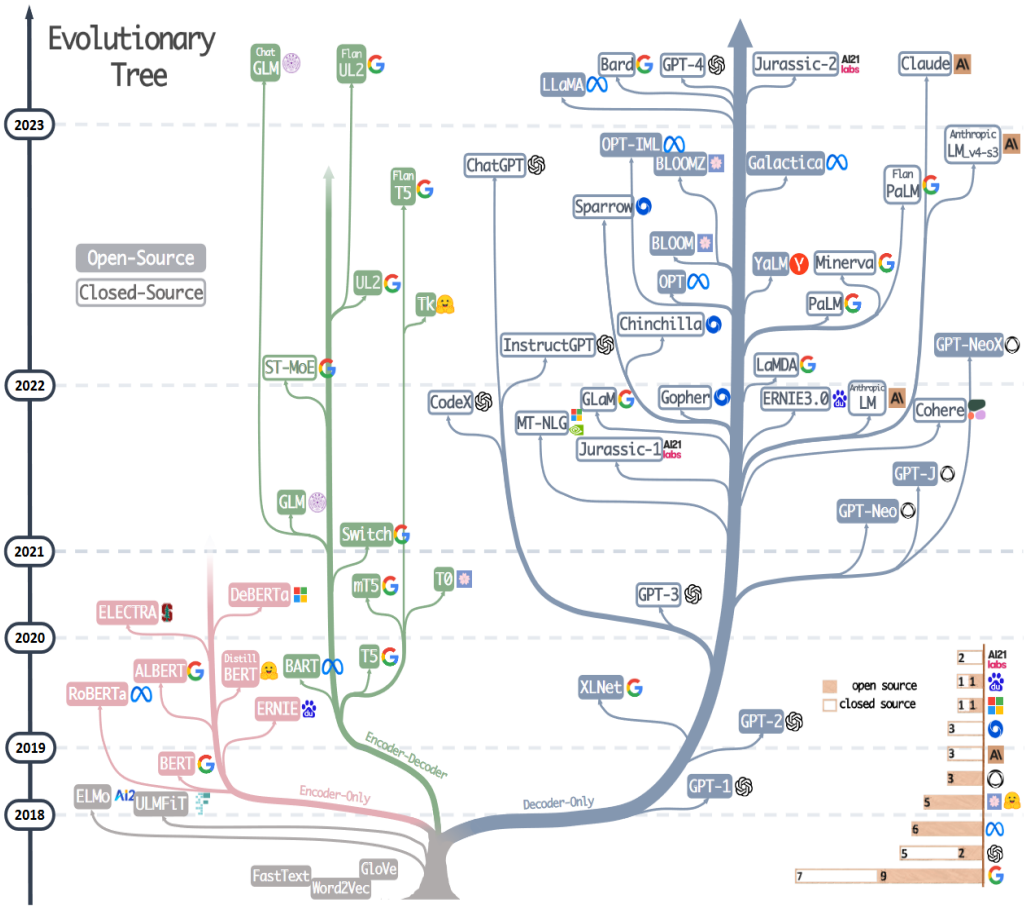

Figura 3 – Árvore evolutiva dos LLMs modernos. Imagem extraída de (Yang *et al.*, 2023)

- **Encoder-Only**: Modelos *encoder only* como o BERT apenas codificam a informação, não gerando texto novo. O modelo BERT, por exemplo, é bidirecional, o que significa que ele pode entender o contexto de uma palavra olhando para todas as palavras da sentença.
- **Decoder-Only**: Modelos *dencoder only* como GPT significa que ele é capaz de gerar texto. Ele é treinado para prever a próxima palavra em uma sequência dada as palavras anteriores.
- **Encoder-Decoder**: A arquitetura *Encoder-Decoder* é frequentemente usada em tarefas de tradução de idiomas e geração de resumos, entre outras. Ele é capaz de fazer ambas as tarefas dos anteriores.

Portanto, a principal diferença entre um *Encoder only* como o BERT e um *Decoder only* feito o GPT é que o primeiro é usado principalmente para tarefas de compreensão de

linguagem, enquanto o segundo é usado para geração de texto. A arquitetura *Encoder-Decoder*, por sua vez, é usada em tarefas de tradução e geração de texto, combinando as capacidades de codificação e decodificação.

2.3.1 Estado da arte para Encoder-only

Entre os destaques dos *encoder only* está o BERT (*Bidirectional Encoder Representations from Transformers*) (Devlin *et al.*, 2019) é um modelo de linguagem desenvolvido pelo Google que revolucionou o campo de processamento de linguagem natural. Ele é projetado para entender o contexto das palavras de forma bidirecional, ou seja, considerando o contexto tanto à esquerda quanto à direita de uma palavra. Alguns dos modelos de maior destaque Encoder only são derivados do BERT estes são: RoBERTa (Robustly Optimized BERT Pretraining Approach) (Liu *et al.*, 2019), DeBERTa (Decoding-enhanced BERT com atenção desembaraçada) (He *et al.*, 2021), ELECTRA (Efficiently Learning an Encoder that Classifies Token Re-placements Accurately) (Clark *et al.*, 2020) e DistilBERT (Distilled BERT) (Sanh *et al.*, 2020), cada um apresentando diferentes melhorias em desempenho, eficiência e arquitetura.

2.3.2 Estado da arte para Encoder-Decoder

Entre os destaques dos modelos *Encoder-Decoder* está o BART (Bidirectional and Auto-Regressive Transformers) (Lewis *et al.*, 2019), um modelo de linguagem desenvolvido pelo Facebook que combina os pontos fortes de codificadores bidirecionais, como o BERT, e decodificadores autorregressivos, como o GPT. Ele é projetado para várias tarefas de geração e compreensão de linguagem natural, usando uma arquitetura de *Transformer* que permite lidar com tarefas complexas de sequência para sequência. Alguns dos modelos de maior destaque Encoder-Decoder são: O T5 (Raffel *et al.*, 2023), GLM (General Language Model) (Du *et al.*, 2022), UL2 (Unifying Language Learning Paradigms) (Tay *et al.*, 2023), BART (Bidirectional and Auto-Regressive Transformers) (Lewis *et al.*, 2019).

2.3.3 Estado da arte para Decoder-only

Uma LLM (Large Language Model) é um modelo de inteligência artificial que visa entender e/ou gerar linguagem natural. Resumidamente, ele aprende a prever a próxima palavra em uma sequência de texto com base no contexto fornecido pelas palavras anteriores, muito útil em tarefas como tradução automática, resumo de texto, correção automática, geração de texto, etc. Alguns dos modelos de maior destaque Decodor only são: O GPT-3 (Generative pre-trained transformer 3) (Brown *et al.*, 2020), GPT-4 (Generative pre-trained transformer 4) (OpenAI *et al.*, 2024), Bard (Bayesian ARgumen-tation via Delphi) (Qin *et al.*, 2023), LLaMA (Large Language Model Meta AI) (Touvron *et al.*, 2023), Jurassic-2, Claude 3 (Anthropic, 2023).

3 METODOLOGIA

Esta seção descreve a metodologia adotada para a preparação dos dados, geração de dados sintéticos, algoritmos escolhidos, configurações de modelos e breves explicações de métricas escolhidas.

3.1 Preparação dos Dados

A escolha e obtenção são a etapa inicial e crucial de qualquer projeto de Machine Learning. A partir de 120 documentos de imposto de renda de pessoas físicas, disponíveis como base de dados, serão extraídos textos que servirão como base para a geração de dados sintéticos.

- **Coleta de Dados**: Serão extraídos os dados desejados da página "Identificação do contribuinte"do imposto de renda, em forma de texto, para a devida geração de dados sintéticos.
- **Análise e Rotulagem**: As informações serão rotuladas manualmente para criar um conjunto de dados inicial que servirá de referência para a geração de dados sintéticos, e também como texto final em dados reais pelos modelos que serão treinados apenas com dados sintéticos.
- **Ocultação de Dados Sensíveis**: Todos os dados financeiros sensíveis, como valores exatos, nomes de entidades, e identificadores pessoais, serão ocultados ou substituídos por valores genéricos para preservar a confidencialidade e garantir a conformidade com leis de proteção de dados.

3.1.1 Imposto de Renda

O Imposto de Renda brasileiro é um documento robusto que detalha as informações financeiras de pessoas físicas e jurídicas ao longo de um ano. Ele abrange dados sobre rendimentos, despesas, bens e direitos, ele é divido nas seguintes partes:

Tabela 1 – Categorias de Dados do Imposto de Renda

Categoria
Identificação do contribuinte
Dependentes
Alimentandos
Rendimentos tributáveis recebidos de pessoa jurídica pelo titular
Rendimentos tributáveis recebidos de pessoa jurídica pelos dependentes

Categoria
Rendimentos tributáveis recebidos de pessoa física e do exterior pelo titular
Rendimentos tributáveis recebidos de pessoa física e do exterior pelos dependentes
Rendimentos isentos e não tributáveis
Rendimentos sujeitos à tributação exclusiva / definitiva
Rendimentos tributáveis recebidos de pessoa jurídica pelo titular (imposto com exigibilidade suspensa)
Rendimentos tributáveis recebidos de pessoa jurídica pelos dependentes (imposto com exigibilidade suspensa)
Rendimentos tributáveis de pessoa jurídica recebidos acumuladamente pelo titular
Rendimentos tributáveis de pessoa jurídica recebidos acumuladamente pelos dependentes
Imposto pago / retido
Pagamentos efetuados
Doações efetuadas
Declaração de bens e direitos
Dívidas e ônus reais
Informações do cônjuge ou companheiro
Espólio
Doações a partidos políticos
Doações diretamente na declaração - ECA
Resumo

Para este experimento, foi optado por focar na primeira página "Identificação do contribuinte", visto que são dados pertinentes a inúmeros tipos de documentos financeiros, mas que também consistem em alguns dados únicos do imposto de renda. Todos os dados da página proposta serão considerados e devidamente extraídos.

3.2 Criação de Dados Sintéticos

Documentos financeiros usualmente são de extrema importancia, e repletos de dados sigilosos, o que ocasioana uma dificil obtenção dos mesmos, na maioria das situações a pessoa não deseja liberar seus dados para uso, o que dificulta uma quantidade substancial de amostras para treinamento. Serão criados três conjuntos de dados sintéticos simulando a primeira página do imposto de renda, criados especialmente para a tarefa de extração de dados.

- **1.000 amostras**: Representando um volume de dados menor para avaliar o desempenho inicial.

- **10.000 amostras**: Um conjunto intermediário para observar a evolução do desempenho com um volume de dados mais significativo.
- **100.000 amostras**: O maior conjunto de dados, criado para avaliar o impacto de grandes volumes de dados no desempenho dos modelos.

Será utilizado os modelos de LLM's para geração de dados sintéticos, desta forma este projeto conseguirá simular diferentes cenários e variações de dados, como também é possível definir sua estrutura, ganhando tempo, pois normalmente a estruturação dos dados seria realizado de forma manual ou seria necessário a criação de algorítimos para esta tarefa.

- **Utilização do GPT-4**: O GPT-4 será a principal forma para a geração de dados sintéticos que imitam os documentos de imposto de renda reais.
- **Utilização do Llama 2**: O Llama 2 será cogitado no caso de dados sintéticos de baixa qualidade pelo GPT-4, visto que o llama é um modelo disponibilizado pela Meta, então é possível realizar finetuning com seus próprios dados aumentando o desempenho da geração sintética.
- **Bibliotecas**: No caso de qualquer limitação do GPT-4, também será utilizado bibliotecas geradoras de dados, como a Faker, capaz de gerar dados simples de forma realista, como nome, cidade, CEP, telefone, email, etc.

Inicialmente será gerado um pequeno dataset de 100 amostras de dados, para testar a eficiência das metodologias propostas acima. Todos os dados serão conferidos manualmente para garantir a qualidade dos dados criados sinteticamente. E após conferido e devidamento validado que as ferramentas gerem dados com alta qualidade, será iniciada a criação dos três datastes propostos anteriormente, de mil, dez mil e cem mil amostras.

3.2.1 Estrutura dos dados sintéticos

O dataset sintético será gerado no formato CSV, com três colunas: ID, tokens, ner_tags. Sendo a primeira apenas o ID, a segunda todas as informações da pessoa em uma única linha dividida apenas por espaços, e a terceira com as devidas *TAGS* para extração de dados. Segue um exemplo abaixo de dados sintéticos:

Tabela 2 – Estrutura de dado sintético

id,tokens,ner_tags
0,Ryan Cavalcanti CPF: 940.375.618-75 26/04/1989 384575043220 Possui cônjuge ou companheiro(a)? Não 902.531.468-60 Houve alteração de dados cadastrais? Sim Um dos declarantes é pessoa

com doença grave ou portadora de deficiência física ou mental? Não Favela de Martins 63 Bairro/Distrito: Acaiaca Silveira MG 94081990 DDD/Telefone: +55 51 7797 6901 mendeslara@ig.com.br 11 8010-7505 Ocupação Principal: Telegrafista Tipo de declaração: Declaração de Bens 59.39.37.06.79-03,0 0 1 1 2 3 4 4 4 4 4 5 6 6 6 6 6 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 8 8 8 9 10 10 11 12 13 14 14 14 14 14 15 16 16 17 17 17 18 18 18 18 18 18 19

A escolhas das *TAGS* ocorre através das labels escolhidas, as quais são: Nome, CPF, Data de Nascimento, Título Eleitoral, Possui Conjuge, CPF Conjuge, Alteração Endereço, Deficiencia, Endereço, Número, Bairro, Município, UF, CEP, DDD/Telefone, Email, DDD/Celular, Ocupação Principal, Tipo de Declaração, Recibo.

Todas as palavras que foram da label "nome", ganharão um "ner_tag"de valor 0, da label "CPF"um "ner_tag"de valor 1, e assim por diante, sendo consideradas palavras por palavra separadas por espaço, segue um exemplo simples abaixo:

Tabela 3 – Exemplo dataset NER

id,tokens,ner_tags
1,João Oliveira CPF: 12345678901 01/01/1990,0 0 1 1 2

- João é tagueado como 0 (Nome).
- Oliveira é tagueado como 0 (Nome).
- CPF: é tagueado como 1 (CPF).
- 123.456.789-01 é tagueado como 1 (CPF).
- 01/01/1990 é tagueado como 2 (Data de Nascimento).

3.3 Detalhes dos modelos

Originalmente, todos os modelos derivados do BERT são originários do idioma inglês e não existem versões em português brasileiro oficiais. Algumas das empresas optam, no entanto, por fazer modelo multilíngue treinadas nas maiores linguais do mundo, português incluso, será utilizado modelo multilíngue por ser o mais próximo oficialmente que se tem de um modelo para português brasileiro, e modelos em destaque treinados por pesquisadores ou empresas brasileiras como o BERTimbau, todos fornecidos e hospedados no site do HuggingFace.

Tabela 4 – Modelos escolhidos

BERT
neuralmind/bert-base-portuguese-cased
neuralmind/bert-large-portuguese-cased
google-bert/bert-base-multilingual-cased
DISTIL BERT
distilbert/distilbert-base-multilingual-cased
ROBERTA
facebookAI/xlm-roberta-base
facebookAI/xlm-roberta-large
ELECTRA
Não há modelos em português brasileiro
DEBERTA
tgsc/debertina-base
tgsc/debertina-large
microsoft/mdeberta-v3-base

Especificamente para o Electra, um dos modelos em destaque derivados do BERT, não há versão multilíngue ou de pesquisadores/empresas treinados em português brasileiro, por isso será descartado.

3.3.1 Fine-Tuning dos Modelos

A etapa de fine-tuning é essencial para adaptar os modelos à tarefa de extração de informações nos documentos de imposto de renda, adicionando nos modelos pré-treinados os dados desejados. Para derivados do BERT utilizaremos as respectivas versões treinadas especificamente para português, caso não existam, será utilizada as versões de multilinguagem.

No caso das LLMs serão utilizadas sem nenhum treinamento, visto que os modelos são de outro escala, precisariam de muito poder computacional para realizar fine-tuning, e as empresas detentoras destes modelos não os liberam para esta finalidade.

Tanto o próprio modelo BERT como seus derivados utilizarão os dados sintéticos gerados, serão aplicadas técnicas de fine-tuning para melhorar a precisão do modelo em extração de dados do imposto de renda, ajustando seus hiper parâmetros conforme necessário.

3.4 Comparação de Desempenho

Para avaliar a eficácia dos modelos, serão realizadas comparações abrangentes entre os modelos derivados do BERT e as LLMs, utilizando as seguintes métricas:

- **Precision (Precisão)**: A precisão mede a proporção de verdadeiros positivos em relação ao total de positivos previstos pelo modelo.
- **Recall (Revocação ou Sensibilidade)**: O recall mede a proporção de verdadeiros positivos em relação ao total de positivos reais.
- **F1 score (Pontuação F1)**: A pontuação F1 que estou utilizando é calculada de forma micro, ou seja, ela considera todos os verdadeiros positivos, falsos positivos e falsos negativos globalmente, agregando os resultados de todas as classes antes de calcular a média harmônica entre precisão e recall.
- **Accuracy (Acurácia)**: A acurácia é a proporção do número total de previsões corretas em relação ao total de previsões feitas.
- **Recursos Computacionais**: Será analisada a complexidade de cada modelo, como o tempo de treinamento, tamanho do modelo.
- **Hospedagem e Implementação**: A viabilidade de hospedagem dos modelos em ambientes de produção será avaliada, considerando fatores como custo, escalabilidade, e facilidade de integração.

As métricas precisão, recall, f1 score serão calculadas através do modulo sklearn.metrics da biblioteca sklearn, todos os valores de parametros serão utilizados na versão *default*, os quais consistem nestes valores:

Tabela 5 – Configurações da Função `precision_recall_fscore_support`

Parâmetro	**Valor**
`beta`	1.0
`labels`	`None`
`pos_label`	1
`average`	`None`
`warn_for`	`('precision', 'recall', 'f-score')`
`sample_weight`	`None`
`zero_division`	`'warn'`
`Versão Utilizada`	`1.0.2`

Com esta metodologia, espera-se desenvolver uma abordagem robusta e eficiente para a extração e categorização de informações em documentos financeiros, promovendo uma análise automatizada precisa e útil em contextos reais.

3.4.1 Similaridade de strings

O cálculo da acurácia irá utilizar a biblioteca fuzzywuzzy versão 0.18.0 do python a qual utiliza a distância de Levenshtein para calcular a similaridade de *strings*.

A distância de Levenshtein é uma métrica que mede a diferença entre duas strings, calculando o número mínimo de operações (inserções, deleções, ou substituições) necessárias para transformar uma string na outra.

Resultados acima de 90% no cálculo serão validados como sucesso. Medida escolhida para evitar situações em que a previsão dos modelos aparece ou some com espaços, ou que a previsão adicionou ou removeu vírgulas, dois pontos, etc., situações onde seria constatado como string não equivalente se comparada de forma padrão, levando a considerar como erro, por exemplo.

3.4.2 Teste com dados reais

Após o devido finetuning dos modelos, será realizado 30 testes com dados reais de documentos do imposto de renda brasileiro, o qual irá nos revelar as métricas precisão, revocação, F1, acurácia. O texto será feito com as strings extraidas dos documentos de imposto de renda manualmente, as quais também foram utilizadas para geração de dados sintatéticos. Abaixo uma amostra anonimizada de como é essa *string*:

"Nome: ROBERT DANIEL ALVES CPF: 127.446.239-53 Data de Nascimento: 19/12/1977 Título Eleitoral: 244326778391 Possui cônjuge ou companheiro (a)? Sim CPF do cônjuge ou companheiro (a): 413.226.117-87 Houve alteração de dados cadastrais? Sim Um dos declarantes é pessoa com doença grave ou portadora de deficiência física ou mental? Não Endereço: RUA LANDEREIRO Número: 2165 Bairro/Distrito: JARDIM SÃO PAULO Município: PRATAS UF: RS CEP: 49224-440 DDD/Telefone: (14) 23457-1610 E-mail: Rob.daniel.alves@gmail.com DDD/Celular: (13) 92287-2145 Ocupação Principal: AGRÔNOMO E AFINS Tipo de declaração: Declaração de Ajuste Anual Original No do recibo da última declaração entregue do exercício de 2021: 19.33.12.53.81-77"

4 ANÁLISE DE DESEMPENHO DOS MODELOS

Esta seção descreve os experimentos realizados com diferentes modelos de linguagem para a tarefa de fine-tuning e extração de dados em documentos reais de imposto de renda, os experimentos focaram nos resultados de precisão, recall e F1-score e Acurácia.

Os 30 testes finais com dados reais para avaliar os modelos treinados, incluiram os modelos BERT, mBERT, RoBERTa, DistilBERT, deBERTa como também *Large Language Models* LLaMA 2, GPT-4, e GEMINI utilizados sem nenhuma forma de treinamente para melhorar sua eficiência.

4.0.1 Dados sintéticos gerados

A geração dos dados sintéticos foi realizada através do GPT-4. Após inserir como prompt algumas amostras da primeira página "Identificação do contribuinte"dos impostos de renda, foi pedido ao GPT-4 que gerasse um arquivo CSV simulando a estrutura escolhida para extração de dados, com os dados sintéticos gerados. Foi pedido mil dados por vez, em tempos variados, até somar o dataset final com cem mil amostras.

Também foram criados datasets dos três tamanhos com a biblioteca Faker do python. Simulando a página "Identificação do contribuinte"dos impostos de renda. Não houve qualquer diferença visível entre as duas formas de geração de datasets, ambas as amostras saíram iguais, com o mesmo nível de qualidade, não sendo viável discernir sem estudos aprofundados quem seria quem, se estivessem misturados. Ainda assim, para este projeto, foi optado por utilizar os dados gerados através do GPT-4.

Foram gerados três conjuntos de dados sintéticos simulando a primeira página do imposto de renda, criados para a tarefa de extração de dados. Os datasets foram compostos por:

- **1.000 amostras**: Representando um volume de dados menor para avaliar o desempenho inicial.
- **10.000 amostras**: Um conjunto intermediário para observar a evolução do desempenho com um volume de dados mais significativo.
- **100.000 amostras**: O maior conjunto de dados, criado para avaliar o impacto de grandes volumes de dados no desempenho dos modelos.

Os dados estavam em formato CSV com três colunas: id, tokens e ner_tags, onde cada ner_tag representa uma classe de extração de dados específica.

4.0.2 Configuração dos Experimentos

Os experimentos foram realizados com as seguintes configurações do "TrainingArguments"utilizando a biblioteca Transformers para python:

Tabela 6 – Configurações dos experimentos

Parâmetro	**Valor**
Evaluation Strategy	epoch
Save Strategy	epoch
Learning Rate	2e-6
Per Device Train Batch Size	8
Per Device Eval Batch Size	8
Num Train Epochs	5
Weight Decay	0.07
Save Total Limit	2
Load Best Model at End	True
Logging Steps	10

Com uma única ressalva, os modelos de 100 mil amostras foram treinados por 3 epochs e com o batch size reduzido para 4. Também foram modificadas configurações de dropout nas configurações do "AutoConfig"da biblioteca transformers. O Dropout foi utilizado como tecnica de regularização para evitar o overfitting.

Tabela 7 – Configurações de dropout

Parâmetro	**Valor**
hidden dropout prob	0.3
attention probs dropout prob	0.3

Com excessão dos modelos de 100 mil amostras, todos os modelos derivados do BERT foram treinados por 5 épocas, *max_length* de 512 em uma divisão 70% treino e 30% teste com 3 datastes diferentes, o primeiro contendo 1 mil amostras o segundo 10 mil amostras e o terceiro com 100 mil amostras de dados sintéticos simulando a primeira página do imposto de renda cujos dados consistem nas informações do contribuinte, propriamente criados para a tarefa de extração de dados em formato CSV contendo essas 3 colunas: id, tokens, ner_tags.

Para a consistência das análises, os testes finais em dados reais foram utilizados todos com um *score_threshold* de 0.8, que se refere ao valor mínimo de confiança que uma previsão deve ter para ser considerada válida.

4.0.3 Especificações da GPU Utilizada

A tabela a seguir apresenta as principais informações da GPU utilizada nos experimentos:

Tabela 8 – Especificações da GPU utilizada nos experimentos

Informação	**Detalhes**
Nome da GPU	NVIDIA GeForce RTX 3060
Versão do Driver	555.42.06
Versão do CUDA	12.5
Memória Total	12 GB (12288 MiB)
Consumo de Energia	15W / 170W

4.0.4 Exemplo de predição

Aqui consta uma amostra devidamente anonimizada de predição de um modelo BERT ('neuralmind/bert-large-portuguese-cased') treinado com 100 mil amostras. A extração foi perfeita, o que é necessário enfatizar é que optei por treinar o modelo para extrair também as labels por exemplo "Nome:CAROLINA"ao invés de apenas "CAROLINA", esta configuração melhorou no desempenho do modelo, principalmente em dados como titulo eleitoral onde são 12 números sem nenhum tipo de máscara.

Tabela 9 – Exemplo de predição de um modelo BERT

LABEL: TEXTO EXTRAIDO
Nome: Nome:CAROLINA NEVAS GOULART
CPF: CPF:416.112.238-59
Data de Nascimento: Data de Nascimento:13/11/1994
Título Eleitoral: Título Eleitoral:234567896125
Possui Conjuge: Possui cônjuge ou companheiro (a)?Sim
CPF Conjuge: CPF do cônjuge ou companheiro (a):123.355.432-16
Alteração Endereço: Houve alteração de dados cadastrais?Não
Deficiencia: Um dos declarantes é pessoa com doença grave ou portadora de deficiência física ou mental?Não
Endereço: Endereço:RUA OTAVIO MOSQUITO
Número: Número:352
Bairro: Bairro/Distrito:RESERVA ANA ROSA
Município: Município:DOURADOS
UF: UF:CE
CEP: CEP:12465-339
DDD/Telefone: DDD/Telefone:(13) 92456-1420
Email: E-mail:CAROL_NEVAS@GMAIL.COM.BR

DDD/Celular: DDD/Celular:(23) 12534-7610
Ocupação Principal: Ocupação Principal:BANCÁRIO,ECONOMIÁRIO,ESCRITURÁRIO, SECRETÁRIO,ASSISTENTE E AUXILIAR ADMINISTRATIVO
Tipo de Declaração: Tipo de declaração:Declaração de Ajuste Anual Original
Recibo: No do recibo da última declaração entregue do exercício de 2022:19.22.52.83.83-11

Por fim, vale ressaltar que o próprio modelo BERT e derivados já categorizam as informações nas labels/classes predefinidas e trazem os resultados devidamente separados. A forma de visualizá-los é através da sua preferência, trabalhando os resultados com Python. Exemplo de predição das duas primeiras classes, extraídas diretamente do modelo, sem edição:

[{
"entity": "Nome",
"word": "Nome:CAROLINA NEVAS GOULART",
"score": 0.98
},
{
"entity": "CPF",
"word": "CPF:416.112.238-59",
"score": 0.99
}]

4.1 Resultados de extração de dado com BERT

Nesta sessão, apresentamos os resultados do modelo BERT (‘neuralmindbert/base-portuguese-cased‘)(Souza; Nogueira; Lotufo, 2020), o modelo BERT (‘neuralmindbert/large-portuguese-cased‘), e o modelo BERT (’google-bert/bert-base-multilingual-cased’) (Devlin *et al.*, 2018). Ambos foram treinados e avaliados em datasets de diferentes tamanhos: 1.000 (mil), 10.000 (dez mil) e 100.000 (cem mil) amostras.

Tabela 10 – Resultados do modelo 'neuralmindbert/base-portuguese-case'

Métricas	**bert-base-1k**	**bert-base-10k**	**bert-base-100k**
Precisão Média (%)	15.24	92.55	91.64
Recall Média (%)	7.38	92.07	88.73
F1-Score Média (%)	8.86	92.23	89.56
Acurácia Média (%)	47.0	95.00	95.50
Tempo de Treinamento (Min)	1.81	16.84	116.93
Tamanho (MB)	414	414	414

O modelo bert-base-1k necessitou utilizar um score_threshold de 0.1, para apresentar resultados.

Podemos observar que o desempenho do modelo BERT ('neuralmindbert/base-portuguese-cased') melhora consideravelmente ao se comparar um pequeno dataset de mil amostras com um dataset de 10 mil amostras. Contudo, a diferença não se manteve ao aumentar o número de amostras para 100 mil. Isso indica que 10 mil amostras já são suficientes para se conseguir um alto desempenho, mas que aumentar o número de dados nem sempre garante melhorias proporcionais na performance.

Tabela 11 – Resultados do modelo 'neuralmindbert/large-portuguese-case'

Métrica (Médias)	**bert-large-1k**	**bert-large-10k**	**bert-large-100k**
Precisão (%)	17.02	88.05	93.64
Recall (%)	17.02	84.64	93.18
F1-Score (%)	17.02	85.78	93.33
Acurácia (%)	13.00	88.50	97.00
Tempo de Treinamento (Min)	5.63	52.10	362.4
Tamanho (MB)	1240	1240	1240

O modelo bert-large-1k necessitou utilizar um score_threshold de 0.1, para apresentar resultados.

O mesmo padrão é observado nos resultados do modelo BERT('neuralmindbert/large-portuguese-cased'), onde o desempenho aumenta significativamente até o dataset de 10 mil amostras, mas não cresce na mesma proporção com o aumento para 100 mil amostras.

Além disso, as diferenças entre as versões base e large não foram significativas, sugerindo que o tamanho do dataset tem um impacto mais substancial no desempenho do modelo do que o tamanho do modelo em si.

Em geral, o dataset de 100 mil amostras proporciona o melhor equilíbrio entre desempenho e complexidade computacional. A versão base do BERT se destacou pelo menor tempo de treinamento, tornando-se a melhor escolha para aplicações onde a precisão e a robustez do modelo são cruciais, com um período de treinamento equilibrado.

Tabela 12 – Resultados do modelo 'google-bert/bert-base-multilingual-case'

Métrica (Médias)	**mbert-base-1k**	**mbert-base-10k**	**mbert-base-100k**
Precisão (%)	19.72	80.19	87.24
Recall (%)	13.35	77.80	85.61
F1-Score (%)	14.62	78.60	86.14
Acurácia (%)	38.50	82.50	88.45
Tempo de Treinamento (Min)	1.53	13.76	82.21
Tamanho (MB)	680	680	680

O modelo mbert-base-1k necessitou utilizar um score_threshold de 0.1, para apresentar resultados.

Analisando os resultados do modelo multilíngua do BERT, o padrão se identifica novamente, onde a quantidade de amostras é o principal fator de melhora de desempenho no modelo. Apesar de alcançar uma precisão alta, este modelo multilíngua não se destaca, os modelos anteriores treinados apenas em português são superiores.

4.2 Resultados de extração de dado com DistilBERT

Nesta seção, apresentamos os resultados do modelo BERT ('distilbert-distilbert/base-multilingual-cased') (Sanh *et al.*, 2019). O modelo foi treinado e avaliado em datasets de diferentes tamanhos: 1.000 (mil), 10.000 (dez mil) e 100.000 (cem mil) amostras.

Tabela 13 – Resultados do modelo 'distilbert-distilbert/base-multilingual-cased'

Métricas	**distilbert-base-1k**	**distilbert-base-10k**	**distilbert-base-100k**
Precisão Média (%)	54.10	56.28	60.91
Recall Média (%)	54.10	52.17	58.65
F1-Score Média (%)	54.10	53.54	59.40
Acurácia Média (%)	46.50	62.50	70.50
Tempo de Treinamento (Min)	1.57	13.79	82.87
Tamanho (MB)	517	517	518

Neste modelo BERT('distilbert/distilbert-base-multilingual-cased'(, observamos que a quantidade de amostras causou um grande impacto no desempenho do modelo, enquanto não houve aumento algum no tamanho em MB do modelo final. Porém, há um impacto grande no aumento de tempo necessário para seu treinamento, conforme o aumento de amostras para treinamento.

Em geral, o modelo apresentou métricas baixas e uma acurácia de apenas 70%, necessitando de uma quantidade maior do que 100 mil amostras para atingir resultados comerciais.

4.3 Resultados de extração de dado com roBERTa

Nesta seção, apresentamos os resultados do modelo roBERTa (‘xlm-roberta-base‘) (Conneau *et al.*, 2019), e do modelo roBERTa (‘xlm-roberta-large‘). Ambos foram treinados e avaliados em datasets de diferentes tamanhos: 1.000 (mil), 10.000 (dez mil) e 100.000 (cem mil) amostras.

Tabela 14 – Resultados do modelo ‘xlm-roberta-base’

Métricas	**roberta-base-1k**	**roberta-base-10k**	**roberta-base-100k**
Precisão Média (%)	16.83	93.60	97.13
Recall Média (%)	9.05	92.64	95.83
F1-Score Média (%)	11.01	92.96	96.26
Acurácia Média (%)	27.00	94.50	98.00
Tempo de Treinamento (Min)	2.35	19.7	153.9
Tamanho (MB)	1050	1050	1050

O modelo roberta-base-1k necessitou utilizar um score_threshold de 0.1, para apresentar resultados.

Podemos observar que o desempenho do modelo roBERTa (‘xlm-roberta-base‘) melhora substancialmente ao aumentar o número de amostras de mil para dez mil, e então novamente não há melhora substancial do aumento de 10 mil para 100 mil amostras. O tamanho do modelo não variou substancialmente, porém o tempo de treinamento tem um aumento grande, o que se deve levar em consideração na hora de optar pela melhor opção.

Tabela 15 – Resultados do modelo ‘xlm-roberta-large’

Métrica (Médias)	**roberta-large-1k**	**roberta-large-10k**	**roberta-large-100k**
Precisão Média (%)	56.34	92.65	95.57
Recall Média (%)	56.34	89.21	92.58
F1-Score Média (%)	56.34	90.36	93.51
Acurácia Média (%)	75.50	96.50	97.50
Tempo de Treinamento (Min)	7.507	67.65	402.8
Tamanho (MB)	2010	2010	2010

O mesmo padrão é observado nos resultados do modelo roBERTa (‘xlm-roberta-large‘), onde o desempenho aumenta significativamente até o dataset de 10 mil amostras, mas não cresce na mesma proporção com o aumento para 100 mil amostras. Novamente, as diferenças entre versões-base e large não foram de grande impacto, sugerindo que o tamanho do dataset tem um impacto maior no desempenho do modelo do que o tamanho do modelo em si.

Em geral, para este modelo ambas as versões-base e large, assim como o modelo treinado com 10 mil e 100 mil amostras, tem um resultado muito próximo, a métrica para escolher o modelo de destaque escolhida será o tempo de treinamento, e o equilíbrio está no modelo roBERTa (‘xlm-roberta-base‘) treinado com 100 mil amostras.

4.4 Resultados de extração de dado com deBERTa

Nesta seção, apresentamos os resultados do modelo deBERTa (‘tgsc/debertina-base‘) (Scandaroli, 2023), do deBERTa (‘tgsc/debertina-large‘), e do modelo deBERTa (‘microsoft/mdeberta-v3-base-‘) (He; Gao; Chen, 2021). Ambos foram treinados e avaliados em datasets de diferentes tamanhos: 1.000 (mil), 10.000 (dez mil) e 100.000 (cem mil) amostras.

Tabela 16 – Resultados do modelo ‘tgsc/debertina-base’

Métricas	**debertina-base-1k**	**debertina-base-10k**	**debertina-base-100k**
Precisão Média (%)	13.73	87.84	95.33
Recall Média (%)	0.69	86.88	94.89
F1-Score Média (%)	1.30	87.20	95.04
Acurácia Média (%)	12.00	87.50	97.00
Tempo de Treinamento (Min)	1.9184	18.327	183.64
Tamanho (MB)	422	423	423

O modelo debertina-base-1k necessitou utilizar um score_threshold de 0.1, para apresentar resultados.

Podemos observar que o desempenho do modelo deBERTa (‘tgsc/debertina-base‘) tem uma melhora considerável em todas as etapas no aumento de amostras para treinamento. Como também vale ressaltar que a arquitetura deBERTa possui um tamanho conciso, um tempo equilibrado para treinamento, mantendo um alto desempenho.

Tabela 17 – Resultados do modelo ‘tgsc/debertina-large’

Métrica (Médias)	**debertina-large-1k**	**debertina-large-10k**	**debertina-large-100k**
Precisão (%)	18.46	85.92	94.33
Recall (%)	2.02	85.08	93.89
F1-Score (%)	3.28	85.29	94.04
Acurácia (%)	19.00	91.00	97.00
Tempo de Treinamento (Min)	4.53	41.68	291.24
Tamanho (MB)	1620	1620	1620

O modelo roberta-large-1k necessitou utilizar um score_threshold de 0.1, para apresentar resultados.

O mesmo padrão é observado nos resultados do modelo deBERT(‘debertina-large‘), as amostras influenciam bastante o desempenho do modelo. O que se destaca no entanto, seria que na versão large, quanto mais amostras, o tempo aumenta exponencialmente, dificultando muito um treinamento com maiores quantidades de dados.

Tabela 18 – Resultados do modelo ‘microsoft/mdeberta-v3-base’

Métrica (Médias)	**mdeberta-base-1k**	**mdeberta-base-10k**	**mdeberta-base-100k**
Precisão (%)	2.12	96.16	97.66
Recall (%)	0.96	95.23	96.11
F1-Score (%)	1.27	95.54	96.54
Acurácia (%)	5.00	97.50	98.00
Tempo de Treinamento (Min)	2.26	20.044	187.21
Tamanho (MB)	1050	1050	1050

O modelo mdeberta-base-1k necessitou utilizar um score_threshold de 0.1, para apresentar resultados.

Por fim, neste modelo multilinguístico do deBERTa, o padrão que se identifica igual a todos os anteriores séria a melhora do modelo através de mais amostras para treinamento, porém, diferente dos modelos deBERTa anteriores, este não demonstrou melhora significativa entre 10 mil e 100 mil amostras.

No geral, para os modelos deBERTa, o modelo multilíngua deBERTa(‘microsoft/mdeberta-v3-base‘) foi o que apresentou melhor resultado, com um tempo de treinamento relativamente curto, um tamanho bem conciso e um alto desempenho.

4.5 Resultados de extração de dados com LLMs

Nesta seção, apresentamos os resultados dos *Large Language Models*, MetaAI Llama2 versão 7B, OpenAI GPT-4 e Google Gemini antigo Google Bard. Sem nenhuma espécie de finetuning, apenas com o modelo original disponibilizado por sua respectiva empresa.

Tabela 19 – Resultados dos testes com LLMs

Métricas	**MetaAI Llama2 7b**	**OpenAI GPT-4**	**Google Gemini**
Precisão Média (%)	100.00	100.00	100.00
Recall Média (%)	100.00	100.00	100.00
F1-Score Média (%)	100.00	100.00	100.00
Acurácia Média (%)	100.00	100.00	100.00
Tamanho (GB)	13	-	-

Os três modelos resolveram os 30 testes, com perfeição. Vale destacar que o Llama 2, mesmo sendo inferior se comparado aos concorrentes, teve um desempenho perfeito. O teste utilizado é o mesmo realizado pelos modelos BERT, através de uma string inicial contendo os dados de "identificação do contribuinte"do imposto de renda. Eles deveriam categorizar as informações e devolve-las de forma correta e estruturada.

4.6 Melhores modelos

Os modelos que se destacaram nas métricas de desempenho são apresentados na tabela a seguir:

Tabela 20 – Resultados dos melhores modelos

Métricas	**bert-large-100k**	**mdeberta-base-100k**	**roberta-base-100k**
Precisão Média (%)	93.64	97.66	97.13
Recall Média (%)	93.18	96.11	95.83
F1-Score Média (%)	93.33	96.54	96.26
Acurácia Média (%)	97.00	98.00	98.00
Tempo de Treinamento (Min)	362.4	187.21	153.9
Tamanho (MB)	1240	1050	1050

Apesar das diferenças entre os modelos, todos se mostram adequados para exploração comercial. No entanto, o deBERTa (microsoft/mdeberta-v3-base) se destacou com as melhores métricas, apresentando resultados ligeiramente inferiores aos dos LLMs em termos de precisão, recall, F1-score e acurácia. A principal vantagem do deBERTa consiste em seu tamanho reduzido de apenas 1 GB, onde o LLM mais leve considerado neste projeto é de aproximadamente 13 GB. Este tamanho implica em menor custo operacional e menores requisitos computacionais, tornando o deBERTa uma escolha mais prática e econômica resultando em uma menor necessidade de memória e poder de processamento, o que influencia diretamente no tempo de treinamento e no custo de implementação e hospedagem.

A hospedagem do modelo deBERTa (microsoft/mdeberta-v3-base), com 1050 MB e tempo de treinamento de 187,21 minutos, em uma instância AWS t3.2xlarge, custaria aproximadamente USD 1,14, valores adiquiridos diretamente no site da AWS em 09/2024. Essa estimativa reflete a eficiência do deBERTa em termos de custo-benefício, especialmente quando comparado a modelos maiores, que exigem recursos mais intensivos e geram despesas consideravelmente maiores, como seria o caso de um LLM feito o Llama2 7B.

- Para uma inferência simples, o modelo DeBERTa pode ser executado em uma instância mais leve, como a t3.2xlarge (32 GiB de memória), que custa USD 0,3328 por hora.
- Para a inferência de um modelo grande como o LLaMA 2 7B, é necessário usar uma instância mais potente, como a g4dn.2xlarge, que custa USD 0,752 por hora.

Com base nas métricas e nos testes realizados, é possível afirmar que o deBERTa seria o modelo ideal para bases de dados com no mínimo 10 mil amostras, equilibrando desempenho e eficiência em extração de dados.

4.7 Trabalhos futuros

Como possíveis trabalhos futuros, seria interessante explorar páginas mais complexas do imposto de renda, como "DECLARAÇÃO DE BENS E DIREITOS", "BENS DA ATIVIDADE RURAL"e "DÍVIDAS E ÔNUS REAIS". Essas seções são geralmente extensas e compostas de dados escritos manualmente, com inúmeros padrões. A geração sintética de dados para essas áreas traria um desafio significativo e potencialmente valioso para o desenvolvimento de modelos mais robustos.

Além disso, uma análise detalhada dos recursos computacionais necessários para treino, fine-tuning, ou executar cada modelo utilizado nesta pesquisa, como também LLMs, seja localmente ou em ambientes hospedados, seria um avanço relevante.

5 CONCLUSÕES

Neste trabalho, foi realizada uma análise de diferentes modelos para a tarefa de extração de dados a partir do imposto de renda, comparando tanto modelos BERT e suas variantes, como o deBERTa, quanto modelos LLM de última geração, como o Llama 2 7B, GPT-4 e Gemini. Os resultados demonstram que os modelos LLM obtiveram um desempenho perfeito nas métricas de precisão, recall, pontuação F1 e acurácia, o que demonstra seu grande potencial em tarefas de NLP como a extração de dados trabalhada neste projeto, mesmo sem ajustes finos específicos. Entretanto, mesmo com desempenho superior, as LLM's apresentam desvantagens substanciais em termos de custo e recursos computacionais. O Llama 2 7B, por exemplo, requer maiores gastos com hospedagem e GPU para realizar suas missões.

Por outro lado, o deBERTa ('microsoft/mdeberta-v3-base') apresenta métricas ligeiramente inferiores aos LLMs, porém suas necessidades computacionais e custos operacionais são significativamente menores. Com apenas 1 GB de tamanho, o multilíngue deBERTa consegue equilibrar de forma eficaz desempenho e eficiência, sendo uma excelente escolha para implementações, principalmente quando se trabalha com bases de dados acima de 10 mil amostras.

Portanto, a escolha do modelo ideal depende diretamente do contexto e das necessidades do projeto. Para cenários que exigem o maior desempenho possível, uma alta precisão e têm suporte financeiro, os LLMs são claramente a melhor opção. No entanto, para aplicações que requerem um bom desempenho com menores recursos e custos, o multilíngue deBERTa surge como a solução mais adequada. Dessa forma, este estudo contribui para uma escolha mais informada na aplicação de modelos de linguagem em cenários comerciais e industriais, levando em conta tanto a eficiência técnica quanto a econômica.

REFERÊNCIAS

ANTHROPIC. **The Claude 3 Model Family: Opus, Sonnet, Haiku**. 2023. Disponível em: https://paperswithcode.com/paper/the-claude-3-model-family-opus-sonnet-haiku.

ARKHANGELSKAYA, S. N. E. **Deep Learning for Natural Language Processing: A Survey**. 2021. Disponível em: http://mi.mathnet.ru/znsl7049.

BROWN, T. B. *et al.* **Language Models are Few-Shot Learners**. 2020. Disponível em: https://arxiv.org/abs/2005.14165.

CLARK, K. *et al.* **ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators**. 2020. Disponível em: https://arxiv.org/abs/2003.10555.

CONNEAU, A. *et al.* Unsupervised cross-lingual representation learning at scale. **CoRR**, abs/1911.02116, 2019. Disponível em: http://arxiv.org/abs/1911.02116.

DEVLIN, J. *et al.* BERT: pre-training of deep bidirectional transformers for language understanding. **CoRR**, abs/1810.04805, 2018. Disponível em: http://arxiv.org/abs/1810.04805.

DEVLIN, J. *et al.* **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**. 2019. Disponível em: https://arxiv.org/abs/1810.04805.

DU, Z. *et al.* **GLM: General Language Model Pretraining with Autoregressive Blank Infilling**. 2022. Disponível em: https://arxiv.org/abs/2103.10360.

HE, P.; GAO, J.; CHEN, W. **DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing**. 2021.

HE, P. *et al.* **DeBERTa: Decoding-enhanced BERT with Disentangled Attention**. 2021. Disponível em: https://arxiv.org/abs/2006.03654.

KHURANA, D. *et al.* **Natural language processing: state of the art, current trends and challenges**. 2022. Disponível em: https://link.springer.com/article/10.1007/s11042-022-13428-4.

LEWIS, M. *et al.* **BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension**. 2019. Disponível em: http://arxiv.org/abs/1910.13461.

LIU, Y. *et al.* **RoBERTa: A Robustly Optimized BERT Pretraining Approach**. 2019. Disponível em: https://arxiv.org/abs/1907.11692.

OPENAI *et al.* **GPT-4 Technical Report**. 2024. Disponível em: https://arxiv.org/abs/2303.08774.

QIN, H. *et al.* How good is google bard's visual understanding? an empirical study on open challenges. **Machine Intelligence Research**, Springer Science and Business Media LLC, v. 20, n. 5, p. 605–613, ago. 2023. ISSN 2731-5398. Disponível em: https://arxiv.org/abs/2307.15016.

RADFORD, A. *et al.* **Improving Language Understanding by Generative Pre-Training**. 2018. Disponível em: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.

RAFFEL, C. *et al.* **Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer**. 2023. Disponível em: https://arxiv.org/abs/1910.10683.

SANH, V. *et al.* Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. **ArXiv**, abs/1910.01108, 2019.

SANH, V. *et al.* **DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter**. 2020. Disponível em: https://arxiv.org/abs/1910.01108.

SCANDAROLI, T. G. Debertina: A portuguese deberta-v3 model. *In*: . [*S.l.: s.n.*], 2023. Disponível em: https://huggingface.co/tgsc/debertina-base.

SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. BERTimbau: pretrained BERT models for Brazilian Portuguese. *In*: **Proceedings of the 9th Brazilian Conference on Intelligent Systems (BRACIS)**. Rio Grande do Sul, Brazil: [*S.l.: s.n.*], 2020.

TAY, Y. *et al.* **UL2: Unifying Language Learning Paradigms**. 2023. Disponível em: https://arxiv.org/abs/2205.05131.

TOUVRON, H. *et al.* **LLaMA: Open and Efficient Foundation Language Models**. 2023. Disponível em: https://arxiv.org/abs/2302.13971.

YANG, J. *et al.* **Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond**. 2023. Disponível em: https://arxiv.org/abs/2304.13712.

YANG, Z. *et al.* **XLNet: Generalized Autoregressive Pretraining for Language Understanding**. 2020. Disponível em: https://arxiv.org/abs/1906.08237.