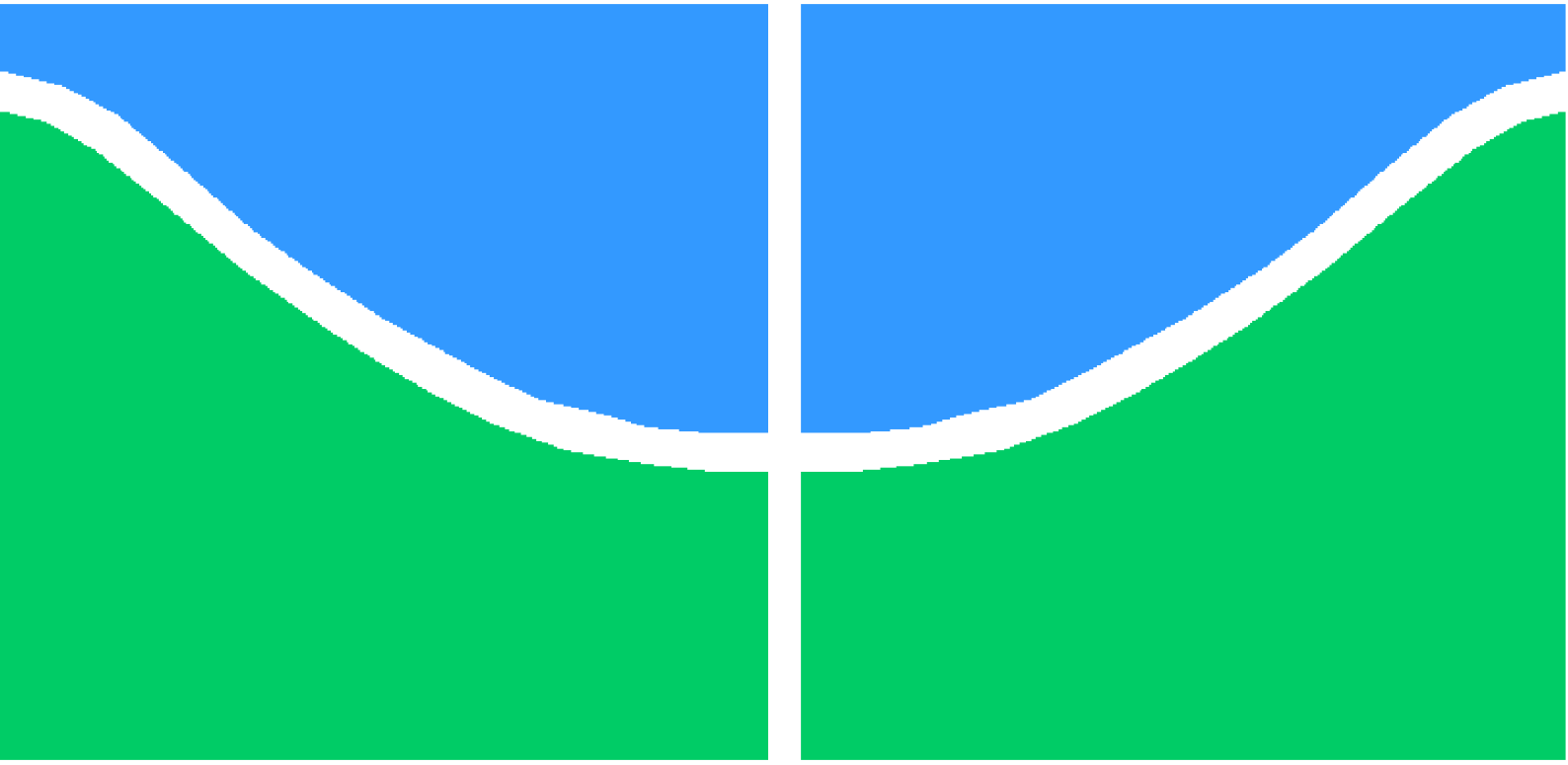


Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Engenharia de Software

Transformação Automatizada de Linguagem Informal para Formal com o Uso de Deep Learning

Autor: Roberto Martins da Nóbrega

Orientador: Prof. Dr. Fabricio Ataides Braz

Brasília, DF

2023

Roberto Martins da Nóbrega

Transformação Automatizada de Linguagem Informal para Formal com o Uso de Deep Learning

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Orientador: Prof. Dr. Fabricio Ataides Braz

Brasília, DF

2023

Roberto Martins da Nóbrega

Transformação Automatizada de Linguagem Informal para Formal com o Uso de Deep Learning

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 24 de julho de 2023:

Prof. Dr. Fabricio Ataides Braz
Orientador

Dr. Nilton Correia da Silva
Convidado 1

Dr. Henrique Marra Taira Menegaz
Convidado 2

Brasília, DF
2023

Agradecimentos

Agradeço primeiramente aos meus pais por todo amor, dedicação e apoio incondicional ao longo de minha vida e formação acadêmica. Sem vocês, eu não teria chegado até aqui. Gostaria de agradecer à minha noiva por ser minha confidente, motivadora e parceira em todos os momentos.

Por fim, agradeço aos meus amigos do grupo KRIMPES, grupo esse que fez total diferença no meu desempenho durante a graduação, um grupo de amigos que se ajudam e preocupam-se uns com os outros e no crescimento de todos os membros. E um agradecimento em especial aos membros deste grupo Matheus Gabriel e Pedro Henrique, por toda a ajuda neste trabalho e estarem sempre dispostos a me ajudar, vocês são verdadeiros amigos.

"O sucesso não é a chave da felicidade.
A felicidade é a chave do sucesso.
Se você ama o que está fazendo, terá sucesso."
(Albert Schweitzer)

Resumo

A necessidade de utilizar uma linguagem formal em textos tornou-se cada vez mais presente em diversas áreas, garantindo que as informações sejam transmitidas de maneira adequada. Isso demonstra respeito e seriedade em relação ao público-alvo do texto, proporcionando credibilidade a tudo o que é lido. Uma abordagem para auxiliar na redação correta é o uso de tecnologias de aprendizado de máquina voltadas para o processamento de linguagem natural. Essas tecnologias podem ser utilizadas para criar um modelo de inteligência artificial capaz de transcrever o texto para a forma culta da língua. No entanto, é importante destacar que recursos como esse ainda não são amplamente populares para línguas derivadas do latim, diferentemente do que ocorre em idiomas como o inglês.

Este trabalho tem como objetivo auxiliar na produção de textos na norma culta da língua portuguesa, utilizando um conjunto de dados que contém frases informais e suas equivalentes na linguagem formal. Para alcançar esse objetivo, serão aplicados métodos de *Deep Learning* para a criação de um modelo que seja capaz de realizar essa transformação. Além disso, o modelo será avaliado e possíveis melhorias serão apontadas, abrindo caminho para futuros aprimoramentos nessa área.

Palavras-chave: Inteligência Artificial; Processamento de Linguagem Natural (NLP); Aprendizado de máquina; Redes neurais; Comunicação escrita; Transformação de texto; Datasets; Papers With Code.

Abstract

The need for written texts in formal language is increasingly present in various fields to ensure that information is properly conveyed, demonstrating respect and seriousness towards the intended audience. This instills credibility in everything that is read. In order to assist in correct writing, machine learning technologies focused on natural language processing can be employed to produce an artificial intelligence model that transcribes the text into the formal form of the language. However, such resources are still not widely available when it comes to languages derived from Latin, unlike in languages like English.

With the aim of aiding the production of texts in the formal norm of the Portuguese language, this work proposes the application of Deep Learning methods to create a model that performs this transformation. This will be accomplished by utilizing a dataset containing informal phrases and their formal equivalents. The model will be evaluated, and suggestions for possible improvements will be provided, paving the way for further advancements in this field.

Key-words: Artificial Intelligence; Natural Language Processing; NLP. Machine Learning; Neural Networks; Written Communication; Text Transformation; Datasets; Papers With Code

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

UnB	Universidade de Brasília
IA	Inteligência Artificial
NLP	Natural Language Processing
PLN	Processamento de Linguagem Natural
ANN	Artificial Neural Networks

Sumário

1 Introdução

1.1 Contexto

A necessidade de produzir textos escritos em linguagem formal é cada vez mais evidente em várias áreas, como acadêmica, negócios e jurídica. Nesse sentido, o desenvolvimento de uma Inteligência Artificial capaz de reescrever textos assume grande importância, pois pode auxiliar na criação de textos mais precisos e profissionais.

Os primeiros esforços para o Processamento de Linguagem Natural (PLN), Seção 2.3 também conhecido como NLP (Natural Language Processing, em inglês), datam da década de 1950. Essas iniciativas visavam desenvolver computadores capazes de compreender e produzir linguagem humana. No entanto, foi na década de 1980 que ocorreram avanços significativos nessa área, impulsionados tanto pelo progresso do *hardware* quanto do *software*, incluindo o uso de algoritmos de Aprendizado de Máquina. Desde então, novas soluções têm sido desenvolvidas constantemente em várias áreas do PLN, tornando-se uma das áreas de crescimento mais rápido na computação.(PETSI, 2019)

1.2 Problema

Os trabalhos acadêmicos são avaliados por professores e pesquisadores que esperam encontrar textos escritos de acordo com as normas acadêmicas e científicas. Além disso, a comunicação científica desempenha um papel fundamental no avanço e na disseminação do conhecimento. Um modelo capaz de transformar textos informais em textos formais pode auxiliar pesquisadores na redação de artigos científicos e relatórios mais precisos e confiáveis, aumentando a compreensão e a aplicação dos resultados obtidos.

No contexto corporativo, os documentos representam a imagem e a reputação da empresa diante de seus *stakeholders*. Em áreas de negócios, um modelo desse tipo seria útil para garantir que documentos como contratos e relatórios estejam em conformidade com as normas e padrões esperados na linguagem escrita, proporcionando clareza e evitando ambiguidades ou interpretações equivocadas. Isso aumentaria a confiança dos clientes, sócios, contribuintes e investidores na empresa.

Quanto aos documentos legais utilizados em processos judiciais, é essencial que estejam redigidos de acordo com as normas da língua culta. Isso garante que as informações contidas sejam adequadamente compreendidas, evitando erros e mal-entendidos. Uma ferramenta que auxilie na criação desses textos seria útil para advogados e funcionários jurídicos na redação de petições ou na revisão de contratos, evitando problemas de

interpretação e aumentando a eficiência nos processos judiciais. Isso contribuiria para a confiança dos juízes e para uma resolução mais ágil dos casos.

1.3 Objetivos

1.3.1 Objetivo Geral

O objetivo deste trabalho é aplicar um modelo de aprendizado de máquina que realize a transformação de textos informais em textos formais na língua portuguesa. Além disso, busca-se realizar uma análise detalhada dos resultados obtidos com a implementação desse modelo.

1.3.1.1 Objetivos Específicos

Para alcançar o objetivo geral, serão seguidos os seguintes objetivos específicos, que fornecem uma visão detalhada do que deve ser alcançado e como isso será feito:

1. Selecionar um conjunto de dados de texto em língua portuguesa que contenha exemplos de textos informais e formais.
    - Realizar uma análise exploratória dos dados.
    - Realizar a limpeza dos dados.
    - Comparar os dados antes e depois do tratamento.
2. Aplicar um modelo de classificação de frases formais e informais.
    - Utilizar a biblioteca *Simpletransformers* para aplicar o modelo de classificação.
3. Aplicar modelo para transformação de texto informal em formal na língua portuguesa.
4. Realizar uma avaliação quantitativa e qualitativa do modelo desenvolvido, comparando seus resultados com os obtidos por outras técnicas já existentes.

1.4 Organização do Trabalho

Este trabalho de conclusão de curso está organizado nos seguintes capítulos:

- **Capítulo 1 - Introdução:** Neste capítulo, são apresentados o contexto do trabalho, o problema de pesquisa, os objetivos deste trabalho e uma síntese da metodologia planejada.

- **Capítulo 2 - Referencial Teórico:** Este capítulo descreve os conceitos teóricos que fundamentam este trabalho, incluindo informações sobre processamento de linguagem natural, aprendizado de máquina e técnicas relacionadas.
- **Capítulo 3 - Materiais e Métodos:** Neste capítulo, é apresentado o plano metodológico adotado, com detalhes sobre as etapas envolvidas, como a seleção de dados, a análise exploratória, o tratamento dos dados e a aplicação de modelos de aprendizado de máquina.
- **Capítulo 4 - Resultados:** Este capítulo apresenta os resultados obtidos no desenvolvimento deste trabalho. São descritas as etapas de busca por um conjunto de dados adequado, a análise exploratória realizada, o tratamento dos dados e a modelagem para obter um modelo capaz de transformar texto informal em formal na língua portuguesa.

Além dos capítulos mencionados, este trabalho também inclui uma conclusão, onde são apresentadas as considerações finais, as contribuições do trabalho e possíveis direções futuras de pesquisa.

2 Referencial Teórico

2.1 Considerações Iniciais

Neste capítulo, serão apresentados os conceitos teóricos relevantes para o desenvolvimento deste trabalho de conclusão de curso. Serão abordados temas relacionados aos objetivos específicos, a fim de fornecer uma base teórica sólida para a compreensão e implementação das etapas propostas.

2.2 Inteligência Artificial - IA

A Inteligência Artificial (IA) é o estudo e desenvolvimento de algoritmos e sistemas que simulam a inteligência humana, é a ciência que busca criar máquinas capazes de simular esta inteligência, permitindo a realização de tarefas como aprendizado, tomada de decisão e resolução de problemas. (RUSSELL; NORVIG, 2010)

Desde sua concepção no Século XX a Inteligência Artificial vem se desenvolvendo rapidamente, principalmente com o avanço tecnológico dos hardwares que possibilitaram um maior processamento de dados. (JIANG, 2021) A IA vem sendo aplicada em várias áreas na sociedade moderna, alguns setores que estão desfrutando muito do seu uso são a área de jogos, robótica, medicina, na área financeira entre outras. No ramo da medicina por exemplo a IA vem sendo utilizada para analisar imagens e auxiliar nos diagnósticos observando radiografias e tomografias ou utilizando Processamento de Linguagem Natural - PLN para triagem automática entre outros usos tanto na área da medicina como em outros setores.

De acordo com Russell e Norvig (RUSSELL; NORVIG, 2010), a Inteligência Artificial é composta por dois tipos principais de algoritmos: a inteligência artificial baseada em regras e a inteligência artificial baseada em aprendizado de máquina. A primeira é baseada em regras explícitas e lógicas, enquanto a segunda aprende a partir de exemplos.

A IA é dividida em vários ramos, incluindo Aprendizado de Máquina mais conhecido como *Machine Learning*, Aprendizado Profundo mais comumente chamado pelo nome em inglês *Deep Learning*, Visão Computacional, Processamento de Linguagem Natural, Robótica, Inteligência Artificial Distribuída e Sistemas Especialistas.(SOLVIMM, 2021)

O futuro da inteligência artificial é promissor e está se desenvolvendo rapidamente, mas ainda há muitos desafios a serem superados antes que possamos realmente perceber o potencial dessa tecnologia. Alguns especialistas acreditam que a inteligência artificial se

tornará a principal força motriz da economia futura e terá um impacto significativo em nossas vidas e em toda a sociedade.

2.3 Processamento de Linguagem Natural (PLN)

O Processamento de Linguagem Natural é uma área da inteligência artificial que se dedica a desenvolver algoritmos e técnicas para a compreensão e geração automática de linguagem natural, tendo como objetivo principal em permitir o "entendimento"da comunicação humana por parte das máquinas e possibilite a comunicação por linguagem natural com os humanos. O que inclui tradução automática, reconhecimento de fala, análise de sentimento além de classificação e geração de texto, estes dois últimos que serão o foco deste trabalho. Este ramo de estudo se mostra bastante desafiador pelo fato da linguagem natural estar em constante mudança e evolução, apresentando variações com o idioma, cultura, idade e gênero o que significa que é necessário considerar muitos fatores para se desenvolver um PLN e este sempre necessitará de atualizações constantes.(FACE, 2021)

Como descrito no curso (FACE, 2021) as tarefas comuns do PLN:

1. Classificação de sentenças completas: Capturar o sentimento de uma revisão, detectar se um email é spam, determinar se a sentença é gramaticalmente correta ou onde duas sentenças são logicamente relacionadas ou não;
2. Classificação de cada palavra em uma sentença: Identificar os componentes gramaticais de uma sentença (substantivo, verbo, adjetivo), ou as entidades nomeadas (pessoa, local, organização)
3. Geração de conteúdo textual: Completar um trecho com autogeração textual, preenchendo as lacunas em um texto com palavras mascaradas;
4. Extrair uma resposta de um texto: Dada uma pergunta e um contexto, extrair a resposta baseada na informação passada no contexto;
5. Gerar uma nova sentença a partir de uma entrada de texto: Traduzir um texto para outro idioma, resumi-lo.

Muitas áreas já utilizam PLN como chatbots na área de comunicação das empresas, análise de sentimento para a área de marketing, triagem automática de sintomas dos pacientes em hospitais entre outros diversos usos e áreas que as utilizam. E com o avanço da tecnologia das IAs é provável que surgirão ainda mais usos de PLN por estas e outras áreas. (GANEGEDARA; LOPATENKO, 2022)

2.3.1 Métricas para PLN

- Matriz de Confusão: Mais comumente chamada de *Confusion Matrix* em inglês é uma tabela utilizada para avaliar o desempenho de um classificador, representando o número de previsões corretas (verdadeiros positivos e verdadeiros negativos) e o número de previsões incorretas (falsos positivos e falsos negativos) como pode ser visto pela Tabela 1. (GÉRON, 2019)

	Previsão Positiva	Previsão Negativa
Verdadeiro Positivo	TP (True Positive)	FN (False Negative)
Verdadeiro Negativo	FP (False Positive)	TN (True Negative)

Tabela 1 – Matriz de Confusão

  Legenda:

  - TP - *TruePositive* representa o número de instâncias positivas previstas corretamente
  - TN - *TrueNegative* representa o número de instâncias negativas previstas corretamente
  - FP - *FalsePositive* representa o número de instâncias negativas previstas como positivas
  - FN - *FalseNegative* representa o número de instâncias positivas previstas como negativas

  Obtendo-se os dados na matriz de confusão será possível a realização das seguintes métricas:

- ***Support***: Representa o número de instâncias da classe.
- ***Precision***: Representa a proporção de verdadeiros positivos em relação a todas as previsões positivas. Serve para avaliar o quão preciso é o modelo em relação a previsões positivas. (GÉRON, 2019)

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

- ***Recall***: Representa a proporção de verdadeiros positivos em relação a todas as instâncias positivas. (GÉRON, 2019)

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

- ***F1-Score***: É a média harmônica entre *Precision* e *Recall*. É uma boa métrica quando existe equilibrio de importância entre *Precision* e *Recall*. (BIRD; KLEIN; LOPER, 2009)

$$F1_Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

- ***Accuracy***: Representa a proporção de previsões corretas em relação ao total de previsões. (BIRD; KLEIN; LOPER, 2009)

$$Accuracy = \frac{TruePositive + TrueNegative}{Total}$$

Com o treinamento do modelo de mudança de estilo de informal para formal será possível avaliá-lo utilizando as seguintes métricas:

- ***BLEU (Bilingual Evaluation Understudy)***: A métrica BLEU, introduzida por (PAPINENI et al., 2001), é uma das métricas mais populares para avaliação automática de geração de texto. BLEU avalia a qualidade de uma tradução (ou qualquer texto gerado) comparando a correspondência de n-gramas (sequências de n palavras) entre o texto gerado e o texto de referência. A pontuação BLEU varia de 0 a 1, sendo 1 uma correspondência perfeita com a referência. Esta métrica é comumente usada em tarefas de tradução automática, mas também é aplicável a outras tarefas de geração de texto.

- ***ROUGE (Recall-Oriented Understudy for Gisting Evaluation)***: A métrica ROUGE, proposta por (LIN, 2004), é uma métrica amplamente utilizada para a avaliação de resumos automáticos. ROUGE é uma família de métricas que compara a saída de um modelo com um ou mais textos de referência. As várias versões do ROUGE (como ROUGE-N, ROUGE-L, ROUGE-S) diferem na maneira como medem a correspondência entre a saída do modelo e a referência, considerando n-gramas, subsequências mais longas ou subsequências permitindo saltos. Por exemplo, o ROUGE-1 calcula o recall de unigramas entre o resumo produzido e o resumo de referência.

- ***METEOR (Metric for Evaluation of Translation with Explicit ORdering)***: A métrica METEOR, introduzida por (BANERJEE; LAVIE, 2005), também é uma métrica de avaliação automática para tradução automática. Difere do BLEU e ROUGE ao tentar alinhar palavras e frases para calcular uma pontuação, e considera uma variedade de características, como a correspondência exata de palavras, correspondência de raiz das palavras (*stemming*), sinônimos e até mesmo a ordem das palavras.

2.4 Nuvem de Palavras

A nuvem de palavras (ou *wordcloud*, em inglês) é uma representação visual de frequência de palavras em um determinado texto ou coleção de textos. Ela consiste em apresentar as palavras mais frequentes em um tamanho maior, enquanto as menos frequentes são apresentadas em tamanhos menores. As nuvens de palavras são geralmente criadas usando técnicas de processamento de linguagem natural e análise de dados. Ela oferece uma visão geral rápida e intuitiva do conteúdo de um texto ou coleção de textos, tornando fácil a interpretação e compreensão dos resultados e apresentação destas informações a não especialistas. (OESPER et al., 2011)

No contexto deste trabalho esta técnica será utilizada através da biblioteca Subseção 3.4.6, auxiliando na compreensão do conjunto de dados após o tratamento, auxiliando a averiguar as palavras com maior taxa de repetição em cada tipo de texto, exibindo seus padrões e tendências. (OESPER et al., 2011)

2.5 Stop Words

Stop words são palavras comuns que são frequentemente usadas em textos, mas não têm significado semântico importante. Essas palavras geralmente são removidas durante a pré-processamento de dados textuais para melhorar a eficiência e precisão de análises de processamento de linguagem natural, pois elas podem prejudicar ou distorcer os resultados.(MANNING; RAGHAVAN; SCHüTZE, 2008)

Algumas das palavras mais comuns que são consideradas *stop words* incluem artigos (como "a"ou "the"), preposições (como "em"ou "para"), pronomes (como "eu"ou "você"), conjunções (como "e"ou "mas") e verbos auxiliares (como "é"ou "tem"). A remoção destas palavras se torna importante devido ao fato de elas não fornecem informações significativas sobre o conteúdo do texto e, portanto, não contribuem para a compreensão do texto ou para a análise dos dados. Além disso, as *stop words* são frequentemente usadas com frequência muito alta em comparação com outras palavras, o que significa que elas podem ocupar muito espaço na representação dos dados textuais, prejudicando a visualização da nuvem de palavras o que dificultará a interpretação do conjunto de dados e muito provavelmente irá prejudicar a eficiência de processamento de linguagem natural, logo sua retirada auxilia a análise e performance do modelo utilizado.(MANNING; RAGHAVAN; SCHüTZE, 2008)

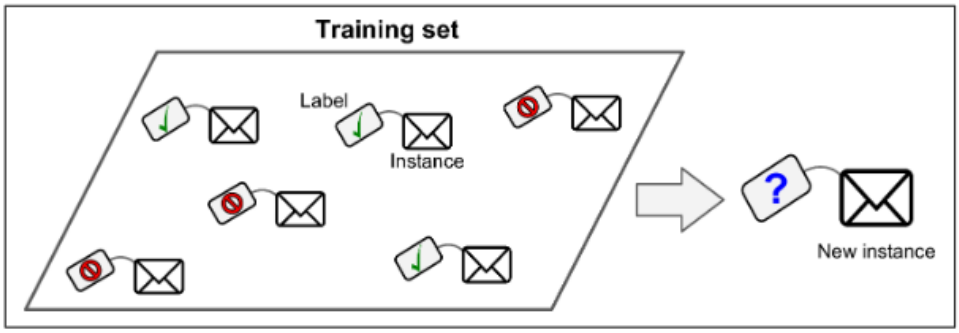
2.6 Machine Learning

"*Machine Learning* é a ciência (e arte) de programar computadores de forma que eles aprendam com os dados".(GÉRON, 2019)

É a subárea da Inteligência Artificial que busca aprender modelos a partir de dados e utilizá-los para prever e/ou tomar decisões sem seguir programação explícitas. Dividindo-se em quatro categorias, aprendizado supervisionado, não-supervisionado, semi-supervisionado e por reforço. (GÉRON, 2019)

1. **Aprendizado Supervisionado** é a técnica onde se tem uma série de dados rotulados e o objetivo é prever a classificação de novos dados se baseando na aprendizagem a partir dos já rotulados como podemos ver na Figura 1.

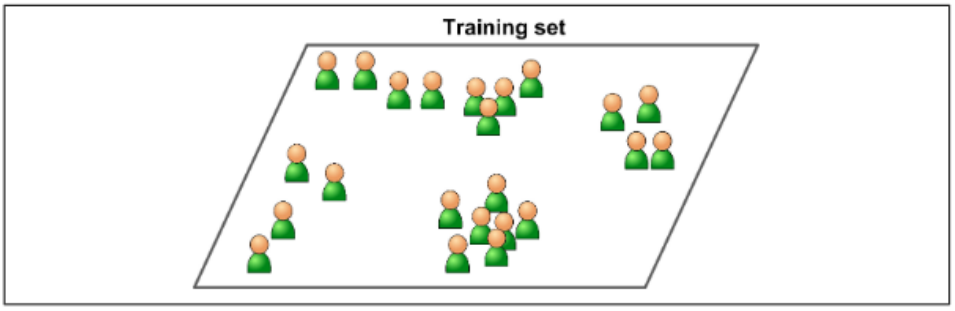
Figura 1 – Aprendizagem Supervisionada



Fonte: (GÉRON, 2019)

2. **Aprendizado Não-Supervisionado** busca identificar padrões ocultos nos dados sem o uso de rótulos, agrupando-os por dados similares, como podemos ver na Figura 2 onde o modelo tenta agrupar automaticamente sem a necessidade de interferência humana. (GÉRON, 2019)

Figura 2 – Aprendizagem Não-Supervisionada



Fonte: (GÉRON, 2019)

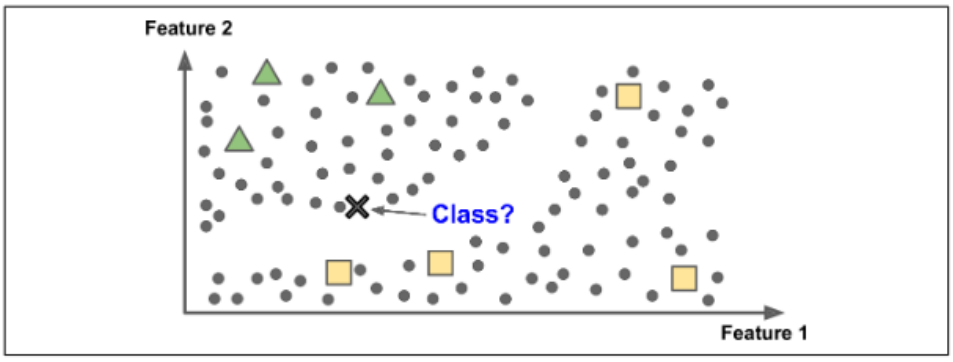
Os algoritmos não-supervisionados mais importantes segundo o Livro (GÉRON, 2019) são:

- ***Clustering***
  - K-Means

  - DBSCAN
  - Hierarchical Cluster Analysis (HCA)
- ***Anomaly detection and novelty detection***
  - *One-class SVM*
  - *Isolation Forest*
- ***Visualization and dimensionality reduction***
  - *Principal Component Analysis (PCA)*
  - *Kernel PCA*
  - *Locally-Linear Embedding (LLE)*
  - *t-distributed Stochastic Neighbor Embedding (t-SNE)*
- ***Association rule learning***
  - *Apriori*
  - *Eclat*

3. **Aprendizado Semi-Supervisionado** nesta abordagem utiliza-se dados parcialmente rotulados para melhorar o desempenho do modelo. Fazendo uso de dados rotulados para guiar o aprendizado dos dados não rotulados como pode-se ver na Figura 3. A maioria dos algoritmos semi-supervisionados são combinações de algoritmos supervisionados e não supervisionados. (GÉRON, 2019)

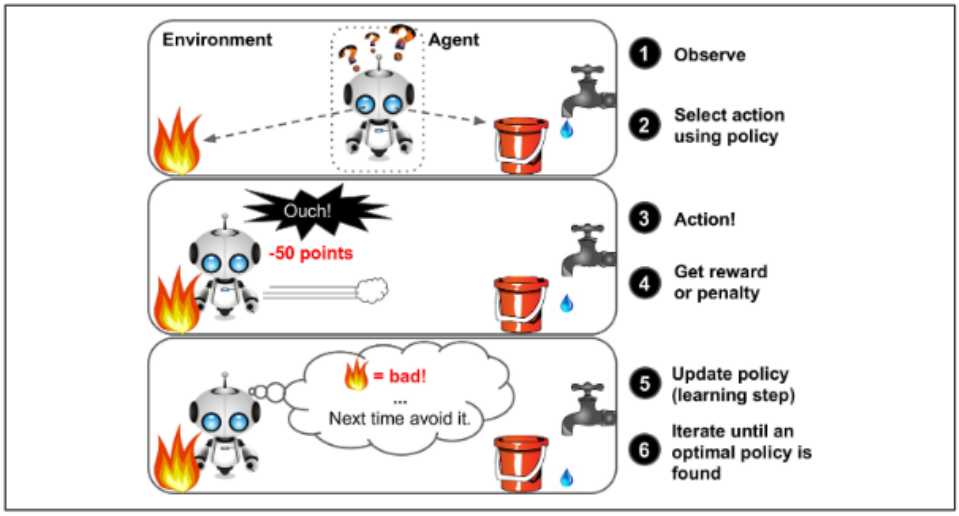
Figura 3 – Aprendizagem Semi-Supervisionada



Fonte: (GÉRON, 2019)

4. **Aprendizado por Reforço** é uma técnica onde o modelo é treinado por meio de recompensas e punições para tomar decisões em situações de incerteza. Ele deve aprender sozinho e escolher a melhor estratégia chamada de política, observando a Figura 4 observamos isso, o robô escolhe sua ação e verifica se a escolha lhe rendeu alguma recompensa ou punição e toma suas decisões futuras levando em consideração esta iteração. (GÉRON, 2019)

Figura 4 – Aprendizagem Por Reforço



Fonte: (GÉRON, 2019)

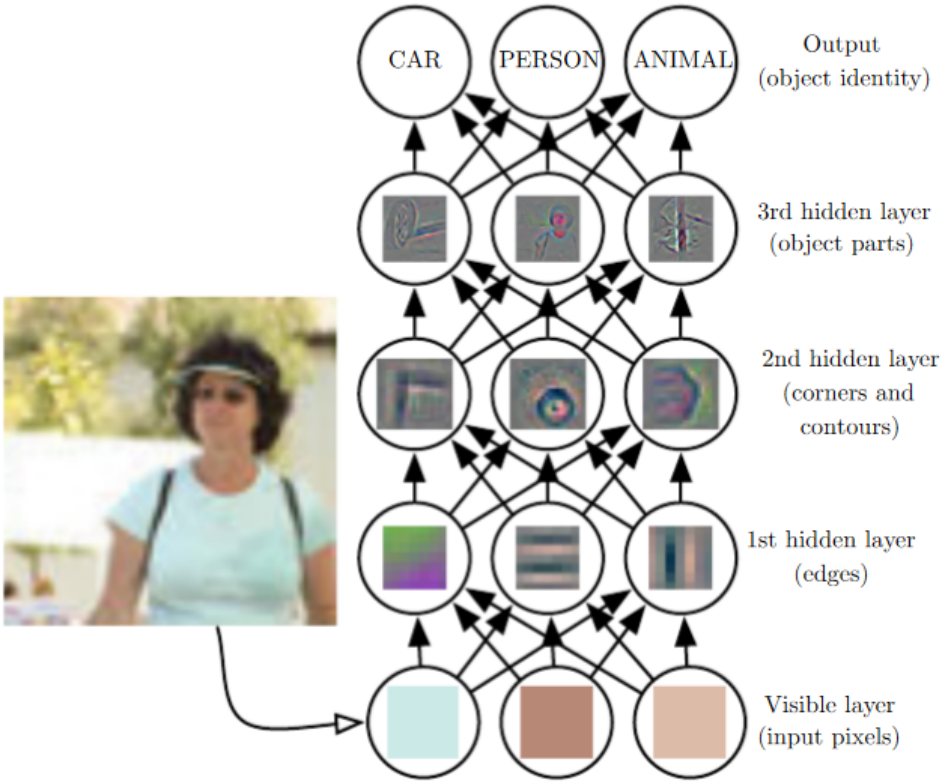
2.7 Deep Learning

A humanidade sempre busca imitar a natureza, foi assim com os pássaros para construção dos aviões, com as plantas burdock para a criação do velcro e inúmeras outras invenções inspiradas ao se observar a natureza, e foi observando o cérebro que veio a inspiração para a criação das Redes Neurais Artificiais (ANNs em inglês), mas apesar desta origem as ANNs evoluíram e se tornaram diferentes de suas versões biológicas. As Redes Neurais Artificiais são a base do *Deep Learning*, oferecendo ferramentas flexíveis e poderosas que podem lidar com tarefas complexas da Aprendizagem de Máquina como classificar imagens, reconhecer fala, recomendar vídeos e até jogar jogos. (GÉRON, 2019)

Deep Learning é uma técnica de *Machine Learning* que organiza os neurônios em camadas. Por possuir várias camadas se comparado com as outras técnicas de aprendizado de máquina dar-se o nome de "profundo". Esta característica permite que o modelo deste tipo capture padrões complexos em grandes quantidades de dados. (GÉRON, 2019)

O Aprendizado Profundo é baseado em técnicas de aprendizado supervisionado e não supervisionado, e é uma área da inteligência artificial que tem tido grande sucesso em aplicações como reconhecimento de imagens Figura 5, processamento de linguagem natural e reconhecimento de fala. Esta técnica se concentra na aprendizagem por representação hierárquica dos dados, esta capacidade de aprender por meio de características abstratas a partir de exemplos de treinamento que a difere das demais técnicas de aprendizagem de máquina. (GOODFELLOW; BENGIO; COURVILLE, 2016)

Figura 5 – Modelo de *Deep Learning*

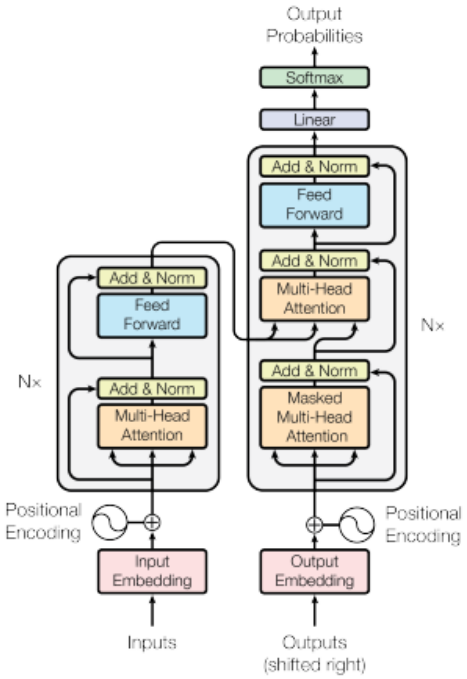


Fonte: (GOODFELLOW; BENGIO; COURVILLE, 2016)

2.8 Transformers

Transformers são arquiteturas de *Deep Learning* utilizadas principalmente em processamento de linguagem natural e visão computacional como pode ser vista na Figura 6. A principal ideia por trás dos transformers é usar atenção para processar sequências de dados, permitindo que a rede aprenda a extrair informações relevantes dessas sequências. Isso os torna muito eficientes para tarefas NLP como tradução automática, gerar texto, resposta a perguntas e outras. (FACE, 2021)

Figura 6 – Arquitetura *Transformer*



Fonte: (VASWANI et al., 2017)

A maior vantagem ao se utilizar *transormers* é devido a possibilidade de se utilizar modelos pré-treinados com por exemplo GPT, BERT, BART, T5, etc. Que foram treinados como modelos de linguagem, sendo treinados por grandes quantidadedes de texto bruto de forma auto-supervisionada. No aprendizado auto-supervisionado não se faz necessária a participação dos humanos para rotular os dados, deixando que a máquina calcule o objetivo automaticamente a partir das entradas do modelo. (FACE, 2021)

Utilizando destes modelos pré-treinados pode-se utilizá-los no método chamado aprendizagem de transferência onde estes modelos serão treinados desta vez de maneira supervisionada (utilizando-se de rótulos no conjunto de dados) para realizar uma determinada tarefa. (FACE, 2021)

3 Materiais e Métodos

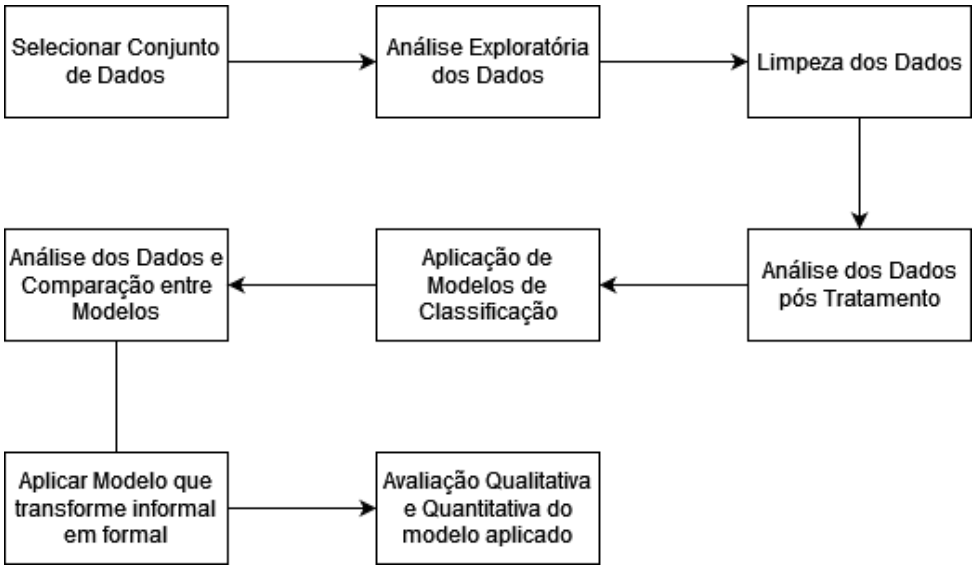
3.1 Considerações Iniciais

Neste capítulo, serão apresentados os materiais e métodos utilizados para alcançar os objetivos propostos neste trabalho. Será descrito o plano metodológico adotado, detalhando as etapas que serão seguidas para aplicar o modelo de aprendizado de máquina que realiza a transformação de textos informais em formais na língua portuguesa.

3.2 Plano Metodológico

Seguindo o fluxo da Figura 7 logo abaixo, temos a sequência de 8 passos que serão realizados neste trabalho.

Figura 7 – Fluxo do Projeto



3.2.1 Selecionar um conjunto de dados de texto na língua portuguesa, contendo exemplos de textos informais e formais

Para obter um bom resultado na aplicação de um modelo de IA que transforme texto informal em formal na língua portuguesa, é necessário selecionar um conjunto de dados adequado. Esse conjunto deve conter exemplos de textos informais e formais, permitindo o treinamento do modelo em pelo menos uma das abordagens utilizadas para esse fim.

A qualidade e a quantidade de dados presentes no conjunto têm um impacto significativo no desempenho e na precisão da IA. Portanto, é importante obter um conjunto de dados representativo da realidade, com uma amostra adequada de textos formais e informais, a fim de evitar que o modelo se baseie em uma perspectiva distorcida. Além disso, o conjunto de dados deve ser grande o suficiente para que o modelo possa aprender com uma quantidade adequada de exemplos.

Os dados também devem ser devidamente etiquetados como formais e informais para que possam ser utilizados no treinamento do modelo. Isso permite que o modelo aprenda a distinguir entre os dois tipos de texto e realize a transformação corretamente.

3.2.1.1 Análise exploratória dos dados.

A análise exploratória dos dados é uma etapa fundamental no desenvolvimento de uma inteligência artificial. Essa análise ajuda a compreender melhor as características dos dados e a identificar possíveis problemas ou necessidades de ajustes nos dados antes do treinamento do modelo.

Durante a análise exploratória dos dados, alguns aspectos relevantes podem ser observados e tratados, tais como:

- Verificação de valores nulos ou faltantes nos dados;
- Observação da distribuição de textos formais e informais no conjunto de dados;
- Cálculo de estatísticas descritivas, como a quantidade de caracteres em cada frase;
- Visualização de gráficos ou histogramas para entender melhor a distribuição dos dados.

Essa análise permite identificar problemas nos dados, como inconsistências, ruídos ou desequilíbrios, e tomar as medidas necessárias para corrigi-los ou ajustá-los, garantindo a qualidade dos dados utilizados no treinamento do modelo.

Para realizar a análise exploratória dos dados, todo o conjunto de dados será reunido em um único arquivo CSV, contendo o texto e uma etiqueta indicando se é formal ou informal. Em seguida, serão realizadas as etapas descritas acima, a fim de compreender melhor a natureza dos dados e prepará-los adequadamente para o treinamento do modelo.

Os procedimentos para a visualização e análise dos dados estão disponíveis no seguinte link do Colab[1]

[1]Google Colab Repositório. Disponível em: <https://colab.research.google.com/drive/16nhCvyBns2siY4wRXde4WDN63AhYCTfK?usp=sharing>

3.2.1.2 Limpeza dos dados

Após a análise exploratória dos dados, é necessário realizar a limpeza dos mesmos a fim de garantir a qualidade e adequação do conjunto de dados para o treinamento do modelo. A limpeza dos dados envolve a aplicação de uma série de procedimentos para remover ou corrigir elementos indesejados nos textos. Os seguintes procedimentos serão realizados:

- Remoção de valores nulos: Serão removidos os exemplos de textos que apresentem valores nulos, pois esses dados não são úteis para o treinamento do modelo.
- Remoção de frases duplicadas: Caso haja frases duplicadas no conjunto de dados, serão mantidas apenas uma ocorrência de cada frase, a fim de evitar redundâncias.
- Remoção de espaços vazios: Serão removidos os espaços vazios no início ou no final das frases, pois não são relevantes para o treinamento do modelo.
- Remoção de frases vazias: Serão removidas as frases que não contêm nenhum texto, pois não contribuem para o treinamento do modelo.
- Conversão para minúsculas: Todas as frases serão convertidas para minúsculas, a fim de uniformizar o texto e evitar diferenciação entre maiúsculas e minúsculas durante o treinamento.
- Remoção de *stop words* Seção 2.5: Serão removidas as *stop words* (palavras comuns que não possuem um significado relevante) das frases. Isso pode ser feito utilizando bibliotecas como NLTK ou Spacy, que possuem listas predefinidas de *stop words* para diferentes idiomas.
- Remoção de frases duplicadas novamente: Após a remoção das *stop words*, algumas frases podem se tornar idênticas. Portanto, será realizada uma nova verificação e remoção de frases duplicadas.
- Remoção de frases vazias novamente: Após a remoção das *stop words*, algumas frases podem se tornar vazias. Essas frases serão removidas do conjunto de dados.

Ao realizar esses procedimentos, o conjunto de dados estará limpo e pronto para ser utilizado no treinamento do modelo de transformação de texto informal em formal. Essa limpeza é essencial para garantir que o modelo receba apenas os dados relevantes e de qualidade, resultando em um treinamento mais eficiente e melhores resultados.

3.2.1.3 Análise de Dados Pós Tratamento

Após a realização dos procedimentos de limpeza dos dados, é importante realizar uma análise dos dados pós tratamento a fim de verificar sua viabilidade para o objetivo proposto. Nesta etapa, os seguintes dados serão observados:

- Distribuição de frases formais e informais: Será verificada a distribuição das frases entre os rótulos de formais e informais no conjunto de dados. Isso permite avaliar se existe um equilíbrio razoável entre as duas classes, evitando um viés em direção a uma classe específica.
- Quantidade de valores duplicados removidos: Será calculada a quantidade de valores duplicados que foram removidos durante o processo de limpeza. Essa informação é importante para verificar se havia uma quantidade significativa de duplicatas no conjunto de dados original.
- Quantidade de caracteres em cada frase: Será calculada a quantidade de caracteres em cada frase do conjunto de dados. Essa informação pode fornecer insights sobre a complexidade e extensão dos textos presentes no conjunto de dados.
- Nuvens de palavras de cada rótulo: Serão geradas nuvens de palavras para os rótulos de formais e informais separadamente. Essas nuvens de palavras representam as palavras mais frequentes em cada rótulo, permitindo identificar padrões e características distintas entre as classes.

A análise desses dados pós tratamento é essencial para garantir que o conjunto de dados esteja adequado e representativo para o treinamento do modelo. Além disso, ela fornece informações valiosas sobre a distribuição dos dados, a qualidade da limpeza realizada e as características dos textos presentes no conjunto de dados. Essa análise contribui para a compreensão e avaliação do conjunto de dados e auxilia na tomada de decisões para o desenvolvimento do modelo de transformação de texto informal em formal.

3.2.2 Aplicar modelo de classificação de frases formais e informais.

Nesta etapa, será realizada a classificação do conjunto de dados utilizando a biblioteca *Simpletransformers*, que facilita o treinamento e avaliação de modelos de aprendizado de máquina baseados em transformers. Será utilizado o modelo BERTimbau (SOUZA; NOGUEIRA; LOTUFO, 2020), um transformer BERT pré-treinado para o português brasileiro.

O objetivo é treinar um modelo de classificação capaz de distinguir entre frases formais e informais. Serão utilizadas as seguintes métricas para avaliar o desempenho do modelo:

- Precisão (*Precision*): mede a proporção de instâncias classificadas como formais que realmente são formais.
- Revocação (*Recall*): mede a proporção de instâncias formais que foram corretamente identificadas pelo modelo.
- F1-score: é uma medida de média harmônica entre a precisão e a revocação, que fornece uma medida geral do desempenho do modelo.
- Suporte (*Support*): indica o número de instâncias de cada classe no conjunto de dados.
- Acurácia (*Accuracy*): mede a proporção de instâncias corretamente classificadas em relação ao total de instâncias.

Além disso, será exibida a matriz de confusão, que mostra a distribuição das classificações feitas pelo modelo em relação às classes reais, permitindo uma análise mais detalhada do desempenho do modelo.

Essas métricas e a matriz de confusão fornecerão uma avaliação abrangente do desempenho do modelo de classificação de frases formais e informais, permitindo identificar sua eficácia na distinção entre as duas classes.

3.2.3 Aplicar um modelo de aprendizado de máquina para transformação de texto informal em formal.

Nesta etapa, será realizada a implementação de modelos *transformers* para a transformação de texto informal em formal nas línguas portuguesa e inglesa. Com base nos resultados obtidos nas etapas anteriores, serão testados diferentes modelos *transformers* para avaliar sua viabilidade nessa tarefa.

Os modelos *transformers* que serão utilizados serão:

- T5
- BART
- PTT5

Esses modelos têm demonstrado excelentes resultados em tarefas de processamento de linguagem natural e segundo a literatura são os mais adequados para a realização desta tarefa, os modelos T5 e BART serão utilizados no conjunto de dados *GYAFC Corpus* ao qual deu origem ao *XFORMAL Corpus* utilizado nas etapas anteriores e que será utilizado com o modelo PTT5(T5 em português brasileiro)(CARMO et al., 2020).

A mudança de estilo é uma forma de geração de texto controlada, esta tarefa envolve a reformulação de um texto de um estilo para o outro mantendo o mesmo conteúdo semântico. Modelos como o T5 (*Text-to-Text Transfer Transformer*) e o BART são especialmente adequados para essa tarefa. (*Bidirectional and Auto-Regressive Transformer*). (FACE, 2021)

Aqui estão algumas razões para esta afirmação, com base nas informações disponíveis a cerca de modelos *transformers* obtidas a partir do curso do Hugging Face (FACE, 2021) e de seus respectivos artigos de origem:

- Arquitetura flexível: Tanto o T5 como o BART são baseados na arquitetura do *Transformer*, que tem uma capacidade significativa de modelar dependências de longo alcance e aprender representações semânticas ricas. Além disso, ambos os modelos são sequenciais e bidirecionais, permitindo que eles entendam o contexto a partir de ambas as direções, o que é crucial para entender o contexto e o estilo do texto original.
- Abordagem Texto-a-Texto: O T5, especificamente, adota uma abordagem texto-a-texto. Ele enquadra todas as tarefas de NLP como tarefas de geração de texto, tornando-o extremamente versátil e adequado para tarefas complexas como a mudança de estilo de texto. (RAFFEL et al., 2019)
- Pré-treinamento *Denoising*: O BART utiliza um processo de pré-treinamento *denoising* (eliminação de ruído) que corrompe o texto de entrada de várias maneiras antes de aprender a reconstruí-lo. Isso o torna bom na geração de texto coerente e fluente, bem como na compreensão do conteúdo semântico e estilístico do texto original. (LEWIS et al., 2019)

Apesar de que outros modelos *Transformers* como o BERT (DEVLIN et al., 2018) e o GPT (RADFORD et al., 2018) sejam bem conhecidos por terem alcançado bons resultados em tarefas de PLN 2.3 eles possuem limitações de uso quando se trata de mudança de estilo de texto. O modelo BERT por exemplo é um modelo auto-encoder unidirecional que é treinado para prever palavras ocultas em uma frase, mas embora obtenha bons resultados em tarefas de classificação e previsão de texto não é adequado para a geração de texto, uma habilidade necessária para a mudança de estilo.

E apesar do GPT-2 ser um modelo auto-regressivo que gera texto palavra por palavra da esquerda para a direita e seja excelente na geração de texto fluente, possui dificuldade em manter consistência semântica ao longo de textos mais longos e como pode-se observar em seu artigo ***Language Models are Unsupervised Multitask Learners*** (RADFORD et al., 2018) sua performance em tarefas de tradução é bem inferior a outros modelos como T5 e BART.

Em resumo, a escolha do T5 e do BART para mudança de estilo não foi feita de forma arbitrária, mas sim fundamentada nas características distintas desses modelos o que os tornam adequados a realizar esta tarefa. Porém é importante notar que pesquisas PLN estão sempre avançando e novas técnicas e modelos podem surgir no futuro que sejam ainda melhores para essa e outras tarefas relacionadas.

Ao aplicar esses modelos, será necessário pré-processar o texto informal de entrada para adequá-lo ao formato exigido pelo modelo. Após a transformação, será possível gerar texto formal como resultado.

Essa etapa permitirá avaliar o desempenho e a eficácia de cada modelo na tarefa de transformação de texto informal em formal, possibilitando identificar qual modelo apresenta os melhores resultados e pode ser utilizado para essa finalidade.

3.2.4 Realizar uma avaliação quantitativa e qualitativa do modelo desenvolvido.

Na fase final deste estudo, realizaremos uma avaliação abrangente dos modelos que desenvolvemos, tanto de uma perspectiva qualitativa quanto quantitativa. Três conjuntos de modelos foram treinados, cada um abordando a tarefa de transformar texto informal em formal em diferentes contextos linguísticos: o PTT5 para o português brasileiro, e os modelos T5 e BART para inglês. O objetivo desta avaliação é verificar a eficácia desses modelos em transformar texto informal em texto formal, bem como comparar o desempenho dos modelos em dois idiomas diferentes.

Para a avaliação quantitativa, faremos uso de métricas estabelecidas na área do Processamento de Linguagem Natural (PLN), que são capazes de fornecer uma avaliação objetiva do desempenho dos nossos modelos. As métricas que iremos utilizar Subseção 2.3.1 incluem o METEOR e o ROUGE, que são comumente usados para avaliar tarefas de geração de texto. Estas métricas consideram aspectos como a precisão, o recall e a sobreposição de n-gramas entre as saídas geradas pelo modelo e as referências. Além disso, também consideraremos o BLEU, uma métrica popular na tarefa de tradução automática, que também se mostrou útil para avaliar a qualidade do texto gerado.

É importante ressaltar que essas métricas têm suas limitações e não capturam totalmente a fluência, a adequação ou a naturalidade do texto gerado, que são aspectos essenciais da mudança de estilo de texto. Desta forma a avaliação qualitativa complementará nossa avaliação quantitativa.

Aqui, analisaremos exemplos de saídas geradas pelos modelos para avaliar a qualidade do texto gerado. Embora as métricas quantitativas forneçam uma medida objetiva do desempenho, a avaliação qualitativa nos permite avaliar as nuances mais sutis da transformação de estilo. Isso incluirá verificar se o conteúdo do texto foi mantido durante

a transformação, se o texto gerado parece natural e fluente e se ele atinge o nível de formalidade desejado. Esta análise manual dos resultados nos permitirá ter uma melhor compreensão das forças e fraquezas dos nossos modelos, bem como identificar áreas para futuras melhorias.

Além disso, faremos uma comparação entre as versões em inglês e português. Nosso objetivo é entender se os modelos performam de maneira semelhante em diferentes idiomas e identificar qualquer desafio ou diferença notável na aplicação desses modelos em diferentes idiomas. Isso nos ajudará a entender melhor a capacidade desses modelos de generalizar para diferentes idiomas e estilos de texto.

Ao final desta avaliação, esperamos ter uma compreensão clara da eficácia de nossos modelos em realizar a tarefa de transformação de estilo de texto e como essa eficácia varia entre diferentes idiomas. Isso nos fornecerá *insights* valiosos para futuras pesquisas e desenvolvimento nesta área.

3.3 Ferramentas

3.3.1 Google Colab

O Google Colab é uma ferramenta bastante útil para o desenvolvimento de projetos de análise de dados e aprendizado de máquina. Ele fornece um ambiente de desenvolvimento baseado em nuvem, onde é possível escrever e executar códigos Python de forma interativa.(GOOGLE, 2019)

Uma das vantagens do Google Colab é que ele já vem pré-instalado com várias bibliotecas populares, como o *TensorFlow* e o *scikit-learn*, o que facilita o desenvolvimento de modelos de aprendizado de máquina. Além disso, o Colab oferece suporte para a utilização de *GPUs* e *TPUs*, o que permite acelerar o treinamento de modelos complexos.

Outro aspecto interessante do Google Colab é a possibilidade de criar notebooks, que são documentos interativos que combinam código, texto e visualizações. Isso torna mais fácil documentar e compartilhar o trabalho realizado, facilitando a colaboração entre os membros de uma equipe.

Por fim, o Google Colab também permite o armazenamento de arquivos na nuvem, o que facilita o acesso aos dados e a execução dos códigos em diferentes dispositivos.

No contexto deste trabalho, o Google Colab será uma ferramenta útil para a análise exploratória dos dados, a implementação e treinamento dos modelos de transformação de texto informal em formal, bem como para a avaliação dos resultados obtidos.

3.4 Bibliotecas

3.4.1 NLTK

A biblioteca NLTK (Natural Language Toolkit) é uma ferramenta poderosa para o processamento de linguagem natural em Python. Ela fornece uma ampla gama de recursos e funcionalidades para o tratamento e análise de texto. Alguns dos recursos mais comuns do NLTK incluem:

- Tokenização: dividir um texto em unidades menores, como palavras ou frases.
- Remoção de stopwords: eliminação de palavras comuns que geralmente não fornecem informações úteis, como artigos, preposições e pronomes.
- Stemming: redução de palavras à sua forma radical ou base, removendo sufixos e prefixos.
- Lemmatização: redução de palavras a uma forma canônica ou de dicionário, preservando sua categoria gramatical.
- Part-of-speech tagging: atribuição de etiquetas gramaticais a palavras em um texto, como substantivos, verbos, adjetivos, etc.
- Análise sintática: análise da estrutura gramatical de uma frase para identificar elementos como sujeito, objeto e predicado.
- Análise de sentimentos: determinação da polaridade ou emoção associada a um texto.
- WordNet: um dicionário lexical que fornece sinônimos, antônimos, hiperônimos, hipônimos e outras relações entre palavras.

Esses recursos do NLTK são úteis para pré-processamento de texto, extração de recursos, análise exploratória de dados e muito mais. Ele oferece uma ampla gama de funcionalidades e é amplamente utilizado na comunidade de processamento de linguagem natural.(BIRD; KLEIN; LOPER, 2009)

3.4.2 SpaCy

A biblioteca SpaCy é uma ferramenta poderosa e versátil de processamento de linguagem natural (PLN) para Python. Ela é amplamente utilizada no campo de PLN para realizar diversas tarefas, como tratamento e limpeza de dados, análise de sentimento, classificação de documentos e muito mais. (HONNIBAL; MONTANI, 2017)

Uma das funcionalidades mais comuns do SpaCy é a remoção de *stop words*, que são palavras comuns que não contribuem significativamente para o sentido de um texto, como artigos, preposições e pronomes. A remoção dessas palavras ajuda a reduzir a dimensionalidade do texto e focar nas palavras mais relevantes. O SpaCy oferece suporte para remoção de *stop words* em diversos idiomas, incluindo o português. (HONNIBAL; MONTANI, 2017)

Além disso, o SpaCy também fornece uma série de recursos para o processamento de textos brutos, como tokenização, lematização, identificação de entidades nomeadas, análise sintática e muito mais. Essas funcionalidades permitem extrair informações estruturadas dos textos e realizar análises mais avançadas. (HONNIBAL; MONTANI, 2017)

3.4.3 Pandas

O Pandas é uma biblioteca de código aberto para Python amplamente utilizada na análise de dados tabulares. Ela fornece estruturas de dados de alto desempenho e ferramentas de manipulação de dados que permitem aos usuários realizar uma variedade de tarefas relacionadas à análise e processamento de dados. (TEAM, 2020)

Uma das estruturas de dados principais fornecidas pelo Pandas é o *DataFrame*, que é uma tabela bidimensional com colunas nomeadas. O *DataFrame* é uma estrutura flexível que permite armazenar e manipular dados de diferentes tipos e realizar operações eficientes em colunas e linhas. Ele facilita a leitura e escrita de dados de diferentes formatos, como arquivos CSV, Excel, SQL, entre outros. (TEAM, 2020)

O Pandas oferece uma ampla gama de funcionalidades para lidar com dados, incluindo a capacidade de limpar e transformar dados, tratar valores ausentes, lidar com dados duplicados, aplicar filtros, realizar agregações e muito mais. Ele também suporta operações de junção e combinação de dados, permitindo que os usuários realizem análises complexas combinando diferentes conjuntos de dados. (TEAM, 2020)

Além disso, o Pandas é altamente integrado com outras bibliotecas populares do ecossistema de análise de dados em Python, como *NumPy*, *Matplotlib* e *Scikit-learn*, o que facilita a realização de análises mais avançadas e visualizações de dados. (TEAM, 2020)

3.4.4 NumPy

O NumPy é uma biblioteca de código aberto para Python que fornece suporte para operações matemáticas eficientes em arrays multidimensionais. Ele é amplamente utilizado em análise de dados, processamento numérico, computação científica e outros campos relacionados. (HARRIS et al., 2020)

O NumPy é uma biblioteca fundamental para a ciência de dados, pois muitas outras bibliotecas populares em Python, como Pandas, Matplotlib e Scikit-learn, depen-

dem dele para realizar operações numéricas eficientes em grandes conjuntos de dados. (HARRIS et al., 2020)

3.4.5 Matplotlib

O Matplotlib é uma biblioteca de visualização de dados em Python amplamente utilizada para criar gráficos e visualizações de alta qualidade. Ele oferece uma ampla gama de tipos de gráficos, incluindo gráficos de linha, gráficos de dispersão, histogramas, gráficos de barra, gráficos de pizza, gráficos de caixa e muitos outros. (HUNTER, 2007)

O Matplotlib é amplamente utilizado na ciência de dados, aprendizado de máquina, visualização de dados, análise exploratória e muitas outras áreas. Ele é frequentemente combinado com outras bibliotecas, como NumPy e Pandas, para manipulação e análise de dados antes de criar visualizações. (HUNTER, 2007)

3.4.6 Wordcloud

A biblioteca Wordcloud é uma ferramenta utilizada para visualização de nuvens de palavras em Python. Uma nuvem de palavras é uma representação gráfica de palavras que aparecem em um conjunto de textos, onde o tamanho de cada palavra é proporcional à sua frequência. (OESPER et al., 2011)

A biblioteca Wordcloud permite gerar nuvens de palavras de forma simples e flexível. Ela oferece diversas opções de personalização, como escolha de cores, formas, tamanhos de fonte, remoção de palavras irrelevantes (stop words) e ajuste de layout. Essas opções permitem criar visualizações atrativas e informativas que destacam as palavras mais relevantes em um conjunto de textos. (OESPER et al., 2011)

3.4.7 TensorFlow

O TensorFlow é uma biblioteca de código aberto desenvolvida pelo Google Brain Team para construir e treinar modelos de aprendizado de máquina, incluindo redes neurais profundas. É uma das bibliotecas mais populares e amplamente utilizadas para implementar algoritmos de inteligência artificial e processamento de dados. (ABADI et al., 2015)

O TensorFlow é projetado para oferecer uma interface flexível e eficiente para a criação de modelos de aprendizado de máquina. Ele suporta uma ampla gama de técnicas de aprendizado de máquina, como redes neurais convolucionais, redes neurais recorrentes, modelos generativos adversariais, entre outros. A biblioteca fornece operações de baixo nível para a manipulação de tensores, que são estruturas de dados multidimensionais usadas para representar dados nos modelos. Ele também fornece ferramentas e utilitários para ajudar no processo de treinamento e validação de modelos, como otimizadores, funções de perda, métricas de avaliação e recursos de visualização de dados. (ABADI et al., 2015)

3.4.8 Scikit-learn

O Scikit-learn é uma biblioteca de aprendizado de máquina em Python que fornece uma ampla gama de algoritmos e ferramentas para tarefas de aprendizado supervisionado e não supervisionado. É uma das bibliotecas mais populares e amplamente utilizadas em ciência de dados e aprendizado de máquina. (PEDREGOSA et al., 2011)

O Scikit-learn oferece uma variedade de algoritmos prontos para uso, incluindo classificação, regressão, agrupamento, redução de dimensionalidade, seleção de características, entre outros. Esses algoritmos são implementados de forma eficiente e otimizada, permitindo a aplicação em conjuntos de dados de diferentes tamanhos. (PEDREGOSA et al., 2011)

3.4.9 Seaborn

O Seaborn é uma biblioteca de visualização de dados em Python, baseada no Matplotlib. Ela fornece uma interface de alto nível para criação de gráficos estatísticos atraentes e informativos. O Seaborn é projetado para trabalhar bem com estruturas de dados do tipo DataFrame do Pandas, tornando-o uma escolha popular para análise exploratória de dados e visualização de resultados de modelos. (WASKOM et al., 2017)

O Seaborn oferece uma variedade de gráficos estatísticos, como gráficos de dispersão, gráficos de barras, gráficos de caixa, histogramas, gráficos de regressão, entre outros. Esses gráficos são estilizados e personalizáveis, permitindo que você ajuste facilmente as cores, os estilos dos marcadores, os esquemas de cores, as legendas e outros elementos visuais. (WASKOM et al., 2017)

3.4.10 PyTorch

A biblioteca PyTorch, conhecida comumente apenas como "torch", é uma das bibliotecas mais populares para a aprendizagem profunda (deep learning), juntamente com TensorFlow. Foi originalmente desenvolvida pela equipe de pesquisa da Meta AI e tem se destacado na comunidade científica e industrial pela sua flexibilidade e eficiência.

PyTorch é caracterizado pela sua interface intuitiva e fácil de usar. Ele utiliza tensores, que são semelhantes aos arrays do NumPy, mas podem ser utilizados em GPUs para aceleração de computação. Um tensor é uma generalização do conceito de matrizes para um número arbitrário de dimensões e é uma estrutura de dados fundamental em aprendizado profundo. (PASZKE et al., 2019)

3.5 Considerações Finais

O trabalho de conclusão de curso proposto é bastante abrangente e aborda uma questão relevante no processamento de linguagem natural, que é a transformação de textos informais em formais na língua portuguesa. As etapas planejadas, desde a seleção do conjunto de dados até a aplicação de modelos de aprendizado de máquina, estão bem estruturadas e devem fornecer resultados interessantes.

A seleção cuidadosa do conjunto de dados é fundamental para o desempenho do modelo, uma vez que dados de qualidade e representativos são necessários para treinar e avaliar o modelo adequadamente. A aplicação do modelo de classificação inicial é uma etapa importante para identificar a formalidade dos textos e direcionar a transformação correta.

A escolha de diferentes modelos de transformação de texto, como BART, PTT5 e T5, é interessante, pois permite comparar o desempenho e a eficácia de diferentes abordagens. Cada modelo tem suas características, vantagens e desvantagens, e é importante analisar esses aspectos para selecionar o mais adequado para o problema em questão.

A avaliação quantitativa e qualitativa do modelo desenvolvido é essencial para verificar sua eficácia e compará-lo com outras técnicas existentes.

No geral, o trabalho proposto apresenta uma abordagem completa e estruturada para a transformação de textos informais em formais, com a aplicação de modelos de aprendizado de máquina e uma avaliação rigorosa dos resultados. Com base nas etapas e ferramentas planejadas, espera-se que o trabalho contribua para avanços no processamento de linguagem natural e ofereça *insights* importantes sobre o tema em questão.

4 Resultados

4.1 Considerações Iniciais

Nesta seção serão descritos os resultados obtidos em cada etapa da execução do Capítulo 3 que implementa o Plano Metodológico para este trabalho.

4.2 Selecionar um conjunto de dados de texto na língua portuguesa, contendo exemplos de textos informais e formais

O conjunto de dados XFORMAL corpus, obtido a partir do artigo ***XFORMAL: A Benchmark for Multilingual Formality Style Transfer***(BRIAKOU et al., 2021), foi selecionado para o desenvolvimento do modelo de transformação de textos informais em formais. Ele é composto por pares de frases, em que cada frase possui uma versão informal e uma versão formal correspondente. O conjunto de dados abrange os idiomas inglês, francês, italiano e português, e está dividido em duas categorias principais: "Família e Relacionamentos"e "Entretenimento e Música".

No conjunto de treinamento, há cerca de 100 mil pares de exemplos em cada idioma, e no conjunto de teste, cerca de 10 mil pares de frases. Além disso, há um conjunto de dados separado para validação.

A obtenção do acesso aos dados do conjunto de dados ***L6 - Yahoo! Answers Comprehensive Questions and Answers version 1.0***(YAHOO!, 2007) foi um processo que envolveu o preenchimento de uma ficha, a assinatura de termos de conduta e de uso, e uma espera de cerca de 20 dias para a aprovação manual. Após a aprovação, foi encaminhado um e-mail de confirmação de acesso ao L6 para Joel Tetreault, que prontamente enviou o link com o conjunto de dados requisitado, incluindo o XFORMAL corpus e o conjunto de dados GYAFC.

Essa seleção de conjunto de dados é importante para o desenvolvimento do modelo, pois fornece exemplos de textos informais e suas correspondentes versões formais, permitindo o treinamento e a avaliação do modelo ao longo das etapas seguintes.

4.3 Análise Exploratória dos Dados

- **Primeiras linhas do conjunto de dados**

  Como descrito anteriormente na Subseção 3.2.1.1 o conjunto de dados possui o seu texto e o tipo de texto de acordo com sua *label* indicando se a frase se refere a informal quando a *label* possui valor 0 ou formal caso seja igual a 1 como pode ser observado na Tabela 2 abaixo.

índice	text	labels
0	Prefiro deixar o cara me perguntar.	1
1	Sofro por abuso verbal da minha esposa.	1
2	Você terá mais amigos do que você quer.	1
3	É bom que você possa ver fotos de quem você es...	1
4	Preciso saber o que fazer.	1

Tabela 2 – Primeiras Linhas do **Dataset**

- **informações Gerais do dataset**

  Utilizando o método *info* da biblioteca pandas é possível ver as informações gerais do conjunto de dados, dados estes dispostos logo abaixo na Tabela 3:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 337303 entries, 0 to 337302
Data columns (total 2 columns):

Column	Non-Null Count	Dtype
text	337303	object
labels	337303	int64

dtypes: int64(1), object(1)
memory usage: 5.1+ MB

Tabela 3 – Informações Gerais do Conjunto de Dados

- **Verificar se existem valores faltando**

  Não foi encontrado nenhum valor faltando como apresentado abaixo na Tabela 4 se utilizando o método *isna* para indicar se a linha está vazia e somando-se todas as ocorrências:

text	labels
0	0

Tabela 4 – Verifica Nulos

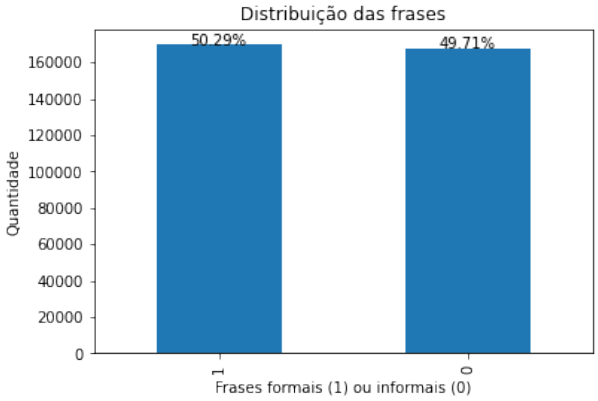
- **Distribuição de frases formais e informais**
  Como citado anteriormente o conjunto de dados possui uma amostra homogênea de frases informais e seu equivalente na forma culta, como pode ser visto na Tabela 5 e Figura 8 a pequena discrepância observada se dá pelo conjunto de amostras provenientes do grupo de validação que foi anexado ao conjunto de dados para esta análise.

formal	informal
169622	167681

Name: labels, dtype: int64

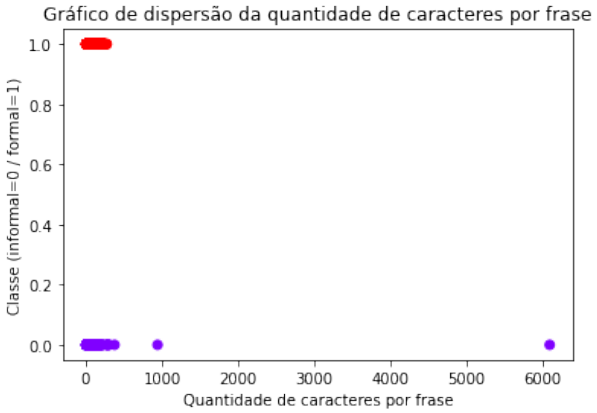
Tabela 5 – Proporção de Formal e Informal no *dataset*

Figura 8 – Distribuição de Frases Formais e Informais



- **Quantidade de caracteres por frase**
  Observando o conjunto de dados como exemplificado pela contagem de caracteres de cada frase na Tabela 6 e buscando a menor e a maior frase é perceptível a necessidade de limpeza dos dados visto que a menor frase contém 7 caracteres enquanto a maior possui 6092 caracteres contendo mais de 6 mil repetições do ponto de interrogação ("?"), como pode ser observado pelo gráfico de dispersão Figura 9 logo abaixo.

Figura 9 – Caracteres por Frase Dispersão



índice	text	labels	text_len
0	Prefiro deixar o cara me perguntar.	1	35
1	Sofro por abuso verbal da minha esposa.	1	39
2	Você terá mais amigos do que você quer.	1	39
3	É bom que você possa ver fotos de quem você es...	1	57
4	Preciso saber o que fazer.	1	26

Tabela 6 – Inicio do *Dataset* com Contagem de Caracteres

4.4 Análise de Dados Pós Tratamento

Ao realizar o tratamento foi verificado que a utilização da biblioteca *Spacy* para a retirada de *Stop Words* prejudicava o conjunto de dados resultando em uma classificação com baixa precisão e acurácia, desta forma os resultados a seguir foram realizados sem a utilização desta biblioteca.

- **Proporção de Formal e Informal no *dataset* pós tratamento**

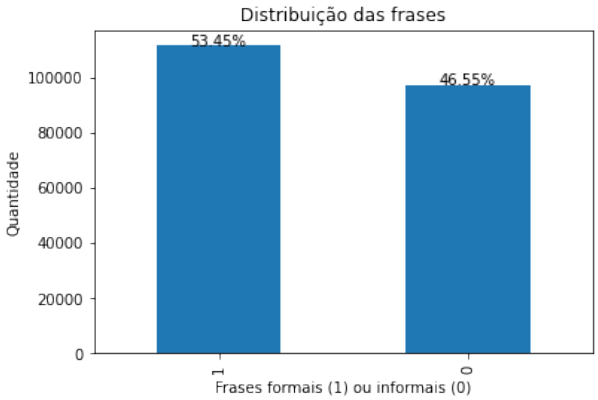
  Como pode-se observar pela Tabela 7 e na Figura 10 houve uma redução na quantidade de frases formais de 169.622 = 50,29% para 111.648 = 53,45% e de informais de 167.681 = 49,71% para 97.249 = 46,55% mantendo uma proporção aceitável para ser utilizado como conjunto de dados para classificação e transformação segundo a Seção 2.3

formal	informal
111648	97249

Name: labels, dtype: int64

Tabela 7 – Proporção de Formal e Informal no *dataset* pós Tratamento

Figura 10 – Distribuição de Frases Formais e Informais Pós Tratamento



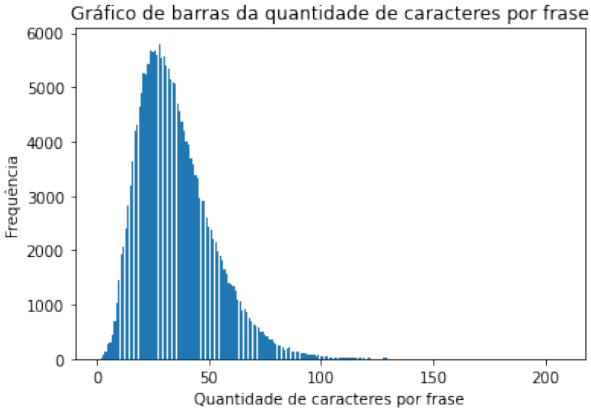
- **Remoção de duplicatas**

  Ao realizar a limpeza dos dados foram observadas e removidas mais de 116 mil frases que possuíam duplicatas antes do tratamento dos dados e outros 12 mil após a limpeza destes dados através das técnicas mencionadas na Subseção 3.2.1.2, conservando apenas uma ocorrência da mesma.

- **Quantidade de caracteres por frase pós tratamento**

  Após a limpeza dos dados pode-se analisar novamente a quantidade de caractere por frase possuindo agora 1 caractere na menor frase do *dataset* contra 7 antes do tratamento dos dados e 207 contra 6092 da maior frase como pode ser observado na Figura 11 e no Apêndice A Figura 22 e Figura 23 que demonstram como está a distribuição de quantidade de caracteres por frase no conjunto de dados.

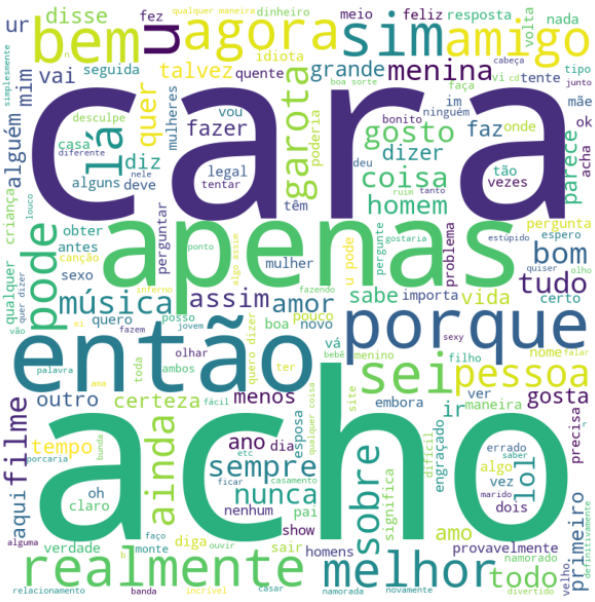
Figura 11 – Caracteres por Frase Pós Tratamento



- **Nuvem de Palavras rotuladas como Informais**

  A nuvem de palavras Figura 12 exibe de maneira gráfica o conjunto de palavras que mais se repete no conjunto de dados com rótulo que indica ser informal.

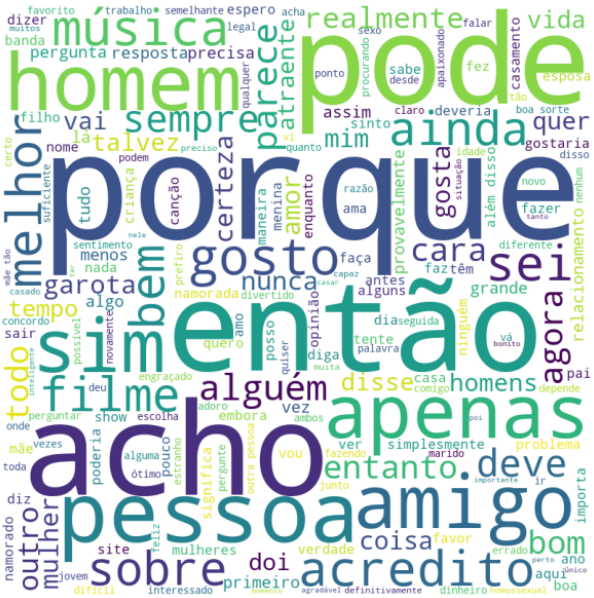
Figura 12 – Nuvem de Palavras Informais



- **Nuvem de Palavras rotuladas como Formal**

  A nuvem de palavras Figura 13 exibe de maneira gráfica o conjunto de palavras que mais se repete no conjunto de dados com rótulo que indica ser formal.

Figura 13 – Nuvem de Palavras Formais



4.5 Aplicar modelo de classificação de frases formais e informais.

Foram utilizados outros métodos *transformers* como RoBERTa e o PTT5 mas a alta necessidade de processamento os tornaram inviáveis para se utilizar para classificação. Modelos mais simples como Regressão Logística também foram aplicados obtendo resul-

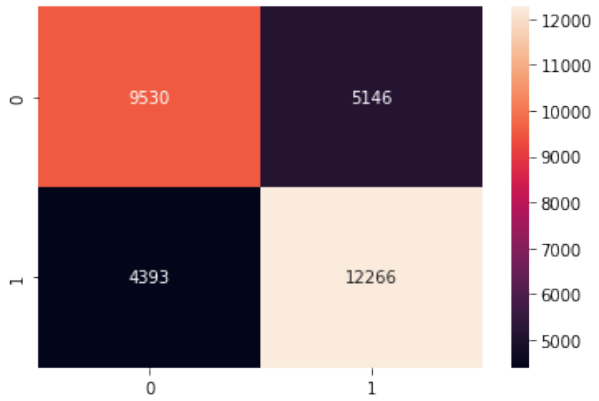
tados consideravelmente inferiores aos obtidos com o *transformer* BERTimbau, ibtendo apenas 65% de precisão e 45% de acurácia o que se mostrou completamente ineficiente para classificar texto formal e informal.

A análise dos resultados obtidos com a aplicação deste modelo pode ser vista na Tabela 8 e na Matriz de Confusão na Figura 14

Labels	Precision	Recall	F1-Score	Support
0	0.68	0.65	0.67	14676
1	0.70	0.74	0.72	16659
Accuracy			0.70	31335
Macro avg			0.69	31335
Weighted avg			0.69	31335

Tabela 8 – Resultados do Treinamento Utilizando BERTimbau

Figura 14 – Matriz de Confusão



4.6 Aplicar um modelo de aprendizado de máquina para transformação de texto informal em formal.

Seguindo com o planejado nesta etapa foram aplicados modelos *transformers* a fim de que se obter um modelo que pudesse realizar a mudança de estilo de textos informais para formais na língua portuguesa utilizando o PTT5 e na lingua inglesa utilizando os transformers T5 e BART.

As seguintes etapas foram necessárias para a aplicação dos modelos:

- **Preparação do *dataset***

  Nesta etapa o conjunto de dados proveniente do XFORMAL Corpus foi separado em um arquivo "csv"com a seguinte estrutura, uma coluna chamada "informal_text"para as frases informais e seu equivalente na linguagem formal no mesmo índice na coluna "formal_text".

– Conjunto de dados em Português:

Figura 15 – Conjunto de dados em português brasileiro

	formal_text	informal_text
0	Prefiro deixar o cara me perguntar.	Claro, está tudo bem, mas eu sempre deixei o c...
1	Sofro por abuso verbal da minha esposa.	Sou um cara que sofre de abuso verbal da minha...
2	Você terá mais amigos do que você quer.	Você terá mais amigos que você quer...;)
3	É bom que você possa ver fotos de quem você es...	É bom, você pode ver fotos de quem você está f...
4	Preciso saber o que fazer.	EU PRECISO SABER O QUE 2 FAZER

– Conjunto de dados em Inglês:

Figura 16 – Conjunto de dados em inglês

	formal_text	informal_text
0	I prefer to let the guy ask me.	Sure, it's ok, but I always have let the guy a...
1	I suffer through verbal abuse from my wife.	Hmmm, I'm a guy suffering from verbal abuse fr...
2	You will have more friends than you want.	You will have more friends that you want... ;)
3	It's nice that you get to see pictures of who ...	It's nice, you get to see pictures of who you ...
4	I need to know what to do.	I NEED TO KNOW WHAT 2 DO

- **Criação da classe "*Dataset*"**

  Para que os dados possam ser utilizados pela biblioteca transformers uma classe "*Dataset*"do PyTorch 3.4.10 se faz necessária, com isso foi criada a classe "FormalityDataset"para este fim.

- **Separação de treino e teste do conjunto de dados**

  Os dados são separados em treino utilizando 80% do total de amostras, com 10% para *tuning* e 10% para teste, e após isso aplica-se a classe "FormalityDataset"criada na etapa anterior para que sejam utilizadas pelos modelos.

- **Definição de hiperparâmetros**

  Devido a algumas limitações por utilizar a plataforma Google Colab para a aplicação dos modelos os hiperparâmetros utilizados foram os seguintes:

Hiperparâmetro	**Valor**
Número de épocas de treinamento	5
Tamanho do batch por dispositivo de treinamento	16
Tamanho do batch por dispositivo de avaliação	16
Número de passos de aquecimento	500
Taxa de decaimento do peso	0.01
Intervalo de passos para registrar os logs	100
Estratégia de salvamento	'epoch'
Número máximo de checkpoints a serem salvos	2
Estratégia de avaliação	'epoch'
Carregar o melhor modelo no final do treinamento	True
Tamanho máximo da sequência	128

Tabela 9 – Hiperparâmetros utilizados para treinamento dos modelos T5 e PTT5

Hiperparâmetro	**Valor**
Número de épocas de treinamento	5
Tamanho do batch por dispositivo de treinamento	32
Tamanho do batch por dispositivo de avaliação	32
Número de passos de aquecimento	500
Taxa de decaimento do peso	0.01
Intervalo de passos para registrar os logs	100
Estratégia de salvamento	'epoch'
Número máximo de checkpoints a serem salvos	2
Estratégia de avaliação	'epoch'
Carregar o melhor modelo no final do treinamento	True
Tamanho máximo da sequência	128

Tabela 10 – Hiperparâmetros utilizados para treinamento do modelo BART

- **Treinar os modelos**

  – Ao realizar o treinamento com o *transformer* PTT5 foram obtidas as seguintes perdas Tabela 11 e os parâmetros do modelo podem ser vistos no Apêndice B Tabela 15 com a descrição de seus atributos na Tabela 16

Epoch	**Training Loss**	**Validation Loss**
1	0.235800	0.218133
2	0.221400	0.209989
3	0.206500	0.206422
4	0.204000	0.205138
5	0.200900	0.204861

Tabela 11 – Perda durante o treinamento e validação para o modelo PTT5

  – Ao realizar o treinamento com o *transformer* BART foram obtidas as seguintes perdas Tabela 12 e os parâmetros do modelo podem ser vistos no Apêndice B Tabela 17 com a descrição de seus atributos na Tabela 18

Epoch	Training Loss	Validation Loss
1	0.177100	0.161664
2	0.155100	0.157823
3	0.141400	0.155332
4	0.126700	0.156267
5	0.118900	0.157206

Tabela 12 – Perda durante o treinamento e validação para o modelo BART

– Ao realizar o treinamento com o *transformer* T5 foram obtidas as seguintes perdas Tabela 13 e os parâmetros do modelo podem ser vistos no Apêndice B nas Tabelas 19 e 21 com a descrição de seus atributos nas Tabelas 20 e 22

Epoch	Training Loss	Validation Loss
1	0.167500	0.154014
2	0.158300	0.150943
3	0.146700	0.149869
4	0.140200	0.149746
5	0.141100	0.149949

Tabela 13 – Perda durante o treinamento e validação para o modelo T5

Após o treinamento os modelos e seus *tokenizers* foram salvos, todos estes passos podem ser vistos no colab de cada modelo, PTT5[1], T5[2], BART[3]

- **Realizar testes nos modelos**

  Com os modelos treinados foram realizados testes utilizando 10% do conjunto de dados para serem utilizados na fase de avaliação quantitativa utilizando as métricas BLEU, ROUGE e METEOR.

[1]Google Colab modelo PTT5 Repositório. Disponível em: <https://colab.research.google.com/drive/1avjJg8NEXY3UCpG6hD4VOsp0uG3Xo_Fw?usp=sharing>

[2]Google Colab modelo T5 Repositório. Disponível em: <https://colab.research.google.com/drive/13OlJH3L0lsQHPWOfWooM5VBw4eXEm7nu?usp=sharing>

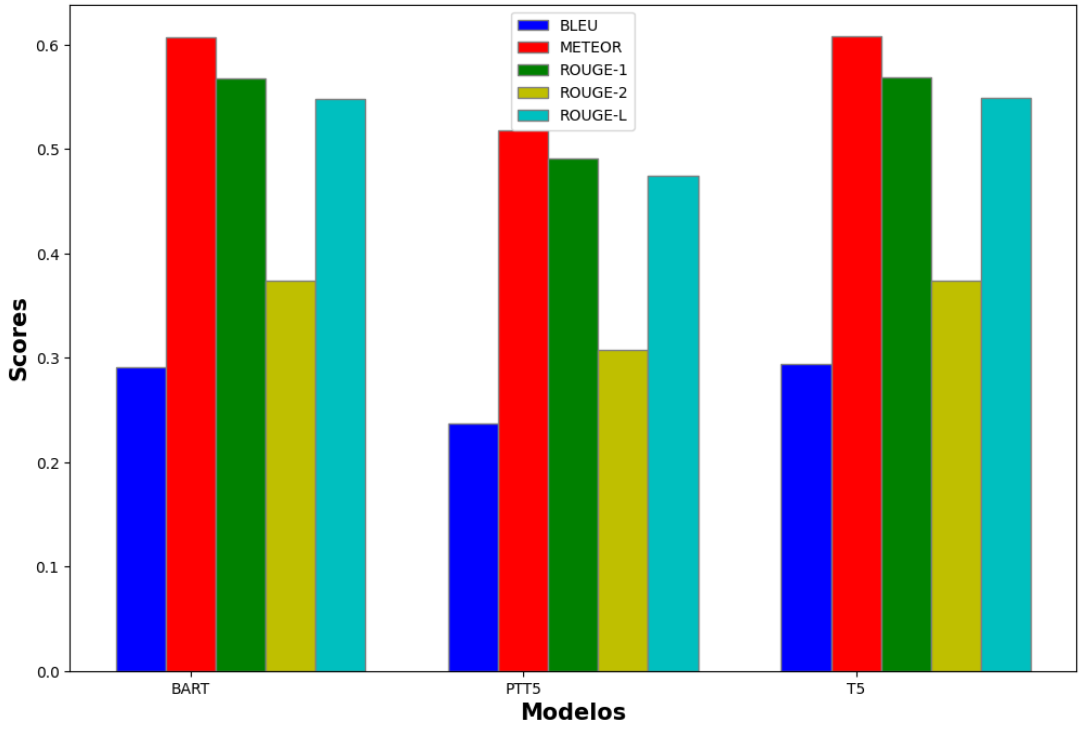
[3]Google Colab modelo BART Repositório. Disponível em: <https://colab.research.google.com/drive/1Nuhcxld3Omz57l8KFeIPIcMIbo_JPw8A?usp=sharing>

4.7 Realizar uma avaliação quantitativa e qualitativa do modelo desenvolvido.

4.7.1 Avaliação Quantitativa

Na avaliação quantitativa dos modelos, é possível observar uma diferença significativa entre os modelos treinados em inglês (BART e T5) e o modelo PTT5 treinado em português. As métricas BLEU, ROUGE e METEOR são amplamente utilizadas para avaliar a qualidade das saídas dos modelos de geração de linguagem, e todas elas indicam que os modelos em inglês tiveram desempenho superior ao modelo PTT5 como pode ser visto na Figura 17 e na Tabela 14.

Figura 17 – Análise quantitativa dos modelos



Modelo	BLEU Score	ROUGE Score	METEOR Score
BART	0.291	0.558/0.369/0.540	0.607
PTT5	0.237	0.484/0.304/0.468	0.518
T5	0.294	0.562/0.371/0.542	0.608

Tabela 14 – Desempenho dos modelos BART, PTT5 e T5. Os escores ROUGE são apresentados como ROUGE-1/ROUGE-2/ROUGE-L.

Os escores BLEU dos modelos BART e T5 são respectivamente 0.291 e 0.294, enquanto o escore BLEU do modelo PTT5 é 0.237. A diferença entre as métricas pode ser atribuída ao fato de que o modelo PTT5 foi treinado em um conjunto de dados que é

uma tradução literal dos dados originais em inglês, o que pode levar a frases sem sentido ou estruturas gramaticais não naturais.

Quanto às métricas ROUGE, que medem a sobreposição de unigramas, bigramas e *longest common subsequences* (sequências comuns mais longas) entre as saídas geradas e as referências, observa-se também que os modelos BART e T5 superaram o PTT5 em todas as sub-métricas (ROUGE-1, ROUGE-2 e ROUGE-L).

Finalmente, o escore METEOR, que considera sinônimos, paráfrases, ordem das palavras e stemização, reflete um padrão similar, com BART e T5 superando o PTT5.

Porém vale ressaltar que apesar de serem utilizadas para medir o desempenho dos modelos PLN estas métricas não são absolutas, ainda se faz necessária a avaliação humana para que seja analisada a performance dos modelos, um BLEU para mudança de estilo de informal para formal que tem padrão ouro por volta de 0,5, não necessariamente indica uma boa performance ao ser feita a avaliação humana, enquanto a métrica METEOR em relação a preservação de semântica se aproxima mais da avaliação humana (JIN et al., 2022).

4.7.2 Avaliação Qualitativa

A diferença de desempenho quantitativo entre os modelos sugere que as saídas do PTT5 não são tão coerentes, fluentes ou precisas em relação ao conteúdo original quanto as dos modelos BART e T5, modelos estes que ao inserir frases para que seja feita a avaliação qualitativa, obtemos praticamente o mesmo resultado, como pode ser visto na Figura 18 referente ao modelo BART e na Figura 19 do modelo T5.

Figura 18 – Análise qualitativa do modelo BART

```
Digite uma frase informal:  WHY DOES EVERYONE SEEM TO WANT SOMEONE THAT THEY CAN'T HAVE AND KNOW THAT THEY SHOULDN'T WANT.
Frase formal: Why does everyone seem to want someone that they cannot have and know that they should not
Digite uma frase informal: Do you like to argue and fight???
Frase formal: Do you like to argue and fight?
Digite uma frase informal: So if you're set on that, that's the way to go!!
Frase formal: If you are set on that, that is the way to go!
Digite uma frase informal: She's strong and loving and most of all...really wasn't as happy as she thought she was.
Frase formal: She is strong and loving, but she was not as happy as she thought she was
Digite uma frase informal: its one way of saying i don't like you!
Frase formal: It is one way of saying I do not like you.
Digite uma frase informal: IT DOESN'T MEAN THAT YOU ARE SLUTTY.
Frase formal: It does not mean that you are slutty.
Digite uma frase informal: Stay 100 miles away from this guy.
Frase formal: Stay 100 miles away from this man.
Digite uma frase informal: (saying sorry to him without commiting a mistake is humilation to ur self...AND DON'T EVER GET WEAK!!)
Frase formal: Saying sorry to him without commiting a mistake is humilation to yourself, and
Digite uma frase informal: THEY BOTH DONT RESPECT YOUR FEELING.
Frase formal: They both do not respect your feelings.
Digite uma frase informal: i can onli say...women r complicated...
Frase formal: I can say that women are complicated.
Digite uma frase informal: We like soft approch and a lil bit of chasing.
Frase formal: We like soft approch and a little bit of chasing.
Digite uma frase informal: Hey, I'm in NYC I'll help you out if your around!
Frase formal: I am in New York City and I will help you out if you are there.
```

A diferença de resultado observado entre os modelos em inglês se resume a falta de pontuação em algumas respostas e a reescrita de fases com palavras aparentemente desconhecidas pelo modelo.

Figura 19 – Análise qualitativa do modelo T5

```
Digite uma frase informal: WHY DOES EVERYONE SEEM TO WANT SOMEONE THAT THEY CAN'T HAVE AND KNOW THAT THEY SHOULDN'T WANT.
Frase formal: Why does everyone seem to want someone they can't have and know they shouldn't want
Digite uma frase informal: Do you like to argue and fight???
Frase formal: Do you like to argue and fight?
Digite uma frase informal: So if you're set on that, that's the way to go!!
Frase formal: If you are set on that, that is the way to go.
Digite uma frase informal: She's strong and loving and most of all...really wasn't as happy as she thought she was.
Frase formal: She is strong and loving, but she was not as happy as she thought she was.
Digite uma frase informal: its one way of saying i don't like you!
Frase formal: It is one way of saying I do not like you.
Digite uma frase informal: IT DOESN'T MEAN THAT YOU ARE SLUTTY.
Frase formal: It does not mean that you are slut.
Digite uma frase informal: Stay 100 miles away from this guy.
Frase formal: Stay 100 miles away from this guy.
Digite uma frase informal: (saying sorry to him without commiting a mistake is humilation to ur self...AND DON'T EVER GET WEAK!!)
Frase formal: Saying sorry to him without commiting a mistake is humilation to yourself.
Digite uma frase informal: THEY BOTH DONT RESPECT YOUR FEELING.
Frase formal: They both do not respect your feelings.
Digite uma frase informal: i can onli say...women r complicated...
Frase formal: I can onli say that women are complicated.
Digite uma frase informal: We like soft approch and a lil bit of chasing.
Frase formal: We like soft approch and a little bit of chasing.
Digite uma frase informal: Hey, I'm in NYC I'll help you out if your around!
Frase formal: I am in New York City and I will help you out if you are around.
```

Observando as respostas obtidas utilizando o PTT5 Figura 20 com o conjunto de dados XFORMAL Corpus que é uma tradução literal do conjunto de dados GYAFC Corpus utilizados nos modelos em inglês, observamos que além da reescrita de palavras desconhecidas o modelo algumas vezes muda o sentido da frase como na ultima frase onde a pergunta para manter o sentido original deveria ser reescrita de forma a manter a primeira pessoa do plural na pergunta, resultando em algo como "Vamos combinar de sair e desfrutar de algumas bebidas no fim de semana?"ao invés de mudar para segunda pessoa do singular resultando em "Você pode sair e tomar umas cervejas no fds?".

Figura 20 – Análise qualitativa do modelo PTT5

```
Digite uma frase informal: Como um cara que geralmente não gosta de filmes românticos, isso foi muito bom.
Frase formal: Como um homem que geralmente não gosta de filmes românticos, isso foi muito bom.
Digite uma frase informal: É a maior porcaria que já vi há séculos.
Frase formal: É a pior coisa que já vi há séculos.
Digite uma frase informal: omg eu amo essa música estou ouvindo agora
Frase formal: Olá, eu amo essa música, estou ouvindo agora.
Digite uma frase informal: Não, não, isso foi TESTEMUNHA...
Frase formal: Não, isso foi uma conversa.
Digite uma frase informal: Onde no mundo você chegar a estas perguntas????
Frase formal: Onde você chegou a estas perguntas?
Digite uma frase informal: Não pareço bonito ou o quê?
Frase formal: Não pareço bonito ou o que é?
Digite uma frase informal: Os bons velhos tempos se foram!
Frase formal: Os bons tempos se foram.
Digite uma frase informal: Eu não posso acreditar que você fez essa pergunta, mas eu vou respondê-la.
Frase formal: Eu não posso acreditar que você fez essa pergunta, mas eu vou respondê-la.
Digite uma frase informal: Eu acho que era uma menina feia lol
Frase formal: Acho que era uma menina feia.
Digite uma frase informal: O cara fugiu com duas mulheres.
Frase formal: O homem fugiu com duas mulheres.
Digite uma frase informal: bota um prato de comida pra mim aí mano!!!!
Frase formal: Você deveria me dar um prato de comida.
Digite uma frase informal: bora sair e tomar umas no fds?
Frase formal: Você pode sair e tomar umas cervejas no fds?
```

A diferença nos resultados ressalta a importância do contexto linguístico e cultural no treinamento de modelos de linguagem. A tradução literal dos dados de treinamento pode não ser suficiente para capturar nuances linguísticas e culturais, o que pode resultar em saídas de menor qualidade quando o modelo é aplicado a dados do mundo real. Como pode ser visto na Figura 21 que exibe algumas frases do conjunto de dados de treinamento, onde a primeira frase não foi completamente traduzida e a ultima provavelmente se refira a pedir para alguém ser preso, mas que em português não possuem sentido.

Figura 21 – Frases presente no conjunto de dados do XFORMAL Corpus

```
DEPOIS QUE NADA SERÁ MATTER.SO, É MELHOR APRECIAR IT.AFTER TUDO U TEM UMA CHANCE DO TEMPO DA VIDA.
Boa sorte Com Encontrar Amor se você ainda não tem!!!!
sair com as meninas com mais freqüência ou fazer algumas namoradas.
mas se as coisas não podem funcionar, então chegar a uma compreensão do que você quer fazer.
não isso não deve preocupar-lhe sua irmã está indo apenas relax.have um bom tempo com fora dele.
Pls não torná-lo uma gaiola.
```

Essas observações apontam para a necessidade de uma melhor adaptação dos modelos de linguagem para diferentes idiomas e contextos, o que pode incluir o uso de dados de treinamento de alta qualidade e traduzidos de forma idiomática, bem como a realização de mais pesquisas sobre como adaptar efetivamente os modelos de linguagem para idiomas específicos.

5 Discussão sobre métodos alternativos para mudança de estilo

Existem várias técnicas para realizar mudanças de estilo em textos, cada uma com suas vantagens e desvantagens. Neste estudo, adotamos o uso de modelos de aprendizado profundo, como BART, PTT5 e T5, que são baseados na arquitetura Transformer (VASWANI et al., 2017). No entanto, existem outras técnicas que poderiam ser consideradas, como a substituição baseada em regras, redes neurais convolucionais (CNNs), redes neurais recorrentes (RNNs) e autoencoders variacionais (VAEs).

A substituição baseada em regras, por exemplo, é um método simples e transparente que envolve a substituição de palavras ou frases por suas equivalentes mais formais. No entanto, esse método pode falhar ao lidar com a complexidade e as nuances da linguagem natural, como a dependência do contexto e as variações regionais (JURAFSKY; MARTIN, 2009).

As CNNs e as RNNs são outras duas arquiteturas de aprendizado de máquina que foram amplamente utilizadas para tarefas de processamento de linguagem natural. As CNNs são especialmente boas em capturar padrões locais e podem ser eficazes na mudança de estilo em nível de palavra ou frase. Por outro lado, as RNNs são capazes de modelar dependências temporais e sequenciais, tornando-as úteis para tarefas de mudança de estilo em nível de sentença (CHO et al., 2014).

No entanto, ambas as arquiteturas têm suas limitações. As CNNs podem não ser capazes de capturar dependências de longo alcance, enquanto as RNNs podem sofrer do problema do esquecimento de longo prazo. Além disso, ambas as arquiteturas podem ser menos eficazes ao lidar com textos longos, que são comuns em tarefas de mudança de estilo (HOCHREITER et al., 2001).

Os VAEs, por outro lado, são uma forma de autoencoder que adiciona uma camada de aleatoriedade à codificação do texto, o que pode permitir uma maior variabilidade nas saídas geradas. No entanto, eles também têm suas desvantagens. Por exemplo, a natureza estocástica dos VAEs pode levar a saídas menos determinísticas, e a sua formação pode ser mais complexa e computacionalmente intensiva (BOWMAN et al., 2015).

Neste estudo, optamos por usar modelos baseados em Transformers, devido às suas várias vantagens. Os Transformers são capazes de capturar dependências de longo alcance graças à sua atenção de múltiplas cabeças. Além disso, eles podem ser facilmente paralelizados durante o treinamento, o que os torna mais escaláveis para lidar com textos longos (FACE, 2021).

No entanto, os modelos de Transformers também têm suas desvantagens. Por exemplo, eles podem ser computacionalmente intensivos para treinar e exigir grandes quantidades de dados para apresentar bom desempenho. Além disso, suas saídas podem ser difíceis de interpretar devido à falta de transparência do modelo (VASWANI et al., 2017).

No geral, a escolha do método para mudança de estilo pode depender de várias considerações, incluindo a disponibilidade de dados, a capacidade de computação, a complexidade do estilo de destino e o nível de formalidade requerido. No futuro, pesquisas podem explorar a combinação dessas técnicas para melhorar a qualidade da mudança de estilo.

6 Conclusão

O presente estudo teve como objetivo primordial a aplicação de um modelo de Aprendizado Profundo 2.7 com vistas à realização de uma transição de estilo de textos informais para seus correspondentes na linguagem culta/formal da língua portuguesa. Para tanto, diversas etapas foram meticulosamente percorridas, a começar pela seleção e aquisição do conjunto de dados, seguida de sua respectiva análise e tratamento. Posteriormente, procedeu-se com a implementação de modelos *transformers*, permitindo que alcançássemos o propósito estabelecido.

No processo, conseguimos analisar o desempenho de tais modelos e efetuar uma comparação entre eles. A partir dessa avaliação, foi possível obter uma melhor compreensão dos resultados alcançados, assim como elencar possíveis melhorias que poderiam ser propostas para pesquisas e trabalhos futuros.

No âmbito deste estudo, observou-se que os modelos baseados na língua inglesa apresentaram um desempenho superior em relação ao modelo baseado na língua portuguesa. Esta disparidade pode ser atribuída, em parte, ao fato do conjunto de dados, XFORMAL Corpus, ser uma tradução literal dos dados originais em inglês, o que em alguns casos resultou em frases sem sentido em português.

Além disso, a vasta quantidade de recursos disponíveis e o nível de refinamento dos modelos em inglês, comparado com os recursos mais limitados para a língua portuguesa, podem ter contribuído para este resultado. É importante ressaltar que a qualidade e diversidade do conjunto de dados têm um papel fundamental na performance dos modelos de aprendizado profundo.

O resultado do experimento revelou que, apesar dos modelos terem se mostrado eficazes na tarefa proposta, há espaço para aperfeiçoamento. Como primeira sugestão para estudos futuros, recomenda-se a utilização de um conjunto de dados que melhor represente as diversas variações culturais e regionais do Brasil. Essa medida poderia tornar o modelo ainda mais robusto e preciso na tarefa de estilização do texto.

Além disso, é prudente considerar aprimorar o conjunto de dados atual, o XFORMAL Corpus. O emprego de APIs de modelos LLM (*Large Language Models*) poderia ser uma estratégia eficaz nesse sentido, uma vez que ela permitiria aprimorar os dados já existentes, de forma a aumentar a qualidade das saídas produzidas pelo modelo.

Em suma, a presente pesquisa alcançou seu objetivo de aplicar o Aprendizado Profundo para realizar a mudança de estilo de texto, mas o caminho para aprimoramentos futuros permanece aberto e repleto de possibilidades. Com a crescente evolução das téc-

nicas de aprendizado de máquina, antecipamos que as próximas iterações desse trabalho serão capazes de lidar de forma ainda mais eficaz com a tarefa proposta.

Referências

ABADI, M. et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. Software available from tensorflow.org. Disponível em: <https://www.tensorflow.org/>. Citado na página 35.

BANERJEE, S.; LAVIE, A. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In: GOLDSTEIN, J. et al. (Ed.). *IEEvaluation@ACL*. Association for Computational Linguistics, 2005. p. 65–72. Disponível em: <http://dblp.uni-trier.de/db/conf/acl/ieevaluation2005.html#BanerjeeL05>. Citado na página 18.

BIRD, S.; KLEIN, E.; LOPER, E. *Natural language processing with Python: analyzing text with the natural language toolkit*. [S.l.]: "O'Reilly Media, Inc.", 2009. Citado 2 vezes nas páginas 18 e 33.

BOWMAN, S. R. et al. Generating sentences from a continuous space. *CoRR*, abs/1511.06349, 2015. Disponível em: <http://dblp.uni-trier.de/db/journals/corr/corr1511.html#BowmanVVDJB15>. Citado na página 52.

BRIAKOU, E. et al. Xformal: A benchmark for multilingual formality style transfer. *arXiv preprint arXiv:2104.04108*, 2021. Citado na página 38.

CARMO, D. et al. Ptt5: Pretraining and validating the t5 model on brazilian portuguese data. *CoRR*, abs/2008.09144, 2020. Disponível em: <http://dblp.uni-trier.de/db/journals/corr/corr2008.html#abs-2008-09144>. Citado na página 29.

CHO, K. et al. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. 2014. Cite arxiv:1406.1078Comment: EMNLP 2014. Disponível em: <http://arxiv.org/abs/1406.1078>. Citado na página 52.

DEVLIN, J. et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2018. Cite arxiv:1810.04805. Disponível em: <http://arxiv.org/abs/1810.04805>. Citado na página 30.

FACE, H. *NLP Course*. 2021. Disponível em: <https://huggingface.co/course/pt/chapter1/2?fw=pt>. Citado 5 vezes nas páginas 16, 23, 24, 30 e 52.

GANEGEDARA, T.; LOPATENKO, A. *Natural Language Processing with TensorFlow*. [S.l.]: Packt Publishing Ltd, 2022. ISBN 9781838647742. Citado na página 16.

GÉRON, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. [S.l.]: O'Reilly Media, Inc., 2019. Citado 5 vezes nas páginas 17, 19, 20, 21 e 22.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. Cambridge, MA: MIT Press, 2016. Citado 2 vezes nas páginas 22 e 23.

GOOGLE. *Google Colaboratory*. 2019. Disponível em: <https://colab.research.google.com/>. Citado na página 32.

HARRIS, C. R. et al. Array programming with NumPy. *Nature*, Springer Science and Business Media LLC, v. 585, n. 7825, p. 357–362, set. 2020. Disponível em: <https://doi.org/10.1038/s41586-020-2649-2>. Citado 2 vezes nas páginas 34 e 35.

HOCHREITER, S. et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In: KREMER; KOLEN (Ed.). *A Field Guide to Dynamical Recurrent Neural Networks*. [S.l.]: IEEE Press, 2001. Citado na página 52.

HONNIBAL, M.; MONTANI, I. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear. 2017. Citado 2 vezes nas páginas 33 e 34.

HUNTER, J. D. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, IEEE COMPUTER SOC, v. 9, n. 3, p. 90–95, 2007. Citado na página 35.

JIANG, H. *Machine learning fundamentals : a concise introduction*. [S.l.]: Cambridge University Press, 2021. ISBN 9781108837040. Citado na página 15.

JIN, D. et al. Deep learning for text style transfer: A survey. *Computational Linguistics*, MIT Press, Cambridge, MA, v. 48, n. 1, p. 155–205, mar. 2022. Disponível em: <https://aclanthology.org/2022.cl-1.6>. Citado na página 49.

JURAFSKY, D.; MARTIN, J. H. *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*. Upper Saddle River, N.J.: Pearson Prentice Hall, 2009. ISBN 9780131873216 0131873210. Disponível em: <http://www.amazon.com/Speech-Language-Processing-2nd-Edition/dp/0131873210/ref=pd_bxgy_b_img_y>. Citado na página 52.

LEWIS, M. et al. *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. 2019. Cite arxiv:1910.13461. Disponível em: <http://arxiv.org/abs/1910.13461>. Citado na página 30.

LIN, C.-Y. ROUGE: A package for automatic evaluation of summaries. In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, 2004. p. 74–81. Disponível em: <https://www.aclweb.org/anthology/W04-1013>. Citado na página 18.

MANNING, C.; RAGHAVAN, P.; SCHüTZE, H. *Introduction to Information Retrieval*. [S.l.]: Cambridge University Press, 2008. 25–26 p. Citado na página 19.

OESPER, L. et al. Wordcloud: a cytoscape plugin to create a visual semantic summary of networks. *Source code for biology and medicine*, Springer, v. 6, n. 1, p. 7, 2011. Citado 2 vezes nas páginas 19 e 35.

PAPINENI, K. et al. Bleu: A Method for Automatic Evaluation of Machine Translation. In: *ACL '02*. Morristown, NJ, USA: ACL, 2001. p. 311–318. Citado na página 18.

PASZKE, A. et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. 2019. Cite arxiv:1912.01703Comment: 12 pages, 3 figures, NeurIPS 2019. Disponível em: <http://arxiv.org/abs/1912.01703>. Citado na página 36.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Citado na página 36.

PETSI. *Processamento de Linguagem Natural: Histórico e Potencial Atual*. 2019. Acessado em: 17/01/2023. Disponível em: <http://www.each.usp.br/petsi/jornal/?p=2577>. Citado na página 12.

RADFORD, A. et al. Language models are unsupervised multitask learners. 2018. Disponível em: <https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf>. Citado na página 30.

RAFFEL, C. et al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. 2019. Cite arxiv:1910.10683Comment: Final version as published in JMLR. Disponível em: <http://arxiv.org/abs/1910.10683>. Citado na página 30.

RUSSELL, S. J.; NORVIG, P. *Inteligência Artificial - Uma Abordagem Moderna*. [S.l.]: GEN LTC, 2010. Citado na página 15.

SOLVIMM. *O que é Inteligência Artificial*. 2021. <https://solvimm.com/blog/o-que-e-inteligencia-artificial/>. Acessado em [20/01/2023]. Citado na página 15.

SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. BERTimbau: pretrained BERT models for Brazilian Portuguese. In: *9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23 (to appear)*. [S.l.: s.n.], 2020. Citado na página 28.

TEAM, T. pandas development. *pandas-dev/pandas: Pandas*. Zenodo, 2020. Disponível em: <https://doi.org/10.5281/zenodo.3509134>. Citado na página 34.

VASWANI, A. et al. Attention is all you need. In: GUYON, I. et al. (Ed.). *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017. p. 5998–6008. Disponível em: <https://papers.nips.cc/paper/7181-attention-is-all-you-need>. Citado 3 vezes nas páginas 24, 52 e 53.

WASKOM, M. et al. *mwaskom/seaborn: v0.8.1 (September 2017)*. Zenodo, 2017. Disponível em: <https://doi.org/10.5281/zenodo.883859>. Citado na página 36.

YAHOO! *L6 - Yahoo! Answers Comprehensive Questions and Answers version 1.0*. 2007. Acessado em: 20/11/2022. Disponível em: <https://webscope.sandbox.yahoo.com/catalog.php?datatype=l>. Citado na página 38.

Apêndices

APÊNDICE A – Análise de Dados Pós Tratamento

Figura 22 – Caracteres por Frase Pós Tratamento Histograma

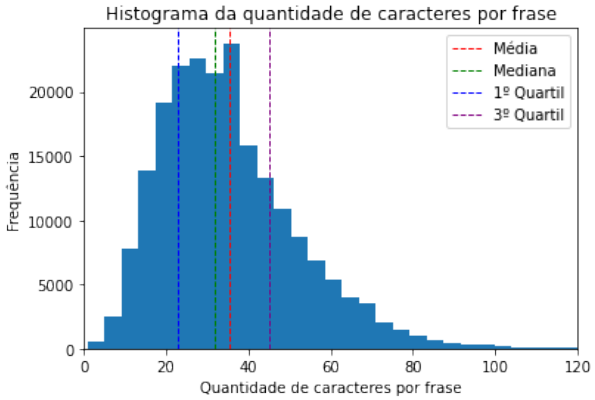
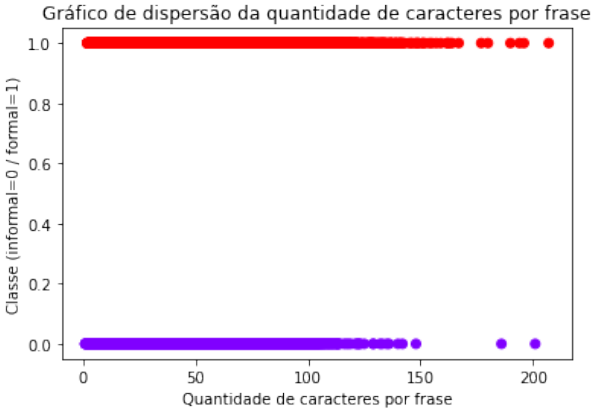


Figura 23 – Caracteres por Frase Pós Tratamento Dispersão



Apêndices

APÊNDICE B – Aplicar um modelo de aprendizado de máquina para transfor- mação de texto informal em formal.

Parâmetro	**Valor**
_name_or_path	./modelPTT5/final_model
architectures	T5ForConditionalGeneration
d_ff	3072
d_kv	64
d_model	768
decoder_start_token_id	0
dense_act_fn	relu
dropout_rate	0.1
eos_token_id	1
feed_forward_proj	relu
initializer_factor	1.0
is_encoder_decoder	true
is_gated_act	false
layer_norm_epsilon	1e-06
model_type	t5
n_positions	512
num_decoder_layers	12
num_heads	12
num_layers	12
output_past	true
pad_token_id	0
relative_attention_max_distance	128
relative_attention_num_buckets	32
torch_dtype	float32
transformers_version	4.30.2
use_cache	true
vocab_size	32128

Tabela 15 – Valores dos parâmetros da configuração do modelo PTT5

Parâmetro	**Descrição**
_name_or_path	Caminho do modelo
architectures	Arquiteturas do modelo
d_ff	Dimensão da rede feed forward
d_kv	Dimensão do vetor-chave no mecanismo de atenção
d_model	Dimensão do modelo
decoder_start_token_id	ID do token que inicia o decodificador
dense_act_fn	Função de ativação usada na camada densa
dropout_rate	Taxa de desativação usada no modelo
eos_token_id	ID do token que indica o final da sequência
feed_forward_proj	Projeção da rede feed forward
initializer_factor	Fator usado na inicialização do modelo
is_encoder_decoder	Indica se o modelo é do tipo codificador-decodificador
is_gated_act	Indica se a função de ativação é controlada por portão
layer_norm_epsilon	Pequeno número para evitar divisão por zero
model_type	Tipo do modelo (aqui, t5)
n_positions	Número de posições
num_decoder_layers	Número de camadas no decodificador
num_heads	Número de cabeças no mecanismo de atenção
num_layers	Número total de camadas
output_past	Indica se o modelo deve ou não retornar o passado
pad_token_id	ID do token de preenchimento
relative_attention_max_distance	Distância máxima considerada na atenção relativa
relative_attention_num_buckets	Número de baldes usados na atenção relativa
torch_dtype	Tipo de dados do torch usado
transformers_version	Versão do transformers utilizada
use_cache	Indica se o modelo usa ou não cache
vocab_size	Tamanho do vocabulário

Tabela 16 – Descrição dos parâmetros da configuração do modelo PTT5

Parametro	**Valor**
_name_or_path	./bartbase/final_model
activation_dropout	0.1
activation_function	gelu
add_bias_logits	false
add_final_layer_norm	false
architectures	BartForConditionalGeneration
attention_dropout	0.1
bos_token_id	0
classif_dropout	0.1
classifier_dropout	0.0
d_model	768
decoder_attention_heads	12
decoder_ffn_dim	3072
decoder_layerdrop	0.0
decoder_layers	6
decoder_start_token_id	2
dropout	0.1
early_stopping	true
encoder_attention_heads	12
encoder_ffn_dim	3072
encoder_layerdrop	0.0
encoder_layers	6
eos_token_id	2
forced_bos_token_id	0
forced_eos_token_id	2
gradient_checkpointing	false
id2label	{0: LABEL_0, 1: LABEL_1, 2: LABEL_2}
init_std	0.02
is_encoder_decoder	true
label2id	{LABEL_0: 0, LABEL_1: 1, LABEL_2: 2}
max_position_embeddings	1024
model_type	bart
no_repeat_ngram_size	3
normalize_before	false
normalize_embedding	true
num_beams	4
num_hidden_layers	6
pad_token_id	1
scale_embedding	false
torch_dtype	float32
transformers_version	4.30.2
use_cache	true
vocab_size	50265

Tabela 17 – Configurações dos parametros do modelo BART

Parâmetro	**Descrição**
activation_dropout	Taxa de desativação usada em todas as camadas, exceto a última
activation_function	Nome da função de ativação a ser usada (aqui, gelu)
add_bias_logits	Se adiciona ou não um viés aos logits
add_final_layer_norm	Se adiciona ou não a normalização da camada final
architectures	Arquiteturas do modelo
attention_dropout	Taxa de desativação usada na camada de atenção
bos_token_id	ID do token indicando o início da sequência
classif_dropout	Taxa de desativação usada na camada de classificação
classifier_dropout	Taxa de desativação usada na camada do classificador
d_model	Dimensão do modelo
decoder_attention_heads	Número de cabeças de atenção no decodificador
decoder_ffn_dim	Dimensão da rede feed forward no decodificador
decoder_layerdrop	Taxa de desativação da camada no decodificador
decoder_layers	Número de camadas no decodificador
decoder_start_token_id	ID do token que indica o início do decodificador
dropout	Taxa de desativação usada em todo o modelo
early_stopping	Se o treinamento é interrompido precocemente ou não
encoder_attention_heads	Número de cabeças de atenção no codificador
encoder_ffn_dim	Dimensão da rede feed forward no codificador
encoder_layerdrop	Taxa de desativação da camada no codificador
encoder_layers	Número de camadas no codificador
eos_token_id	ID do token indicando o final da sequência
forced_bos_token_id	ID do token forçado no início da sequência
forced_eos_token_id	ID do token forçado no final da sequência
gradient_checkpointing	Se usa ou não a verificação de ponto do gradiente
max_position_embeddings	Máximo número de embeddings de posição
model_type	Tipo do modelo (aqui, bart)
no_repeat_ngram_size	Tamanho do n-grama que não pode ser repetido
normalize_before	Se a normalização é feita antes ou não
normalize_embedding	Se normaliza ou não o embedding
num_beams	Número de feixes usados no beam search
num_hidden_layers	Número de camadas ocultas
pad_token_id	ID do token de padding
scale_embedding	Se dimensiona ou não o embedding
torch_dtype	Tipo de dados do torch usado
use_cache	Se usa ou não cache
vocab_size	Tamanho do vocabulário

Tabela 18 – Descrição dos parâmetros da configuração do modelo BART

Parâmetro	**Valor**
_name_or_path	./modelT5-english/final_model
architectures	T5ForConditionalGeneration
d_ff	3072
d_kv	64
d_model	768
decoder_start_token_id	0
dense_act_fn	relu
dropout_rate	0.1
eos_token_id	1
feed_forward_proj	relu
initializer_factor	1.0
is_encoder_decoder	true
is_gated_act	false
layer_norm_epsilon	1e-06
model_type	t5
n_positions	512
num_decoder_layers	12
num_heads	12
num_layers	12
output_past	true
pad_token_id	0
relative_attention_max_distance	128
relative_attention_num_buckets	32
task_specific_params	Ver nota abaixo
torch_dtype	float32
transformers_version	4.30.2
use_cache	true
vocab_size	32128

Tabela 19 – Valores dos parâmetros da configuração do modelo T5

Parâmetro	**Descrição**
_name_or_path	Caminho do modelo
architectures	Arquiteturas do modelo
d_ff	Dimensão da rede feed forward
d_kv	Dimensão do vetor-chave no mecanismo de atenção
d_model	Dimensão do modelo
decoder_start_token_id	ID do token que inicia o decodificador
dense_act_fn	Função de ativação usada na camada densa
dropout_rate	Taxa de desativação usada no modelo
eos_token_id	ID do token que indica o final da sequência
feed_forward_proj	Projeção da rede feed forward
initializer_factor	Fator usado na inicialização do modelo
is_encoder_decoder	Indica se o modelo é do tipo codificador-decodificador
is_gated_act	Indica se a função de ativação é controlada por portão
layer_norm_epsilon	Pequeno número para evitar divisão por zero
model_type	Tipo do modelo (aqui, t5)
n_positions	Número de posições
num_decoder_layers	Número de camadas no decodificador
num_heads	Número de cabeças no mecanismo de atenção
num_layers	Número total de camadas
output_past	Indica se o modelo deve ou não retornar o passado
pad_token_id	ID do token de preenchimento
relative_attention_max_distance	Distância máxima considerada na atenção relativa
relative_attention_num_buckets	Número de baldes usados na atenção relativa
task_specific_params	Parâmetros específicos de tarefas (ver tabela separada)
torch_dtype	Tipo de dados do torch usado
transformers_version	Versão do transformers utilizada
use_cache	Indica se o modelo usa ou não cache
vocab_size	Tamanho do vocabulário

Tabela 20 – Descrição dos parâmetros da configuração do modelo T5

Tarefa	**Parâmetros**
summarization	early_stopping: true, length_penalty: 2.0, max_length: 200, min_length: 30, no_repeat_ngram_size: 3, num_beams: 4, prefix: "summarize: "
translation_en_to_de	early_stopping: true, max_length: 300, num_beams: 4, prefix: "translate English to German: "
translation_en_to_fr	early_stopping: true, max_length: 300, num_beams: 4, prefix: "translate English to French: "
translation_en_to_ro	early_stopping: true, max_length: 300, num_beams: 4, prefix: "translate English to Romanian: "

Tabela 21 – Parâmetros específicos de tarefa para a configuração do modelo T5

Parâmetro	**Descrição em Português**
early_stopping	Indica se o processo de geração de sequências deve parar assim que for gerado um token de finalização.
length_penalty	Ponderação aplicada na decodificação beam search para sequências de diferentes comprimentos.
max_length	Comprimento máximo da sequência de saída que será gerada.
min_length	Comprimento mínimo da sequência de saída que será gerada.
no_repeat_ngram_size	O tamanho do n-grama que não deve ser repetido na geração de sequências.
num_beams	Número de feixes (beams) para usar na decodificação beam search.
prefix	O prefixo adicionado à entrada para fornecer uma dica ao modelo sobre a tarefa que deve ser realizada.

Tabela 22 – Descrição dos parâmetros da seção *task_specific_params* da configuração do modelo T5