

Identificação de autoria de contos usando técnicas de Processamento de Linguagem Natural

Igor Brito dos Santos Garcia, Pâmela Coca dos Santos Ramos, Henrique Dezani

e-mail: igorbrito1254@outlook.com; pamelacsramos@gmail.com;
henrique.dezani@fatec.sp.gov.br

Faculdade de Tecnologia de São José do Rio Preto

Resumo: Este artigo apresenta o estudo do processamento de linguagem natural para a identificação de autoria em contos de Márcia Denser e Sônia Coutinho. O trabalho de diferenciação de autoria por meio da NLP se faz relevante por fazer parte da área de *Digital Humanities* e por se tratar do estudo da NLP aplicada em textos literários e em português. Para a sua realização, foram digitalizados os contos das autoras e construído um dataset com os contos divididos em parágrafos. Após isso, os dados foram pré-processados e usados para a treino de um modelo rede neural convolucional – nesta etapa o treino foi realizado com 80% dos parágrafos. Após o treino, analisou-se a acurácia da identificação de autoria com o modelo apresentado e comparou-se estes resultados com a acurácia da identificação realizada por meio do ChatGPT, que também foi treinado com os parágrafos presentes no dataset na mesma proporção em que a rede foi treinada. Com base nessa comparação, pode-se concluir que, para a identificação de autoria com o dataset escolhido, a rede convolucional tem um melhor desempenho do que o ChatGPT.

Palavras-chave: Processamento de linguagem natural; classificação de texto; identificação de autoria.

Abstract: This paper aims the study of natural language processing for authorship identification in short stories of the Brazilian authors Márcia Denser and Sônia Coutinho. The authorship identification done through NLP techniques is relevant because is part of the Digital Humanities field and, in this paper, because it studies NLP applied to literary texts in Brazilian Portuguese. To use NLP in such texts, the published short stories of the authors were digitalized and divided into paragraphs, and a dataset was built with the result of this process. After this step, the data was preprocessed and used to train a convolutional neural network – using 80% of the paragraphs. Then, after the training, the accuracy of the identification was analyzed and its results were compared to the accuracy obtained through the identification done using ChatGPT, also trained using the same proportions used in the CNN training. Based on this comparison, this paper arrives at the conclusion that the CNN presents a better performs better than ChatGPT in the authorship identification of the dataset used for the tests.

Keywords: Natural Language Processing; text-classification; authorship identification.

1. INTRODUÇÃO

O processamento de linguagem natural é uma técnica que teve sua origem no século passado com propostas para tradutores mecânicos antes mesmo da invenção dos computadores de acordo com o artigo “A brief history of language processing” publicado pelo Peter Hancox, “A primeira aplicação reconhecível de NLP foi um dicionário de busca desenvolvido em Birkbeck College, em Londres, 1948” (HANCOX - tradução nossa)¹.

A princípio, o processamento de linguagem natural era feito com base em uma abordagem simbólica na qual eram escritos um conjunto de regras para a tradução dos símbolos. Com o passar dos anos e o desenvolvimento dos computadores, essa abordagem passou a ser substituída por uma abordagem estatística que se aproveitava de corpus de textos existentes em

¹ Trecho original: “The first recognisable NLP application was a dictionary look-up system developed at Birkbeck College, London in 1948”.

línguas diferentes. Um exemplo desta técnica é o artigo “*Aligning sentences in parallel corpora*” publicado em 1991 e que “descreve uma técnica estatística para o alinhamento de frases e sua tradução em um corpus de texto paralelo” (BROWN et al, 1990, 169 - tradução nossa)². Apenas em meados dos anos 2000 é que a abordagem estatística abriu espaço para uma das abordagens mais utilizadas no processamento de linguagem natural atualmente, que consiste no uso de Redes Neurais.

Além das mudanças de abordagem apontadas, o uso do processamento de linguagem natural também mudou ao longo do tempo. Ainda que sua origem tenha tido como foco a construção de dicionários e a tradução, hoje em dia o uso do processamento de linguagem natural é muito mais abrangente e ultrapassou a linguagem escrita. A definição da IBM é a seguinte:

O PLN combina linguística computacional, modelagem baseada em regras da linguagem humana, com modelos estatísticos, de machine learning e de deep learning. Juntas, essas tecnologias permitem que os computadores processem a linguagem humana na forma de dados de texto ou de voz e 'entendam' seu significado integral, completo com a intenção e o sentimento do locutor ou do escritor (IBM).

A linguística computacional, por sua vez, é um campo de estudo interdisciplinar que combina os estudos sobre a linguagem humana com os estudos sobre modelagem computacional. Apesar de ele ser necessário para o processamento de linguagem natural, ele não é o único campo que trabalha com a interdisciplinaridade entre estudos computacionais e estudos da área de humanidades.

Existe uma área de estudos chamada *Digital Humanities* que engloba essa interdisciplinaridade e que se divide em diversos ramos, abordando várias disciplinas de humanidades além da linguística como, por exemplo, história, música e comunicação. Essa área de estudos, assim como o processamento de linguagem natural, não é recente. De acordo com Berry no artigo “*What are the digital humanities*”, a área

está na vanguarda da aplicação de tecnologias baseadas em computadores nas humanas. Inicialmente chamada de “computação humana” o campo cresceu tremendamente nos últimos 40 anos. Focava originalmente no desenvolvimento de ferramentas e na criação de arquivos e bases de dados para textos, obras de arte e outros materiais (BERRY, 2019 - tradução nossa).³

Mais recentemente, no entanto, o foco dos estudos de *Digital Humanities* se expandiu e passou a abranger estudos como o processamento de linguagem natural ao englobar estudos mais críticos. Ainda de acordo com Berry, “o foco disciplinar se alargou para incluir estudos digitais críticos, bem como campos comumente associados com a engenharia tais como *machine learning*, ciência de dados e inteligência artificial” (BERRY, 2019 - tradução nossa).

Muitos progressos feitos recentemente e que ajudam no dia a dia podem se encaixar no campo de *Digital Humanities* tais como o desenvolvimento de assistentes virtuais como Siri e Alexa e os próprios tradutores online, que tanto avançaram em comparação com os tradutores

² Texto original: “describe a statistical technique for aligning sentences with their translations in two parallel corpora”.

³ Trecho original: Digital humanities are at the leading edge of applying computer-based technology in the humanities. Initially called ‘humanities computing’, the field has grown tremendously over the past 40 or so years. It originally focused on developing digital tools and the creation of archives and databases for texts, artworks, and other materials (BERRY, 2019).

mencionados anteriormente. Além do desenvolvimento dessas tecnologias que hoje estão imbricadas no cotidiano, parte do trabalho dos pesquisadores da área é desenvolver novos métodos computacionais para a análise de dados das áreas de humanas:

Como parte do seu trabalho, humanistas digitais desenvolveram novos métodos, como a análise estatística de base computacional, pesquisa e recuperação, modelagem de tópicos e visualização de dados. Eles aplicam essas técnicas a arquivos e coleções que são vastamente maiores do que qualquer pesquisador humano ou grupo de pesquisadores pode confortavelmente manusear (BERRY, 2019 - tradução nossa).⁴

É sob esta vertente que este trabalho se encaixa. Será apresentado, neste artigo, a construção de um classificador de textos que usa do processamento de linguagem natural para identificar a autoria de contos literários das autoras Márcia Denser e Sônia Coutinho. Apesar de não ser um corpus muito vasto, ele é composto de doze coletâneas de contos, o que justifica o uso do processamento de linguagem natural para a construção de um classificador.

Além do desenvolvimento de novos métodos, o uso de ferramentas já existentes também faz parte do escopo dos estudos de *Digital Humanities*. Um exemplo de ferramenta é o ChatGPT, onde de acordo com Marcin no artigo “*ChatGPT Prompts and the Potential of AI-Powered Writing for Digital Humanities*”, tecnologia é uma nova fronteira para a escrita da Inteligência Artificial nas *Digital Humanities*, dessa forma, o sistema “*Generative Pre-trained Transformer*” (GPT), mais conhecido como “ChatGPT”, é uma das principais tecnologias que possuem o potencial para revolucionar as interações humano-máquina. Essa IA (Inteligência Artificial) generativa é uma combinação de aprendizado profundo e processamento de linguagem natural, para gerar uma escrita semelhante à humana.

Como indicado por Marcin em seu artigo, o uso dessa plataforma nas pesquisas digitais oferece inúmeras vantagens. Ele auxilia os pesquisadores na produção rápida de relatórios mais precisos e detalhados do que os que poderiam ser criados manualmente. Além disso, a ferramenta alimentada por IA (inteligência artificial) reduz significativamente o tempo necessário para analisar grandes conjuntos de dados, capacitando-a a gerar prontamente resumos e insights valiosos.

Conforme estudos realizados pela Equipe AContentfy em 2023 no artigo “*The Evolution of ChatGPT: From Past to Present*”, nos primeiros estágios de desenvolvimento do GPT, o modelo foi inicialmente treinado com um conjunto limitado de textos da internet. Já a versão 2, apresentou uma evolução ao modelo original, pois foi treinado com um conjunto de dados que ultrapassou várias ordens de magnitude o tamanho do conjunto de dados do modelo anterior. Um ano após foi lançado o GPT-3 e o GPT-3.5, com uma maior quantidade de dados incorporado em seu treinamento conforme os novos modelos foram lançados.

Hoje, os modelos de linguagem natural são viáveis graças a tecnologia criada pelo Google em 2017, o Transformer. Como indicado por Rick (2022) no artigo da NVIDIA “O que é um Modelo Transformer?”, um modelo Transformer é uma rede neural que adquire o contexto através da análise das relações em dados sequenciais, como as palavras em uma frase. De acordo com ele,

Os modelos Transformer aplicam um conjunto em evolução de técnicas

⁴ Trecho original: “As part of their work, digital humanists have developed new methods, such as computer-based statistical analysis, search and retrieval, topic modelling, and data visualisation. They apply these techniques to archives and collections that are vastly larger than any human researcher or research group can comfortably handle (BERRY, 2019)”

matemáticas, chamadas de atenção ou auto atenção, para detectar as maneiras sutis como até mesmo elementos de dados distantes em uma série influenciam e dependem uns dos outros. Descritos pela primeira vez em um artigo de 2017 do Google, os Transformers estão entre os modelos mais novos e potentes já inventados até hoje (RICK MERRITT).

Dessa forma, dentro do campo de *Digital Humanities*, podemos perceber que a literatura sobre Processamento de Linguagem Natural (NLP – sigla em inglês) contém múltiplos exemplos de análise de textos. Isso se dá, pois, a NLP é um ramo dos estudos sobre a Inteligência Artificial que busca trabalhar com textos produzidos em linguagem humana e torná-los compreensíveis para que os computadores possam tratá-los, ou seja, textos mais abstratos como contos, posts em redes sociais, falas transcritas, roteiros etc., sejam manipulados por meio de algoritmos desenvolvidos com esse propósito.

O próprio ChatGPT, que se encontra em alta no momento⁵, se utiliza de técnicas de NLP para processar os inputs feitos pelos usuários. Chatbots, de maneira geral, recorrem a NLP. O uso da NLP para a área literária, no entanto, é mais recente. Tendo isso em vista, este trabalho busca usar de técnicas de NLP em contos literários com o objetivo de identificar a autoria destes contos.

Este trabalho, mais especificamente, procura usar a NLP em conto de duas autoras específicas: Márcia Denser e Sônia Coutinho. A diferenciação de autoria por meio da NLP neste contexto se justifica na medida em que, por se tratar de autoras que trabalham os mesmos temas e que publicaram durante um mesmo período histórico, será possível avaliar o desempenho da(s) técnica(s) escolhidas para a análise dos textos definidos no corpus. Além disso, esse trabalho se justifica na medida em que enriquece a literatura sobre NLP em língua portuguesa, que ainda é pequeno em comparação a literatura de língua inglesa.

O objetivo geral deste trabalho é avaliar o uso de Processamento de Linguagem Natural para uma análise de contos literários que objetiva diferenciar quais contos pertencem à autora Márcia Denser e quais contos pertencem a autora Sônia Coutinho. Para isso, temos os seguintes objetivos específicos: a) levantar e digitalizar todos os contos presentes na coletânea de ambas as autoras; b) selecionar a maneira mais relevante de treinar a classificação dos textos; c) realizar o pré-processamento dos contos para a sua classificação; d) classificar a autoria usando o ChatGPT e e) analisar e comparar os resultados das classificações.

2. METODOLOGIA

Tendo em vista o objetivo de classificar a autoria os contos selecionados entre as duas autoras previamente mencionadas, foi utilizada uma abordagem exploratória por meio do uso de métodos de NLP já existentes e utilizados em outros contextos, selecionados a partir da similaridade dos problemas solucionados em outros estudos e da eficácia desses métodos no processamento de textos em linguagem natural.

Como a proposta de classificação dos dados se delimita entre as duas autoras escolhidas (Sônia Coutinho e Márcia Denser) e no gênero textual escolhido (contos literários), os dados a serem manipulados na pesquisa consistem em todos os contos publicados pelas autoras em coletâneas de contos disponíveis no mercado editorial. Dessa forma, os dados são compostos de todos os contos presentes nas seguintes coletâneas: *Nascimento de uma mulher*,

⁵ Um exemplo dessa alta pode ser lido na notícia “ChatGPT sets record for fastest-growing user base” publicada na revista Forbes no início deste ano.

Uma certa felicidade, Os venenos de Lucrecia, O último verão de Copacabana, Mil olhos de uma rosa, Ovelha negra e amiga loura, Toda a verdade sobre a tia de Lúcia, de Sônia Coutinho e *Tango fantasma, Animal dos motéis, Diana Caçadora, Toda prosa e Toda prosa II*, de Márcia Denser.

Todos os livros selecionados foram adquiridos em forma física, uma vez foram publicados em diferentes décadas e não possuem publicações digitais. Dessa forma, todos os contos presentes nos livros em questão (um total de 112 contos), foram digitalizados por meio de ferramentas de reconhecimento ótico de caracteres (OCR – sigla em inglês) e revisados e corrigidos manualmente para adequação e conformidade dos textos gerados por meio da ferramenta ao texto impresso. Após isso, os textos coletados foram conferidos para remoção dos contos publicados em mais de uma obra e divididos por autoria e compilados em dois arquivos de texto individuais para a construção dos dataset de cada uma das autoras.

Como foram selecionados apenas os contos de ambas as autoras e removidos os contos que foram publicados de maneira duplicada, a quantidade de contos únicos coletados de cada autora foram 78 de Sônia Coutinho e 33 de Márcia Denser. Essa quantidade, no entanto, não é suficiente para o treinamento e o teste do classificador. Dessa forma, optou-se pela divisão dos contos em parágrafos como unidade mínima de sentido nesse tipo de texto.

Como as autoras têm estilos diferentes e publicaram quantidades diferentes de contos, a quantidade de parágrafos de cada uma também foi diferente. Sendo assim, foi selecionado como corpus para o classificador o número máximo de parágrafos da escritora cujo *dataset* continha menos parágrafos, o que totalizou 1614 parágrafos para cada escritora. Após essa divisão, foi realizado o tratamento dos dados para transformá-los em algo compreensível para a máquina. Isso foi feito por meio da tokenização (transformação das palavras presentes em cada parágrafo para tokens numéricos) e da normalização dos dados.

No que se diz a respeito ao uso do ChatGPT, pode-se usar dois métodos para treinar modelos de contos literários. O primeiro método envolve a utilização da plataforma web do ChatGPT, que oferece uma abordagem mais interativa e acessível. Através dessa plataforma, é possível treinar o modelo de forma intuitiva utilizando o prompt, fornecendo exemplos de textos literários e orientando-o a aprender o estilo do autor ou o contexto desejado.

A segunda opção é utilizar a API do ChatGPT, que requer a instalação das bibliotecas correspondentes e a configuração do ambiente de desenvolvimento. Com a API, tem-se maior controle sobre o processo de treinamento, incluindo a capacidade de ajustar hiper parâmetros e personalizar o modelo de acordo com as necessidades específicas do projeto, usando até mesmo “fine-tuning e Embeddings. Para a realização do processo de treinamento e classificação dos textos, o ideal é a utilização de um modelo na API que possua a função de fine-tuning, pois possuem melhora de aprendizado treinando com muitos mais exemplos do que os que cabem em apenas um prompt. Porém, devido ao alto custo para o uso desses modelos, optamos por realizar a análise e classificação utilizando ChatGPT presente na web.

No que diz respeito às métricas, na rede neural proposta neste trabalho, será utilizada a acurácia, que fornece uma medida da capacidade da rede neural em distinguir as diferentes autoras literárias analisadas, expressa em forma de porcentagem. A mesma métrica será utilizada na análise da identificação de autoria realizada pelo ChatGPT, a saber: $Acurácia = \frac{\text{Predições Corretas}}{\text{Total de predições}}$.

Para a realização desse trabalho foi escolhida a linguagem Python por ser fortemente usada em tarefas de processamento de linguagem natural. Como framework, foi selecionado o Pytorch, um pacote de Python que fornece computação de tensores como o NumPy e redes neurais profundas. E, como ferramenta para facilitar a programação, foi escolhido o Jupyter Notebook, um notebook computacional que serve para facilitar a codificação e a documentação uma vez que fornece a possibilidade de armazenar códigos, descrições, imagens,

gráficos e outras coisas em um único arquivo. Para a tokenização, foi utilizada a NLTK, uma das plataformas mais usadas e conhecidas para o processamento de linguagem natural com python, uma vez que fornece várias interfaces, métodos e até mesmo corpus prontos.

Para a realização desse trabalho é necessário apenas alguma ferramenta que se conecte com a internet, uma vez que existem opções como o Google Collab que podem ser acessados via navegador. No entanto, optou-se por desenvolver o classificador usando as máquinas pessoais por meio do VSCode.

3. REVISÃO DA LITERATURA

O processamento de linguagem natural diz respeito a técnicas computacionais usadas para o estudo da linguagem usada na comunicação humana cotidiana, como por exemplo inglês, português, espanhol etc. De acordo com Steven Bird, Ewan Klein e Edward Loper no livro *Natural Language Processing with Python*, o processamento de linguagem natural pode ser entendido, em um sentido amplo, como

qualquer tipo de manipulação computacional de linguagem natural. Em um extremo, pode ser tão simples quanto contar a frequência de palavras para comparar estilos de escrita diferentes. No outro extremo, a NLP envolve o “entendimento” completo de enunciados humanos, pelo menos no que diz respeito a dar respostas uteis a esses enunciados (BIRD et al. 2009, p. ix - tradução nossa).⁶

De acordo com o livro *Speech and Language Processing* de Daniel Jurafsky e de James Martin (2023), algumas das aplicações mais conhecidas da NLP são a tradução automática feita por máquina, resposta a perguntas e recuperação de informação, chatbots e sistemas de diálogo e reconhecimento automático da fala e texto para fala.

Uma das ferramentas usadas para essas aplicações mais conhecidas é a Classificação de texto. De acordo com Jacob Eisenstein em "*Natural Language Processing*" (2018, p. 05 - tradução nossa)⁷ "O processamento de linguagem natural tem um papel significativo em campos interdisciplinares emergentes como ciência social computacional e em *digital humanities*. Classificação de Texto, Clusterização e extração de informação são ferramentas particularmente uteis".

Ainda de acordo com o autor, a classificação de texto consiste na atribuição de rótulos a documentos de texto e tem muitas aplicações, desde coisas cotidianas como filtragem de spam até a análise de prontuários eletrônicos. Além disso, a classificação de texto é base para tarefas de processamento de linguagem natural mais complexas (p. 13). De acordo com Jurafsky e Martin, “tarefas relacionadas a classificação de texto como atribuição de autoria – determinando o autor de um texto - são relevantes para *Digital Humanities*, Ciências Sociais e Linguística

⁶ Trecho original: "any kind of computer manipulation of natural language. At one extreme, it could be as simple as counting word frequencies to compare different writing styles. At the other extreme, NLP involves “understanding” complete human utterances, at least to the extent of being able to give useful responses to them” (BIRD et al., 2009, p. ix).

⁷ Trecho original: “Natural language processing plays a significant role in emerging interdisciplinary fields like computational social science and the digital humanities. Text classification (chapter 4), clustering (chapter 5), and information extraction (chapter 17) are particularly useful tools;” (EISENSTEIN, 2018, p. 05).

Forense” (JURAFSKY, MARTIN, 2023, P. 58 - tradução nossa)⁸.

A classificação de texto pode ser executada de diversas formas. Ela pode usar da técnica *Bag of Words*, *Naive Bayes*, Aprendizado discriminativo, Regressão Logística e Redes Neurais, por exemplo. De acordo com Einsenstein, o uso de redes neurais é “a abordagem predominante para a classificação não linear no processamento de linguagem natural hoje” (EINSENSTEIN, 2018, p. 48 - tradução nossa)⁹. Isso se dá devido a sua eficiência diante das outras técnicas.

Existem vários tipos de redes neurais que são utilizadas para a classificação de texto como, por exemplo, as redes neurais Feed-Forward, Recorrente, Perceptron e Convolutacional. Para este trabalho, optamos pelo uso da rede Convolutacional em decorrência de sua aplicação em trabalhos similares. Além disso, de acordo com Einsenstein, o uso de redes convolucionais para a classificação de texto supre desafios enfrentados por outros modelos de classificação:

Uma fraqueza básica do modelo bag-of-words é a incapacidade de lidar com as maneiras com que as palavras se combinam para gerar significado [...]. Pesquisas anteriores em NLP tentaram capturar fenômenos linguísticos multipalavras com padrões lexicais desenhados a mão. [...] a saída dos filtros e padrões poderia funcionar características básicas em um classificador linear. Mas ao invés de desenhar esses extratores de característica a mão, uma melhor abordagem é aprendê-los, usando a mágica de backpropagation. Essa é a ideia por trás das redes neurais convolucionais (EINSENSTEIN, 2018, p. 62 - tradução nossa)¹⁰.

Dessa forma, apesar de ter sido originalmente utilizada com foco no processamento de imagens, as redes neurais convolucionais têm sido utilizadas para a classificação de texto e funcionam, de acordo com Cezanne Camacho em “*CNNs for text classification*” da seguinte forma: em um primeiro momento, é necessário realizar a tokenização das palavras e realizar seu tratamento de forma que elas se transformem em vetores, uma vez que as camadas convolucionais trabalham de maneira espacial. Os kernels convolucionais trabalharão em cima desses vetores e capturará características e padrões nas sequências de palavras. É por meio desse reconhecimento de padrões que a classificação será realizada.

Um dos trabalhos mais citados sobre o uso de CNN para a classificação de texto é o artigo de Yoon Kim, “*Convolutional Neural Networks for Sentence Classification*”. Neste artigo o autor mostra os resultados de uma série de experimentos realizados com redes convolucionais e sua comparação com outros métodos, chegando à conclusão de que mesmo com pouco refinamento de configuração de parâmetros, uma rede convolutacional simples com uma camada performa muito bem em comparação com os outros 17 modelos estudados em sete bases de dados diferentes.

Outro artigo que aborda a classificação de textos é “*SLCNN: Sentence-Level*

⁸ Trecho original: “Related text classification tasks like authorship attribution— determining a text’s author— are also relevant to the digital authorship attribution humanities, social sciences, and forensic linguistics” (JURAFSKY, MARTIN, 2023, p. 58).

⁹ Trecho original: “neural networks, which are the dominant approach for nonlinear classification in natural language processing today” (EINSENSTEIN, 2018, p. 48).

¹⁰ Trecho original: “A basic weakness of the bag-of-words model is its inability to account for the ways in which words combine to create meaning, [...] earlier NLP research attempted to capture multiword linguistic phenomena by hand-designed lexical patterns [...] the output of the filters and patterns could then act as base features in a linear classifier. But rather than designing these feature extractors by hand, a better approach is to learn them, using the magic of backpropagation. This is the idea behind convolutional neural networks” (EINSENSTEIN, 2018, p. 62).

Convolutional Neural Network for Text Classification". Neste trabalho, os autores Ali Jarrahi, Leila Safari e Ramin Mousa estudam modelos de rede convolucionais voltadas ao nível da sentença em que a rede neural é alimentada com tensores tridimensionais para extração de características posicionais na sentença. Além disso, os autores fizeram a comparação desses modelos com datasets diferentes (seis no total) - chegando à conclusão de que a maior diferença se dá na classificação de datasets que têm documentos longos.

Também a nível de sentença, podemos citar o artigo "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification". Neste estudo, os autores Ye Zhang e Byron C. Wallace fazem uma análise sobre a performance das redes neurais convolucionais a partir da arquitetura adotada, e fornecendo, como resultado de sua pesquisa, descobertas empíricas e um guia para pesquisadores que possam fazer o uso de CNNs para tarefas de classificação de sentença.

Mais voltado a identificação de autoria, podemos citar o artigo "*Code authorship identification using convolutional neural networks*", de Abuhamed et al, no qual os autores propõem três sistemas baseados em redes neurais convolucionais para a identificação de autoria em códigos open source chegando a resultados de acurácia que vão de 95% a 99,4% a depender da fonte do código e da linguagem de programação utilizada neles.

Outro exemplo, desta vez voltado a textos curtos, é o artigo de Shrestha et al, denominado "*Convolutional Neural Networks for Authorship Attribution of Short Texts*" no qual os autores apresentam um modelo que usa CNN para atribuição de autoria em Tweets utilizando, como ferramenta para o tratamento dos dados, N-grams sobre os caracteres dos textos. Realizando a comparação da abordagem adotada no trabalho com outras abordagens já conhecidas os autores concluem que o uso de CNN tem uma performance melhor do que as abordagens analisadas por eles.

No que se diz respeito ao GPT, de acordo com Allan Segovia Spadini no artigo "*O que é IA Generativa? A importância e o uso das Inteligências Artificiais como ChatGPT, MidJourney e outras*",

As inteligências artificiais generativas têm a capacidade de criar novas informações a partir de conjuntos de dados pré-existentes. Essas IAs são "ensinadas" a partir de grandes bases de dados com a intenção de que sejam capazes de adquirir o padrão de construção desses dados. Com essa compreensão adquirida, se tornam capazes de gerar novos dados, semelhantes aos dados utilizados para ensinar a IA, mas que podem ser únicos e originais. (SPADINI, 2023).

Dentre a capacidade de treinamento que possui o GPT, pode-se citar o modelo *Large Language Model* (LLMs), que consegue analisar maiores quantidades de dados e inferir a relação de palavras dentro do texto, e o *Long-Short-Term-Memory* (LSTM), que preenche a lacuna com a palavra estatisticamente mais provável, dado o contexto circundante. Segundo Molly Ruby em pesquisas realizadas no artigo "*How ChatGPT works the models behind the bot*", esses modelos de linguagem aumentaram na medida que foram se intensificando seus treinamentos e o tamanho do conjunto de dados para parâmetros de entrada aumentaram, este aprendizado acelerado unido ao aprendizado por reforço do feedback humano, tornou o ChatGPT em algo excepcional. Segundo a autora "O treinamento mais básico de modelos de linguagem envolve a previsão de uma palavra em uma sequência de palavras. Mais comumente, isso é observado como '*Next token prediction e masked language modeling*'" (Molly, 2023)¹¹.

¹¹ Trecho original: "The most basic training of language models involves predicting a word in a sequence of words. Most commonly, this is observed as either next-token-prediction and masked-language-modeling." (RUBY, 2023).

O ChatGPT trabalha com a previsibilidade e probabilidade. O *Next-token-prediction* (NTP) é um termo usado para descrever a tarefa de prever o próximo token. Por exemplo ao inserir a frase “Rafaela é a ...” ele começará a prever o elemento seguinte “irmã”, “líder”, “inteligente”, etc. Por outro lado, o *masked-language-modeling* (MLM), trabalha levando em consideração o contexto redor, como por exemplo “A Rafaela ... ler”, o token do meio agora a ser preenchido não será aleatório como (NTP), levará em consideração toda a sentença, trazendo exemplos como, “prefere”, “odeia”, “ama”, etc.

A arquitetura do GPT-3.5, baseada na arquitetura Transformer, publicada pela google o artigo “Attention is All You Need”, escrito por Ashish Vaswani et al., no ano de 2017, foi uma revolução nos modelos de IA e processamento de linguagem natural. Isso se dá em decorrência da maneira como os Transformers funcionam, permitindo a avaliação da relevância da relação de palavras no contexto dado. De acordo com Jurafsky e Martin:

Transformers são feitos a partir do empilhamento de blocos de Transformer, cada um deles sendo compostos de uma rede multicamadas feita a partir da combinação de camadas lineares de auto atenção, redes neurais de alimentação antecipada e camadas de auto atenção, inovação chave dos Transformers. A auto atenção permite que a rede extraia diretamente e use informações de contextos grandes. No centro de uma abordagem baseada em atenção está a habilidade de comparar um item de interesse em uma coleção de outros itens de uma maneira que revela a sua relevância no contexto atual. No caso da auto atenção, o conjunto de comparações é relacionado a outros elementos de uma determinada sequência (JURAFSKY, MARTIN, 2023, p. 220 - tradução nossa).¹²

Neste contexto, este estudo busca realizar também a classificação a autoria usando o ChatGPT. Um trabalho similar é o artigo “*Text classification challenge with extra small datasets fine tuning versus ChatGPT*”, escrito por Viacheslav Zhukov, em que foi realizado um experimento para avaliar a capacidade de compreensão de vários modelos. Neste trabalho, foi selecionado um conjunto de dados multiclasse composto por resumos de artigos científicos a fim de determinar o domínio de cada artigo. Após a realização dos testes chegou-se à conclusão de que modelos maiores treinados em dados mais extensos têm desempenho significativamente melhor em conjunto de dados extra pequenos.

4. DESENVOLVIMENTO

Neste tópico serão apresentadas as etapas que foram essenciais para o desenvolvimento deste trabalho, desde a preparação dos dados até a comparação dos resultados obtidos na identificação de autoria realizadas pela rede neural convolucional implementada e pelo ChatGPT.

¹² Trecho original: “Transformers are made up of stacks of transformer blocks, each of which is a multilayer network made by combining simple self-attention linear layers, feedforward networks, and self-attention layers, the key innovation of transformers. Self-attention allows a network to directly extract and use information from arbitrarily large contexts [...] At the core of an attention-based approach is the ability to compare an item of interest to a collection of other items in a way that reveals their relevance in the current context. In the case of self-attention, the set of comparisons are to other elements within a given sequence” (JURAFSKY, MARTIN, 2023, p. 220).

4.1 Preparação do dataset

O primeiro passo para o desenvolvimento deste trabalho foi a digitalização das obras para a construção do dataset com os contos das autoras Márcia Denser e Sônia Coutinho, que foram divididos em dois arquivos de texto separados, conforme explicitado na Metodologia deste trabalho. Em posse dos dois arquivos e diante da decisão de dividir os datasets das escritoras em parágrafos, foi desenvolvido um script responsável por fazer a divisão dos dois arquivos de texto com todos os contos em arquivos de texto individuais que contivessem apenas um parágrafo cada e por salvar esses arquivos em duas pastas, uma para cada escritora, com 1614 arquivos cada.

Para dividir o dataset entre treino e classificação, foi construída uma Classe em Python responsável por acessar os arquivos nas pastas das escritoras e devolver os datasets construídos a partir do embaralhamento dos parágrafos e da divisão deles a partir da proporção definida no parâmetro do método.¹³

Com a criação dessa classe, a construção dos datasets de treino e de teste foram feitas utilizando a proporção de 80% dos dados para treino e 20% dos dados para teste, como pode ser visto na Figura 1:

```
train_dataset_1, test_dataset_1 = ds_gen.get_dataset_from_author(dsg.DS_Gen.COUTINHO, 0.8, 1614)
train_dataset_2, test_dataset_2 = ds_gen.get_dataset_from_author(dsg.DS_Gen.DENSER, 0.8, 1614)
```

Figura 1 Construção do dataset

Como o treino e o teste são feitos usando apenas um dataset para cada um, os datasets de treino e de teste de cada uma das escritoras foram copiados para um terceiro dataset de treino e um terceiro dataset de testes. A partir do tamanho dos datasets de cada autora foram criados os vetores contendo os *labels* “0” para a autora Sônia Coutinho e “1” para a autora Márcia Denser.

Como os parágrafos são, por sua vez, de tamanhos diferentes e precisam ser normalizados antes de serem usados na Rede Neural, foi feito o percentil de 85 por cento do comprimento dos parágrafos, que resultou em 521. Na imagem abaixo (Figura 2) foram plotados os comprimentos dos parágrafos e a linha de percentil:

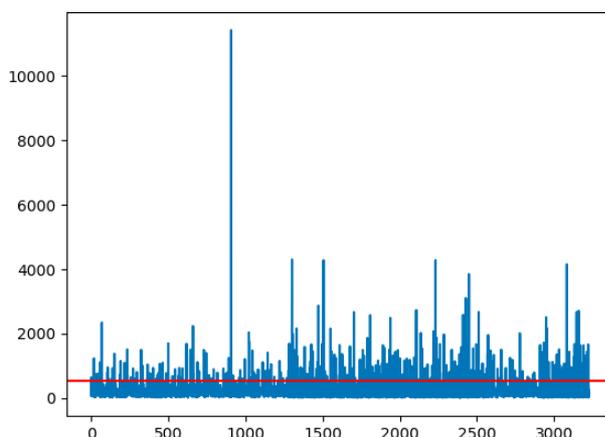


Figura 2 Comprimento e percentil dos parágrafos

¹³ Código aberto e disponível no GitHub:

<https://github.com/pamelacoca/authorClassification/blob/main/datasetgenerator.py>

A partir dessa informação, optamos por normalizar os parágrafos para que todos contivessem 520 tokens cada de maneira arbitrária. Para essa normalização, os parágrafos foram tokenizados com a biblioteca NLTK e com um dicionário construído a partir desses *tokens* e recortados quando maiores do que 520 palavras ou completados com tokens “0” quando menores, como pode-se ver na Figura 3:

```
paragrafo_original = "Seus parentes moram em outra cidade, só veio para o Rio, como ela, o irmão com quem se dá pior."
paragrafo_tokenizado = tokenize_words_in_paragraph(paragrafo_original, dictAuthors)
paragrafo_normalizado = normalize_data(paragrafo_tokenizado, 520)
print("Parágrafo original: ", paragrafo_original)
print("Tokenização normalizada: ", paragrafo_normalizado)
✓ 0.0s
Resultado da tokenização: [18361, 145, 863, 39, 1190, 179, 14, 625, 462, 89, 44, 18362, 14, 175, 328, 14, 44, 793, 57, 96, 27, 468, 1765, 46]
Parágrafo original: Seus parentes moram em outra cidade, só veio para o Rio, como ela, o irmão com quem se dá pior.
Tokenização normalizada: [18361, 145, 863, 39, 1190, 179, 14, 625, 462, 89, 44, 18362, 14, 175, 328, 14, 44, 793, 57, 96, 27, 468, 1765, 46, 0, 0, 0, 0, 0]
```

Figura 3 Exemplo de tokenização e normalização

4.2 Bibliotecas usadas e Construção da rede

Para a construção do trabalho e da rede neural convolucional foram importadas várias bibliotecas. Dentre elas, vale mencionar as bibliotecas próprias do Pytorch como a `torch.utils.data` que contém as classes `TensorDataset`, `DataLoader`, `RandomSampler`, `SequentialSampler` e `optim` para empacotar e iterar sobre os dados do Dataset. Foram utilizadas, também, as importações do módulo “json” para a manipulação do arquivo de dicionário construído, o módulo “copy” e “numpy” para manipulação dos vetores de treino e de teste, e as biblioteca “matplotlib” e “torchviz” para a geração de parte dos gráficos apresentados neste texto.

Como a rede neural convolucional precisa ser alimentada com uma matriz e a tokenização retorna apenas um vetor de apenas uma dimensão, fez-se necessário o uso de uma biblioteca responsável pelo preenchimento deste vetor. Para este trabalho, optou-se pelo uso da biblioteca “FastText” em detrimento de bibliotecas conhecidas como “Word2Vec” e “Glove” em decorrência do tamanho do corpus e do tamanho dicionário construído a partir da tokenização. Ao passo que “Glove” e “Word2Vec” consideram a palavra como unidade mínima para o treinamento, o “FastText” considera os caracteres a nível de N-gram, gerando melhores embeddings para palavras não repetidas ou não vistas durante o treinamento.

A arquitetura da rede convolucional construída neste trabalho foi concebida a partir do projeto implementado por Chris Khanhtran para análise e classificação de reviews de filmes publicado em seu blog com o título “*A Complete Guide to CNN for Sentence Classification with PyTorch*” e implementado a partir do estudo dos artigos de Yoon Kim e de Zhang e Wallace, mencionados no Referencial Teórico. Desta forma, a rede construída para a análise de autoria pode ser visualizada na Figura 4:

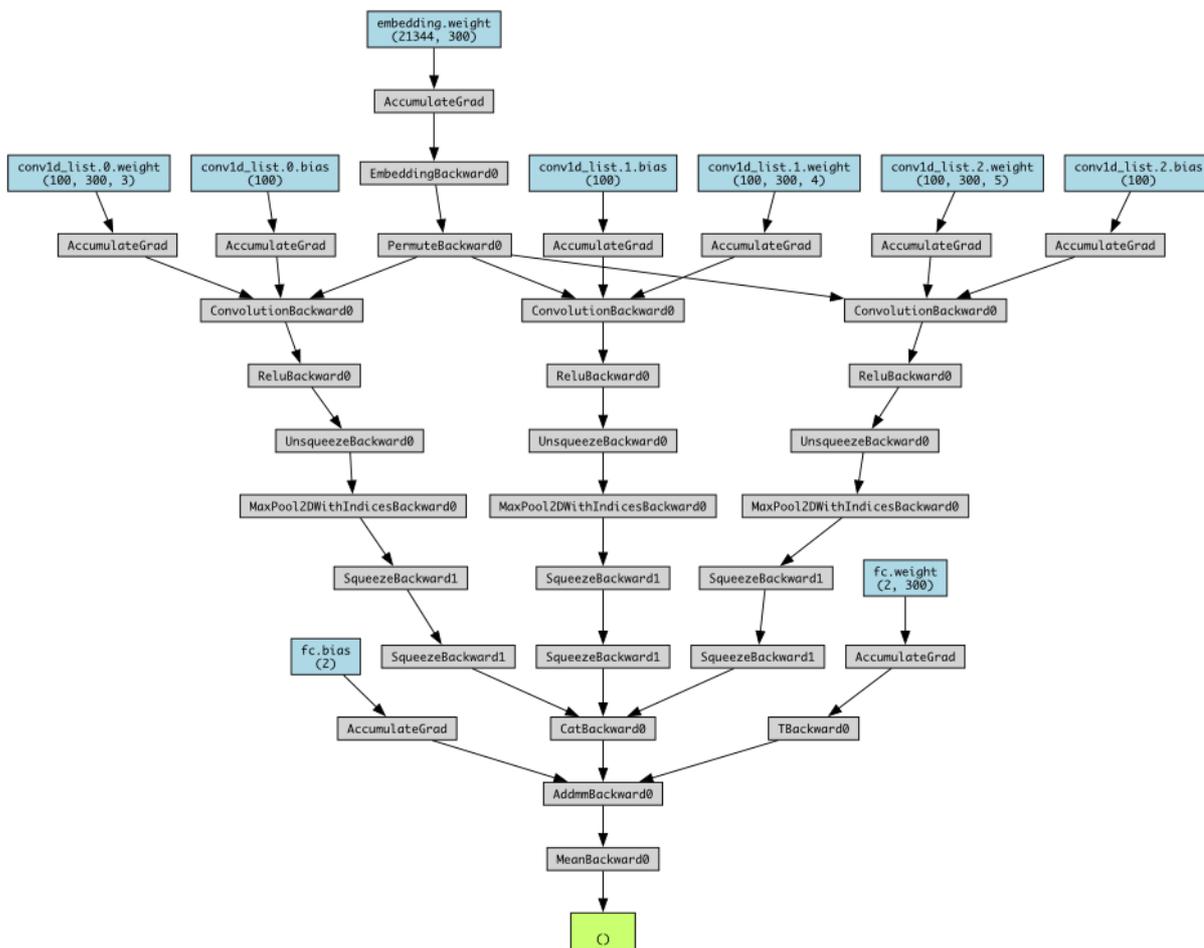


Figura 4 Visualização da rede convolucional

Como é possível observar pela imagem, a rede é construída da seguinte forma: a camada de embedding converte os vetores das sentenças em matrizes multidimensionais a partir do uso da biblioteca “FastText” com sua dimensão configurada em 300 (configuração padrão) e do Dicionário construído a partir da tokenização das palavras presentes no corpus, totalizando 21344 tokens. Os filtros e os tamanhos dos filtros das camadas convolucionais aplicadas, por sua vez, permaneceram com sua configuração padrão apresentadas no artigo de Zhang e Wallace: 100, 100, 100 e 3, 4, 5 respectivamente. Desta maneira, cada uma das camadas de convolução de uma dimensão receberá um input de dimensão 300 e devolverá um output filtrado em dimensão 100. Na função de ativação, por sua vez, foi utilizada a função ReLu e MaxPooling para a redução dos feature-maps para um tipo escalar único. Feito isso, é utilizada uma camada totalmente conectada para mapear as classes de saída desejadas (uma para cada autora a ser classificada). A Figura 5 explicita o modelo adotado na rede:

```

CNN_NLP(
  (embedding): Embedding(20803, 300)
  (conv1d_list): ModuleList(
    (0): Conv1d(300, 100, kernel_size=(3,), stride=(1,))
    (1): Conv1d(300, 100, kernel_size=(4,), stride=(1,))
    (2): Conv1d(300, 100, kernel_size=(5,), stride=(1,))
  )
  (fc): Linear(in_features=300, out_features=2, bias=True)
  (dropout): Dropout(p=0.5, inplace=False)
)

```

Figura 5 Modelo da rede convolucional

4.3 Treinamento da rede

Como a rede tem uma camada de self-embedding (ou seja, que aplica o embedding do FastText durante o treinamento), a execução do seu treinamento é mais longa do que outras redes. Por esse motivo, foi realizada uma comparação de: a) acurácia dos testes e b) perda nos testes sendo feito com 520 épocas (Figura 6) e 1040 épocas (Figura 7):

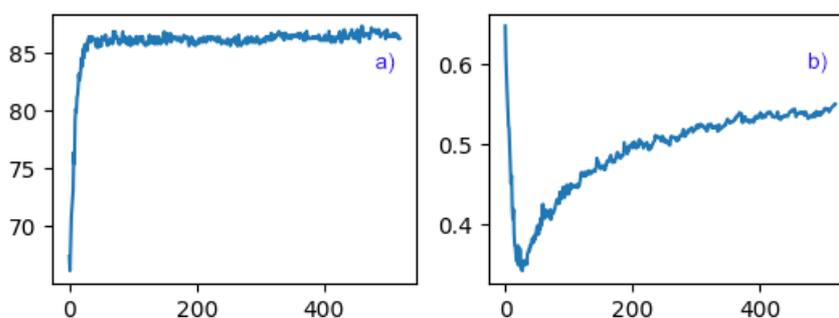


Figura 6 Execução do treino com 520 épocas, “a)” representa acurácia dos testes e “b)” perda nos testes

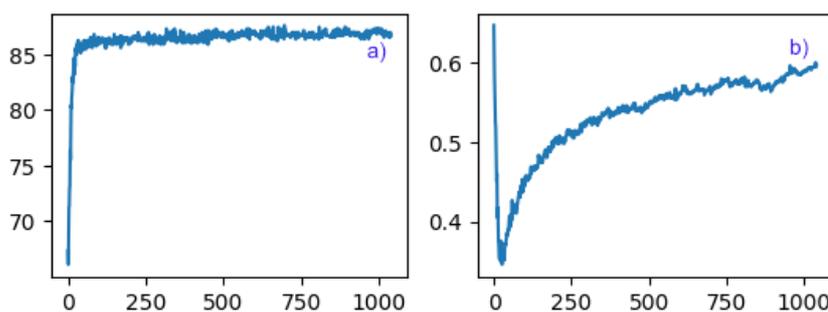


Figura 7 Execução do treino com 1040 épocas “a)” representa acurácia dos testes e “b)” perda nos testes

Como é possível perceber por meio das figuras acima, a acurácia se mantém constante logo após a execução das primeiras épocas e a perda nos testes, por sua vez, aumenta com um número maior de épocas. Além disso, a execução do treino com 520 épocas levou 5361.37 segundos (aproximadamente uma hora e meia). A execução do treino com 1040 épocas, por sua vez, levou 10657.99 segundos (aproximadamente, três horas). Desta maneira, optou-se pela execução dos testes com cinquenta épocas para a otimização do tempo, o resultado de sua acurácia e de sua perda são apresentados na Figura 8:

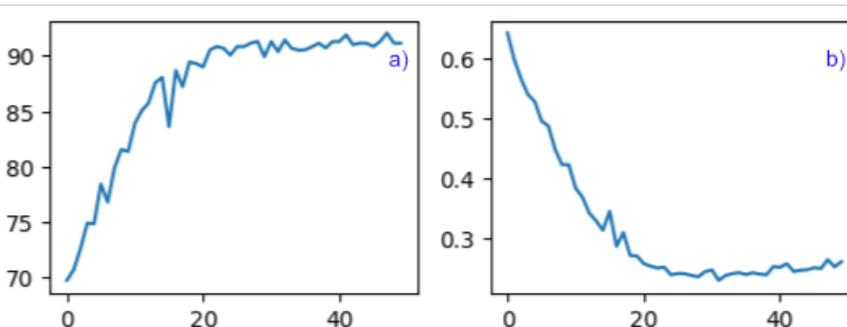


Figura 8 Execução do treino com 50 épocas, “a)” representa acurácia dos testes e “b)” perda nos testes

Esse treinamento levou 599.94 segundos, o que são cerca de 10 minutos para ser executado e teve, como melhor acurácia, 92.06 por cento. A Figura 9 apresenta as épocas executadas e suas respectivas acurácias e perdas e a Figura 10 apresenta a matriz de confusão gerada neste treinamento:

Start training...

Epoch	Train Loss	Test Loss	Test Acc	Elapsed
1	0.671620	0.643659	69.77	12.38
2	0.621332	0.599085	70.72	14.54
3	0.578723	0.566891	72.61	12.64
4	0.552195	0.541043	74.93	12.19
5	0.528609	0.529245	74.88	11.78
...
46	0.009867	0.251851	90.84	12.04
47	0.010368	0.250525	91.30	11.87
48	0.007790	0.265469	92.06	12.00
49	0.009103	0.253393	91.15	11.61
50	0.007892	0.261855	91.14	11.75

Training complete! Best accuracy: 92.06%.
 Training finished, took 599.94s

Figura 9 Execução do treino com 50 épocas

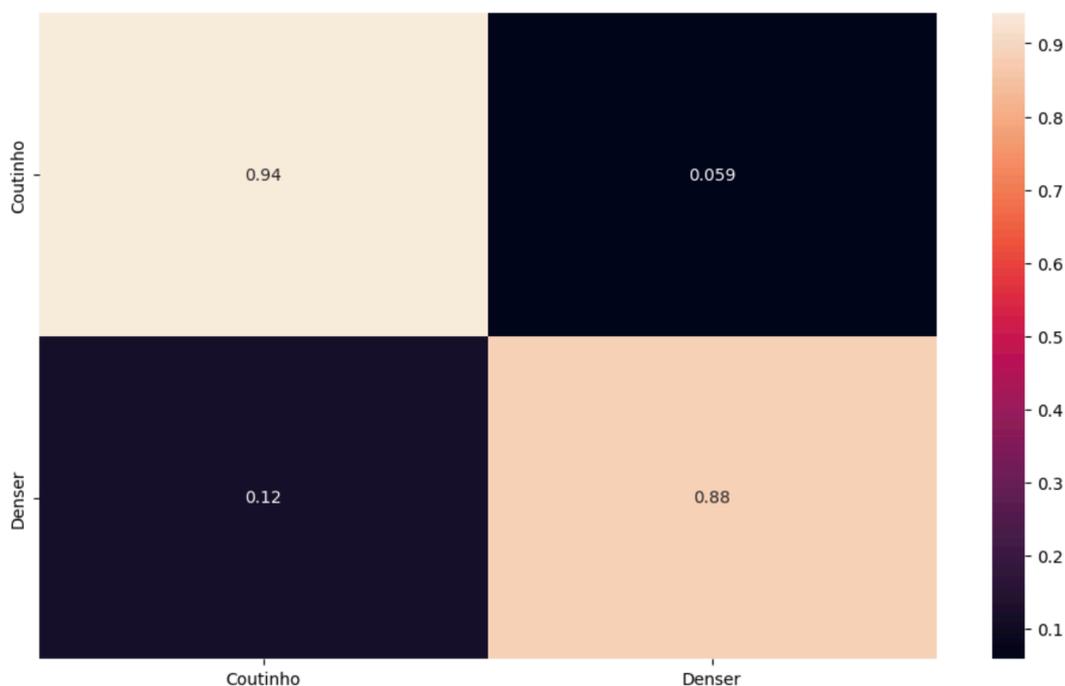


Figura 10 Matriz de confusão dos testes do treinamento

4.4 Testes com GPT

Inicialmente, foi realizado testes com o GPT em sua API, utilizando o GPT-3.5 através do modelo “text-davinci-003”, um modelo com suporte a até 4,097 tokens (unidades de texto, como palavras ou caracteres) e que pode realizar qualquer tarefa linguística com melhor qualidade, produção mais longa e seguimento consistente de instruções do que os modelos Curie, Babbage ou Ada (modelos de API do GPT). Esse modelo também oferece suporte a alguns recursos adicionais, como inserção de texto. Podemos visualizar melhor o modelo na Figura 11:

```
response1 = openai.Completion.create(  
    engine="text-davinci-003",  
    prompt=prompt2,  
    max_tokens=4096,  
    n=1,  
    stop=None,  
    temperature=0.7  
)
```

Figura 11 Exemplo do modelo GPT-3.5 “Text-davinci-003”

Porém, não foi possível utilizar a API para a realização da classificação de autoria pelo dataset ser muito extenso e pelo custo financeiro da utilização da API proposta pela OpenAI. Portanto, foi necessário utilizar o ChatGPT 3.5 para a execução e treinamento propostos neste trabalho. Abaixo está a tabela de preço para a utilização do modelo de “fine-tuning” text-davinci-003.

Modelo	Treinamento	Uso na entrada	Uso na saída
davinci-003	\$0.0060 / 1K tokens	\$0.0120 / 1K tokens	\$0.0120 / 1K tokens

Tabela 1 Tabela de precificação para utilizar o modelo “davinci-003”

Para utilizar o ChatGPT foi necessário utilizar a mesma Classe em Python para gerar um dataset reduzido de 400 parágrafos de cada autora e dividir este dataset entre treino e teste. Obteve-se, desta forma, um prompt de treino 620 parágrafos, sendo 320 parágrafos para a autora 1 (Sônia Coutinho) e 320 da autora 2 (Márcia Denser) e um prompt de teste de 160 (aproximadamente 5% do dataset completo), 80 da autora 1 e 80 da autora 2. Foi necessário gerar um dataset reduzido pois os parágrafos de treinamento e de teste precisam ser alimentados ao ChatGPT manualmente. A maneira como o prompt ficou organizado pode ser vista na Figura 12:

```
“Parágrafos de treino  
  
Parágrafos da autora 1: ..., ...,  
Parágrafos da autora 2: ..., ...,  
  
* Agora irei colocar os parágrafos de teste*  
  
Parágrafo de teste 1: ...  
Paragrafo de teste 2: ...”
```

Figura 12 Exemplo do esqueleto do prompt a ser enviado em partes

O ChatGPT, no entanto, limita a quantidade de caracteres que podem ser enviadas pelo usuário. Pensando neste limite, utilizou-se uma ferramenta para auxiliar o treinamento do Chat, chamada “*ChatGPT PROMPTs Splitter*”, que faz o corte do conteúdo a ser informado ao prompt em um formato de treino para o ChatGPT, como na Figura 13:

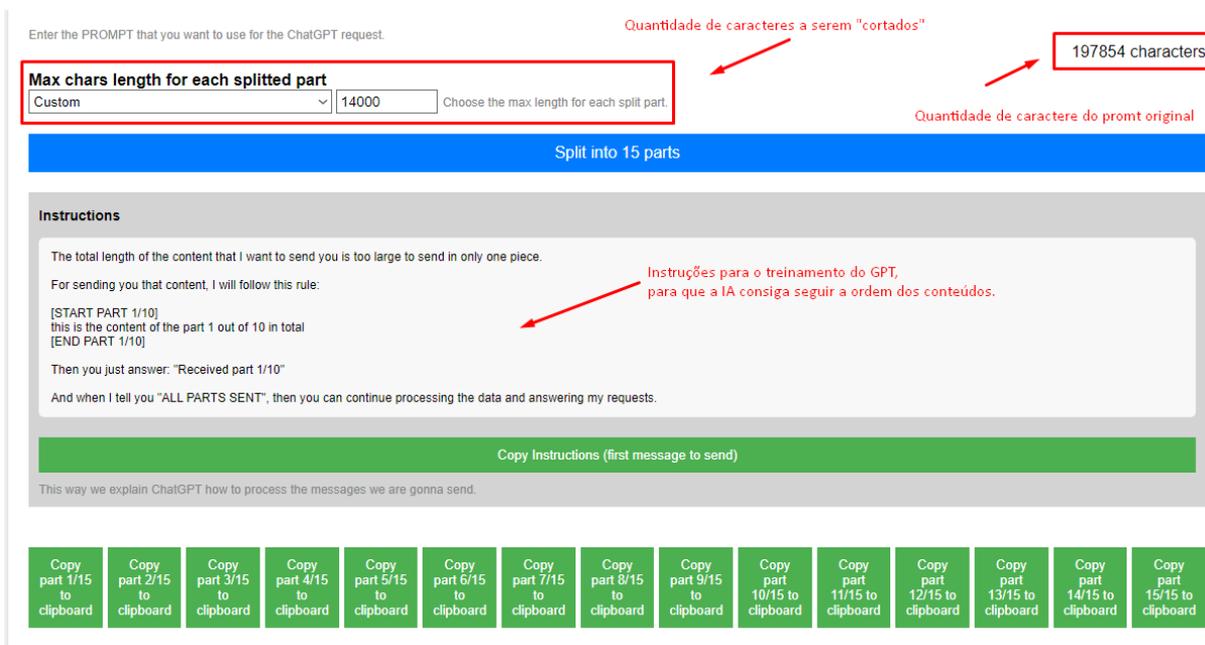


Figura 13 Ferramenta “ChatGPT PROMPT’s Splitter” cortando um prompt em 15 partes.

A ferramenta é bastante útil para treinamento, pois além de ajustar o prompt para que caiba dentro do GPT, ela integra a entrada de dados para que o ChatGPT consiga entender e analisar o conteúdo corretamente, como ilustrado demonstra na Figura 14.

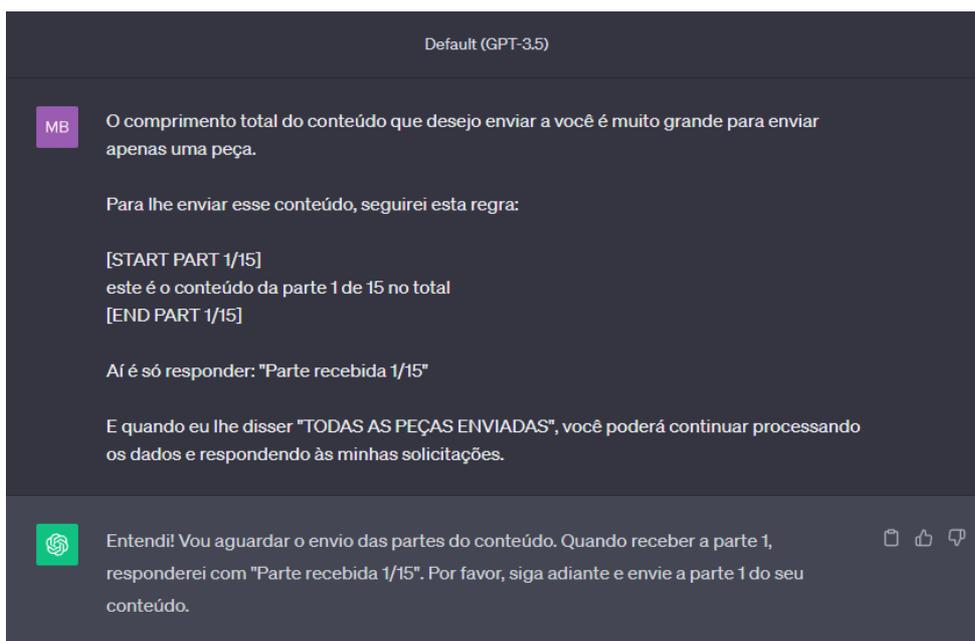


Figura 14 Instrução para o GPT entender que o conteúdo será enviado em partes

Para cada parte do texto, a ferramenta adiciona ao início a instrução ilustrada na Figura 15:

Não responda ainda. Esta é apenas mais uma parte do texto que quero lhe enviar. Basta receber e confirmar como "Parte 1/15 recebida" e aguardar a próxima parte. [...] Conteúdo [...] Lembre-se de não responder ainda. Apenas confirme que recebeu esta parte com a mensagem "Parte 1/15 recebida" e aguarde a próxima parte.

Figura 15 Instrução para o GPT entender esperar as próximas partes

É necessária essa integração de instrução com as partes “cortadas”, para que o chat entenda que não é para ele responder ainda e que é necessário analisar o conteúdo apenas quando todas as partes tenham sido enviadas. Dessa forma, quando tudo tiver sido enviado, a instrução “TODAS AS PEÇAS ENVIADAS. Agora você pode continuar processando a solicitação” é inserida ao prompt e o conteúdo previamente enviado pode ser analisado, como ilustrado na Figura 16:

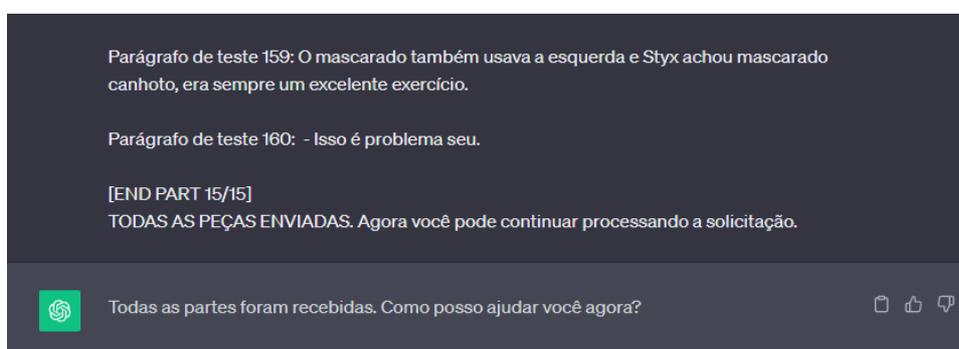


Figura 16 Exemplo de texto inserido no ChatGPT com trecho de confirmação de envio ao final do discurso.

Além do ajuste do dataset em partes menores para envio, houve, também, a necessidade de que parte dos parágrafos tivesse parte do conteúdo substituído por “****”. Isso foi necessário pois algumas partes de alguns contos das autoras usam palavras de baixo calão, contêm conteúdos eróticos, de teor violento ou que referem ao suicídio. Esses conteúdos ferem a política de uso do ChatGPT e, quando passados ao prompt, são exibidos um alerta e o treinamento da parte que contém este conteúdo não é realizado. Com os dados devidamente tratados, foi dado início ao processo de identificação das autoras 1 e 2 utilizando um prompt para que ele analisasse as partes enviadas anteriormente e para retornar à probabilidade determinado parágrafo de ser da autora identificada por ele (ChatGPT) O prompt utilizado para essa instrução pode ser visto na Figura 17.

“Acima temos alguns parágrafos para que sejam analisados, parágrafos da autora 1 e parágrafos de autora 2.

Abaixo deles, você tem também 160 parágrafos de teste, listados para que processe e analise cada um deles e classifique, levando em consideração o estilo de escrita das autoras 1 e 2.

Por exemplo coloque desta forma: o Parágrafo de teste 1 parecem corresponder à descrição da autora 1.

Use como parâmetro os parágrafos das autoras que eu te enviei acima, presentes em "Parágrafos da autora 1 " e "Parágrafos da autora 2".

Coloque na frente da sua classificação a probabilidade de ser a autora x. Coloque em porcentagem esta probabilidade”

Figura 17 Modelo do prompt utilizado

Inicialmente, utilizando o ChatGPT, foi feito um teste com os 160 parágrafos de teste em sequência, com os parágrafos de teste 1 até o 80 sendo da autora 1 e do parágrafo de teste 81 ao 160 eram da autora 2. O uso desta abordagem, no entanto, fez com que o ChatGPT realizasse a identificação correta de todas as autoras, conforme demonstrado na Figura 18:

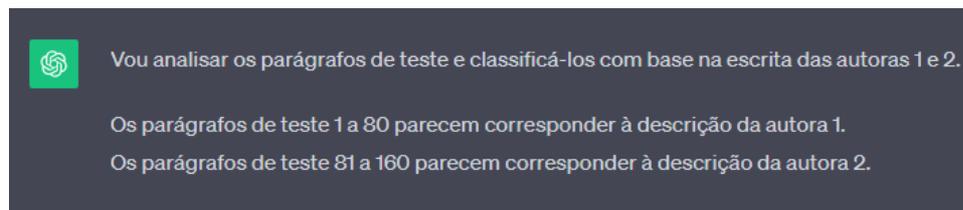


Figura 18 Análise com 100% de acerto do ChatGPT em relação aos 160 parágrafos de teste

Desta forma, foi necessário realizar um novo teste, desta vez com os parágrafos embaralhados para que o ChatGPT não obtivesse o padrão por meio da ordem em que os parágrafos foram passados. Neste teste, o ChatGPT não deu a mesma resposta e os resultados apresentados por ele foram catalogados em uma planilha do Excel para documentar as identificações de autoria realizada pelo ChatGPT e posteriormente comparar com o resultado da rede. A Figura 19 representa a resposta dessa identificação:

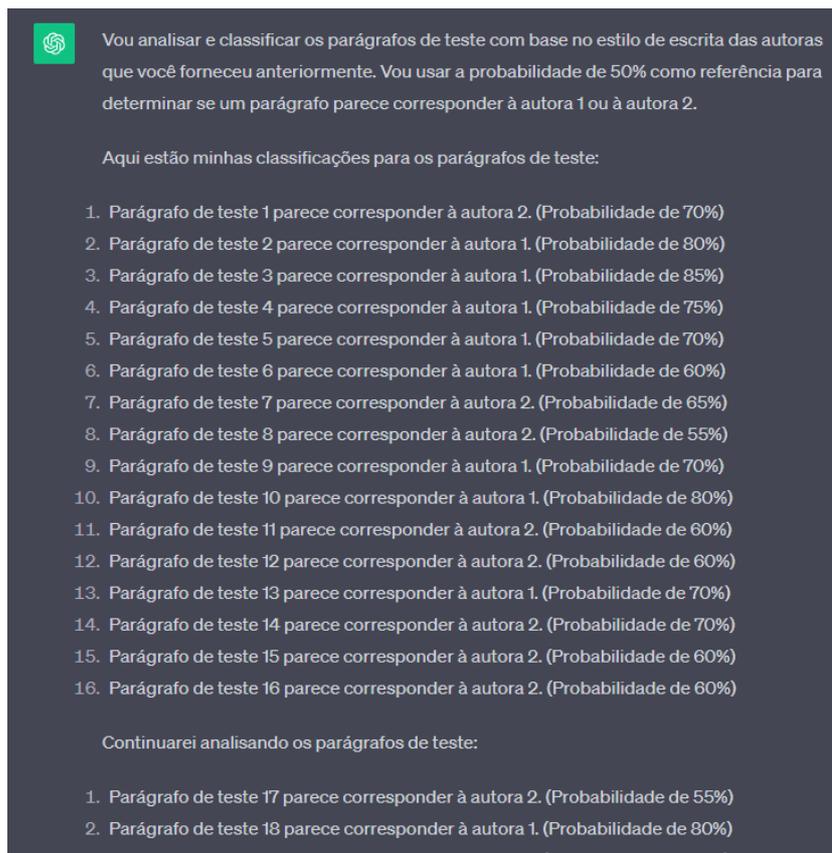


Figura 19 Resposta do ChatGPT na identificação de autoria

Após a resposta do ChatGPT, foi solicitada uma explicação sobre como foi realizada identificação de autoria e o cálculo da probabilidade apresentada para cada uma das identificações. A partir disso, obteve-se a resposta apresentada na Figura 19:

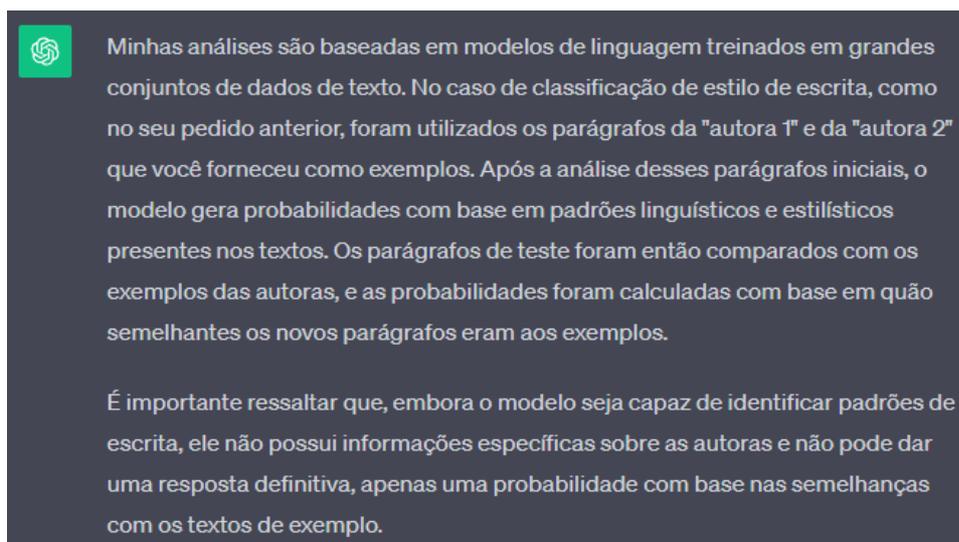


Figura 19 Explicação do ChatGPT na identificação de autoria

O resultado da classificação para a autora 1 (Sônia Coutinho) foi de 39 acertos e 41 erros. Já a autora 2 (Marcia Denser) foi de 30 acertos 50 erros. Um total de 69 acertos e 91 erros

de 160 parágrafos de teste. Nas Figuras 20 e 21 são apresentadas a relação de acertos e erros para cada autora e a matriz de confusão da identificação gerada a partir da identificação realizada pelo ChatGPT, respectivamente:

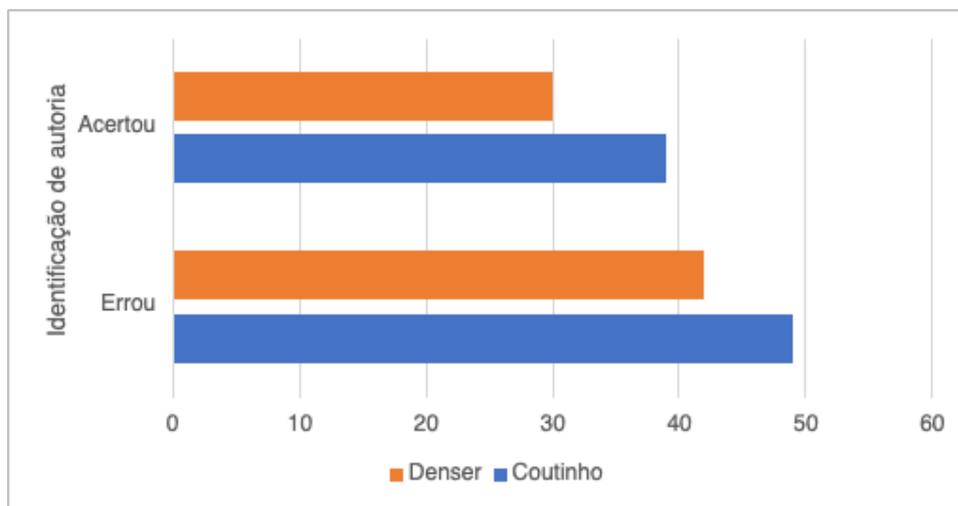


Figura 20 Identificação de autoria com o ChatGPT

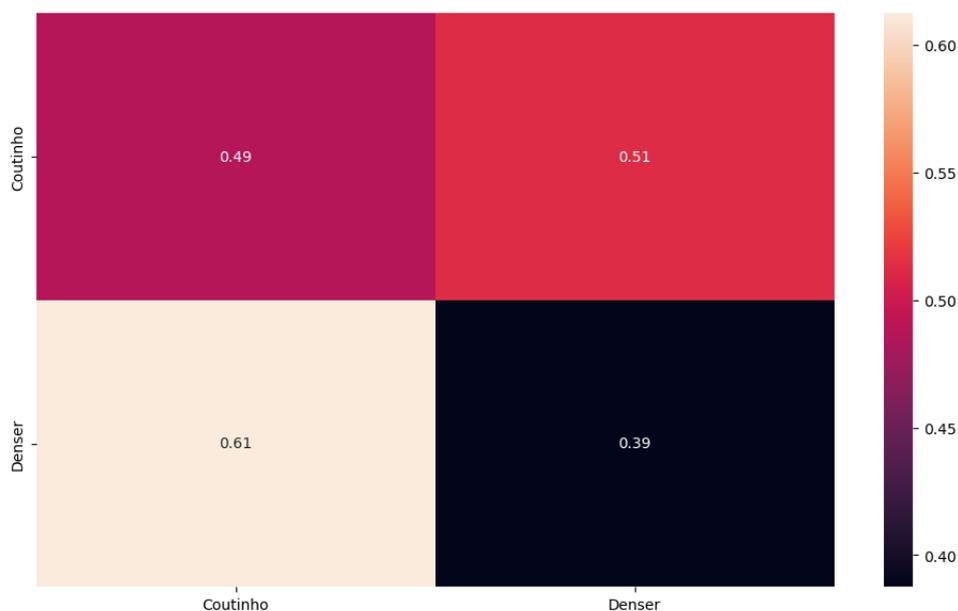


Figura 21 Matriz de confusão da identificação realizada pelo ChatGPT

4.5 Testes com Rede

Para realizar a comparação dos resultados da previsão de autoria da rede convolucional implementada, foi necessário padronizar os dados de teste com os dados de teste utilizados para avaliar o desempenho do ChatGPT. Dessa forma, os dados de teste foram removidos dos arquivos originais do dataset, que totalizaram 1534 parágrafos de cada autora após a remoção.

Além disso, os dados de treinamento que são gerados de maneira aleatória tiveram seus últimos 320 parágrafos de treino removidos para que fossem acrescentados os mesmos

parágrafos de treino que foram utilizados no treino do ChatGPT. Com essa adaptação realizada nos dados, o desempenho da rede caiu, uma vez que ela foi treinada com menos dados, resultando, dessa forma, em uma acurácia máxima de 88.07%, como pode ser visto na Figura 21 e com uma matriz de confusão como pode ser visto na Figura 22:

Start training...

Epoch	Train Loss	Test Loss	Test Acc	Elapsed
1	0.679278	0.648551	62.37	11.15
2	0.639948	0.588997	72.88	10.98
3	0.603390	0.550442	73.74	11.12
4	0.584271	0.559688	71.71	11.39
5	0.576214	0.524737	75.89	11.31
...
46	0.281735	0.301301	88.07	9.41
47	0.280779	0.304528	87.30	9.37
48	0.279378	0.299260	86.07	9.35
49	0.274928	0.351107	85.76	9.34
50	0.262527	0.296354	86.99	9.32

Training complete! Best accuracy: 88.07%.

Training finished, took 510.78s

Figura 21 Execução do treino com dataset reduzido

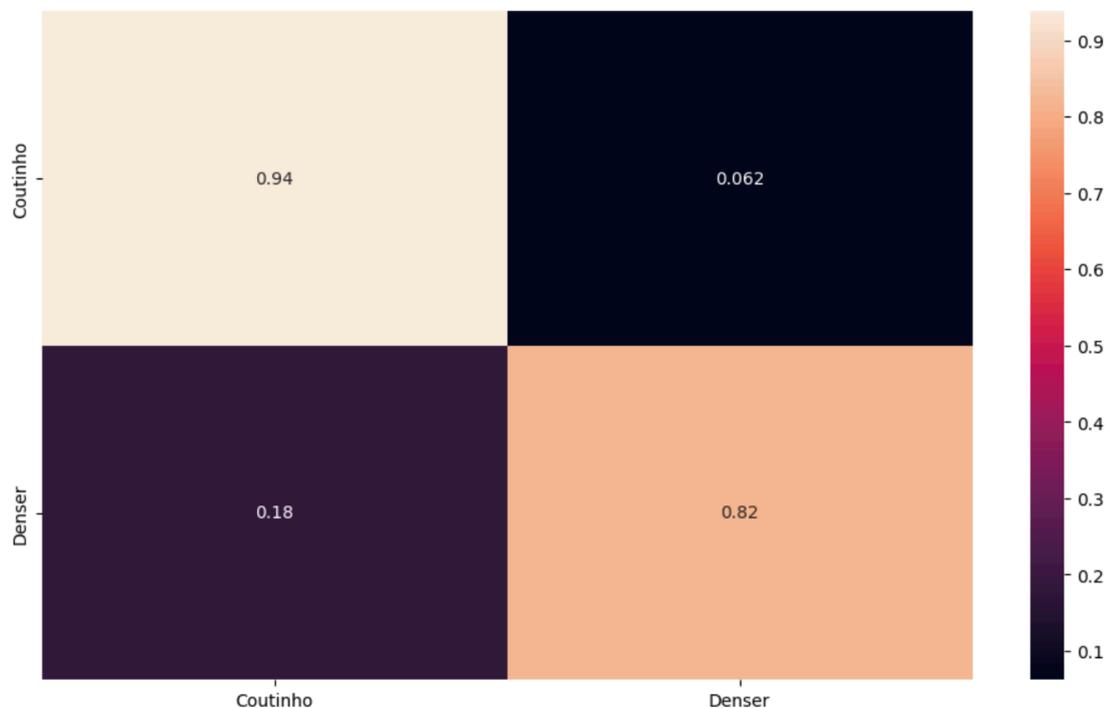


Figura 22 Matriz de confusão com dataset reduzido

A partir do treinamento realizado com o dataset modificado, os parágrafos usados para teste com o ChatGPT foram usados como argumento para o método “predict()”, que realiza a normalização do parágrafo e usa o modelo treinado para definir a probabilidade de este ser de uma autora ou outra, como pode ser visto no exemplo da Figura 23:

```
predict("Ou cruces.")
✓ 0.0s
[326, 13466, 22]
[326, 13466, 22, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Esse parágrafo tem 99.54% de chance de ser Denser.
```

Figura 23 Método “predict()” com um parágrafo da autora Márcia Denser como argumento

Dos 160 parágrafos usados para este teste, o modelo utilizado identificou corretamente a autoria de 113 parágrafos e errou 47, conforme demonstrado na Figura 24 e gerou a matriz de confusão demonstrada na Figura 25:

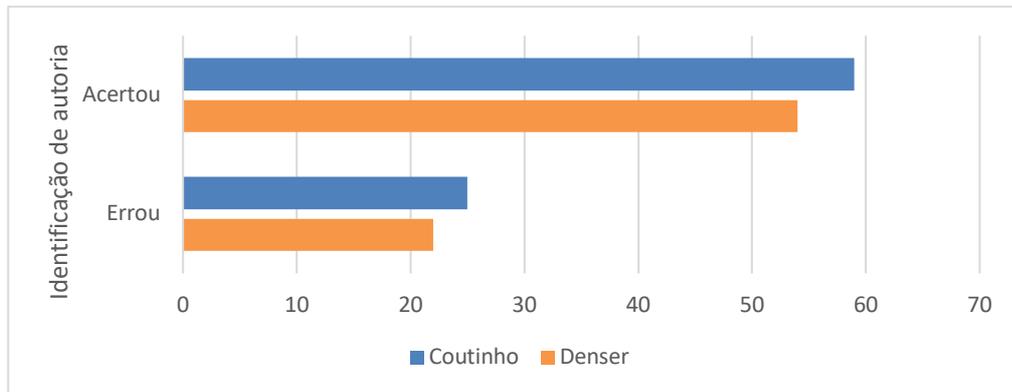


Figura 24 Identificação de autoria com o modelo da rede

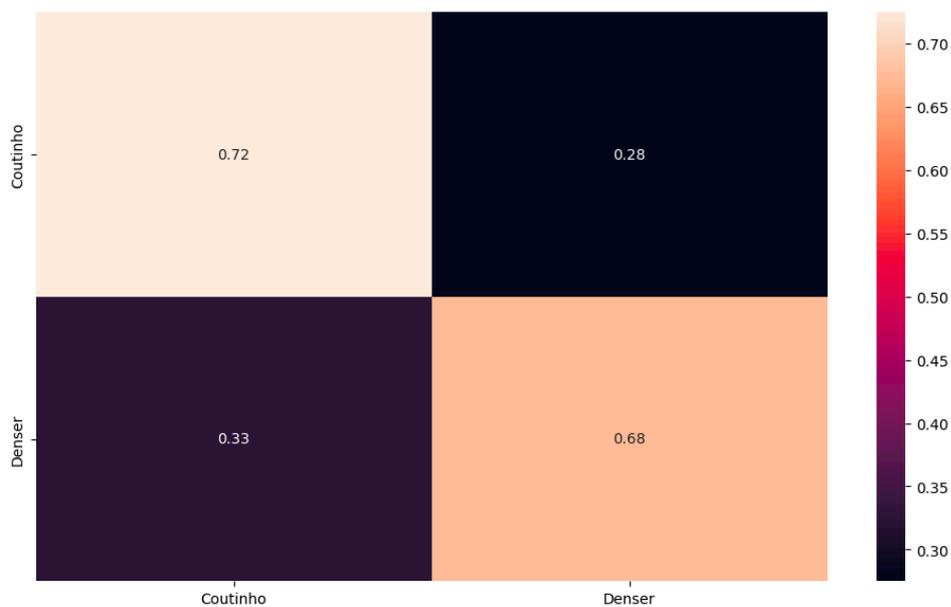


Figura 25 Matriz de confusão da identificação de autoria da rede

5. RESULTADOS E DISCUSSÕES

Com base nos resultados obtidos por meio da rede neural convolucional construída com o objetivo de identificar a autoria dos contos literários das autoras Sônia Coutinho e Márcia Denser, pode-se perceber a impossibilidade de fazer a identificação utilizando contos inteiros, uma vez que para um bom treinamento são necessários mais dados para que a rede possa iterar sobre. Por esse motivo, foi adotada a identificação de autoria por parágrafos como unidade para o treino e para a análise.

A identificação de autoria nos parágrafos das autoras escolhidas com o processamento de linguagem natural realizado pela rede neural convolucional, por sua vez, apresentou resultados positivos, uma vez que durante o treino, apresentou uma acurácia de 89,44% usando o dataset completo para o treinamento e de 82,87% com o dataset reduzido.

A identificação de autoria com o ChatGPT, por outro lado, apresentou um desempenho muito abaixo do modelo da rede neural, com uma acurácia de 43,12%. Com 69 parágrafos de teste classificados corretamente e 91 classificados de maneira incorreta. Porém, como citado anteriormente, a versão utilizada não possui o “fine-tuning” para processamento e treinamento de texto em várias prompts.

Nota-se que, a partir da classificação dos mesmos parágrafos, a rede convolucional apresentou melhores resultados na identificação da autoria, uma vez que a identificação feita pelo ChatGPT identificou a maioria dos parágrafos incorretamente. Uma amostra da comparação entre as identificações pode ser vista na Figura 26:

Parágrafos de teste	Autora	Autoria ChatGPT	ChatGpt	Rede Conv.	Autoria Rede
Ah, meu Deus, não adianta, não adianta mesmo, ...	Coutinho	Denser	Errou	Acertou	Coutinho
E foi cortado ao meio.	Denser	Coutinho	Errou	Errou	Coutinho
Na pista, casais oscilavam ao som de algo entre ...	Denser	Coutinho	Errou	Errou	Coutinho
Seu dinheiro da aposentadoria é suficiente para a...	Coutinho	Coutinho	Acertou	Acertou	Coutinho
- Quem é você?	Denser	Coutinho	Errou	Acertou	Denser
- Talvez seja falta de treino - riu Mestre - o truque...	Denser	Coutinho	Errou	Acertou	Denser
Acompanhados das três crianças caminham para ...	Coutinho	Denser	Errou	Acertou	Coutinho
Estou ouvindo, sim, não pare de falar que em ...	Coutinho	Denser	Errou	Acertou	Coutinho
Gostei de você desde aquele dia em que seu Max...	Denser	Coutinho	Errou	Errou	Coutinho
Branca voltou o rosto para a parede:	Denser	Denser	Acertou	Errou	Coutinho

Figura 26 Amostra da comparação realizada entre as classificações

6. CONCLUSÃO

Com base na digitalização, normalização e tratamento dos contos de Sônia Coutinho e Márcia Denser, no desenvolvimento de uma rede convolucional para a identificação de autoria e, por fim, nos testes executados e nos resultados apresentados, pode-se concluir que é possível identificar a autoria de parágrafos com o uso de ferramentas de NLP, ainda que apresente uma margem de imprecisão significativa nos testes realizados. Conclui-se também que, em comparação com o ChatGPT, a rede construída obteve um melhor desempenho na classificação, como pode ser visto na Figura 27:

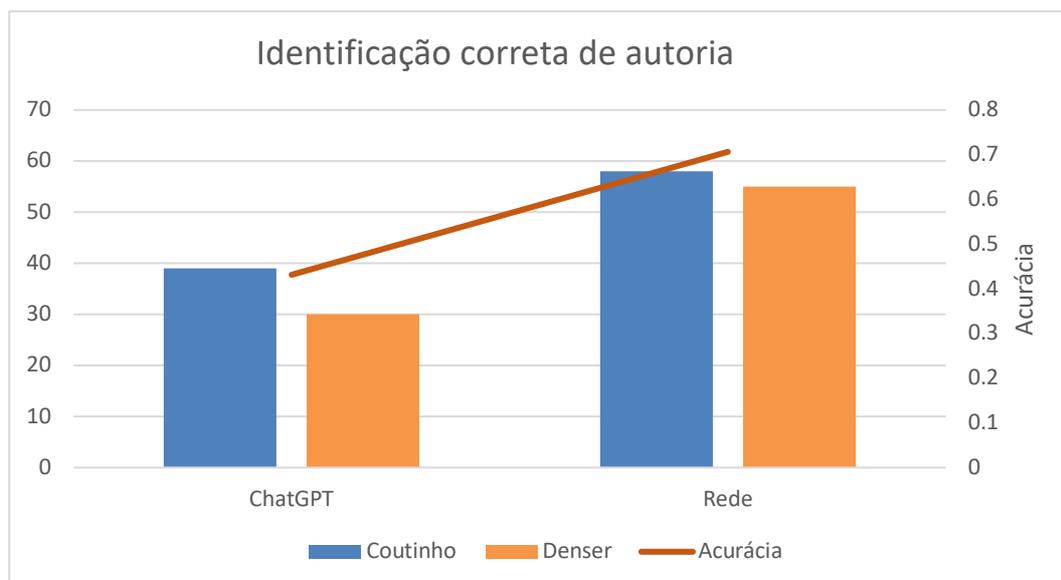


Figura 27 Identificação correta de autoria

Por fim, como possíveis trabalhos futuros, podem ser realizados testes envolvendo a seleção de textos de outros autores que possuam um corpus maior para avaliar o desempenho da rede desenvolvida com um volume maior de dados. A melhor possibilidade de um bom resultado é utilizando alguns modelos do GPT API que possuem a função fine tuning, como o gpt-3.5-turbo, babbage-002 e o davinci-003.

REFERÊNCIAS

ABUHAMED, M.; RHIM, J.; ABUHMED, T.; ULLAH, S.; KANG, S.; NYANG, D. Code authorship identification using convolutional neural networks. In: **Future Generation Computer Systems**, 2019, p. 104-115. Disponível em: <https://abuhamad.cs.luc.edu/pub/1-s2.0-S0167739X18315528-main.pdf>. Acesso em: 11 set. 2023.

BERRY, D. M. **What are the digital humanities**. Disponível em: <https://www.thebritishacademy.ac.uk/blog/what-are-digital-humanities/>. Acesso em: 11 set. 2023.

BERT x T5 x GPT-3 e o que achamos de cada modelo. NAMA. Disponível em: <https://simple.nama.ai/post/bert-x-t5-x-gpt-3>. Acesso em: 10 set 2023

BIRD, S.; KLEIN, E.; LOPER, E. **Natural Language Processing**. United States: O'Reilly Media, 2009.

BROWN, P. F.; LAI, J. C.; MERCER, R. L. **Aligning sentences in parallel corpora**. Disponível em: <https://dl.acm.org/doi/pdf/10.3115/981344.981366>. Acesso em: 10 set 2023.

CAMACHO, C. **CNNs for text classification**. Disponível em: https://cezannec.github.io/CNN_Text_Classification/. Acesso em: 10 set 2023.

COUTINHO, S. **Mil olhos e uma rosa**. Rio de Janeiro: 7 Letras, 2001.

COUTINHO, S. **Nascimento de uma mulher**. Rio de Janeiro: Ed. Civilização Brasileira S.A, 1971.

COUTINHO, S. **O último verão de Copacabana** [1985]. Rio de Janeiro: 7 Letras, 2004

COUTINHO, S. **Os venenos de Lucrecia** [1979]. Rio de Janeiro: 7 Letras, 2007.

COUTINHO, S. **Ovelha Negra e Amiga Loura**. Rio de Janeiro: Ed. 7 Letras, 2006.

COUTINHO, S. **Toda a verdade sobre a tia de Lúcia**. Rio de Janeiro: 7 Letras, 2011.

COUTINHO, S. **Uma certa felicidade** [1976]. Rio de Janeiro: Rocco, 1994.

DENSER, M. **Animal dos Motéis**. Rio de Janeiro: Ed. Civilização Brasileira S.A, 1981.

DENSER, M. **Toda prosa**. São Paulo: Nova Alexandria, 2001.

DENSER, M. **Diana Caçadora, Tango fantasma**: duas prosas reunidas. São Paulo: Ateliê Editorial, 2003.

DENSER, M. **Toda prosa II**. Rio de Janeiro: Record, 2008.

EISENSTEIN, J. **Natural Language Processing**. United States: MIT Press, 2018. Disponível em: <https://cseweb.ucsd.edu/~nnakashole/teaching/eisenstein-nov18.pdf>. Acesso em: 10 set. 2023.

FRĄCKIEWICZ, M. **ChatGPT Prompts and the Potential of AI-Powered Writing for Digital Humanities**. Disponível em: <https://ts2.space/en/chatgpt-prompts-and-the-potential-of-ai-powered-writing-for-digital-humanities/>. Acesso em: 10 set.

HANCOX, P. J. **A brief history of language processing**. Disponível em: https://www.cs.bham.ac.uk/~pjh/sem1a5/pt1/pt1_history.html. Acesso em: 10 set. 2023.

HU, K. **ChatGPT sets record for fastest-growing user base**. Disponível em: <https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/1>. Acesso em: 10 set. 2023.

SEGOVIA, A. **O que é IA Generativa? A importância e o uso das Inteligências Artificiais como ChatGPT, MidJourney e outras**. Disponível em: <https://www.alura.com.br/artigos/inteligencia-artificial-ia-generativa-chatgpt-gpt-midjourney>. Acesso em: 10 set. 2023.

JURAFSKY, D.; MARTIN, J. **Speech and Language Processing**. Disponível em: <https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf>. Acesso em: 13 set. 2023.

JARRAHI, A.; SAFARI, L. MOUSA, M. SLCNN: Sentence-Level Convolutional Neural Network for Text Classification. Disponível em: <https://arxiv.org/pdf/2301.11696.pdf>. Acesso em: 10 set. 2023.

LIU, Y.; OTT, M.; GOYAL, N.; DU, J.; JOSHI, M.; CHEN, D.; LEVY, O.; LEWIS, M.; ZETTMOYER, L.; STOYANOV, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. In: **CoRR** abs/1907.11692 (2019). Disponível em: <http://arxiv.org/abs/1907.11692>. Acesso em: 10 set 2023.

KHANHTRAN, C. **A Complete Guide to CNN for Sentence Classification with PyTorch** Disponível em: https://chriskhanhtran.github.io/_posts/2020-02-01-cnn-sentence-classification. Acesso em: 10 set 2023.

KOWSARI, K.; BROWN, D.; HEIDARYSAFA, M.; JAFARI MEIMANDI, K.; GERBER, M.; BARNES, L. HDLTex: Hierarchical Deep Learning for Text Classification. In: **Machine Learning and Applications (ICMLA)**, 2017 16th IEEE International Conference On. IEEE; 2017.

KIM, Y. Convolutional Neural Networks for Sentence Classification. In: **Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, 2014, p. 1746–1751. Disponível em: <https://aclanthology.org/D14-1181.pdf>. Acesso em: 10 set. 2023.

MERRITT, R. O que é um Modelo Transformer? Disponível em: <https://blog.nvidia.com.br/2022/04/19/o-que-e-um-modelo-transformer/> Acesso em: 10 set 2023.

O que é processamento de linguagem natural (PLN)? IBM, 2023. Disponível em: <https://www.ibm.com/br-pt/topics/natural-language-processing>. Acesso em: 10 set 2023.

RIGUETTO, G. **O uso da rede neural convolucional como extrator de características aplicado ao problema da identificação de escritores**. Monografia, Departamento acadêmico de Computação, Universidade Tecnológica Federal do Paraná, Campo Mourão, 2016, 48 p. Disponível em: https://repositorio.utfpr.edu.br/jspui/bitstream/1/6031/1/CM_COCIC_2016_2_02.pdf. Acesso em: 10 set. 2023.

RUBY, M. **How ChatGPT Works: The Model Behind The Bot**. Disponível em: <https://towardsdatascience.com/how-chatgpt-works-the-models-behind-the-bot-1ce5fca96286>. Acesso em: 10 set 2023

SHRESTHA, P.; SIERRA, S.; GONZÁLEZ, F.; ROSSO, P.; MONTES-Y-GÓMEZ, M.; SOLORIO, T. **Convolutional Neural Networks for Authorship Attribution of Short Texts**. Disponível em: <https://aclanthology.org/E17-2106.pdf>. Acesso em: 10 set. 2023.

The Evolution of ChatGPT: From Past to Present. AICONTENTFY. Disponível em: <https://aicontentfy.com/en/blog/evolution-of-chatgpt-from-past-to-present>. Acesso em: 10 set 2023.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, L.; POLOSUKHIN, I. Attention is All you Need. In: **Advances in Neural Information Processing Systems 30 (NIPS 2017)**. Disponível em: https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html. Acesso em: 10 set 2023.

ZHANG, Y.; WALLACE, B. C.; A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. Disponível em: <https://arxiv.org/pdf/1510.03820.pdf>. Acesso em: 10 set 2023.

ZHUKOV, V. **Text Classification on Extra Small Datasets: Fine-tuning vs ChatGPT.** Disponível em: <https://toloka.ai/blog/text-classification-on-extra-small-datasets/>. Acesso em: 10 set 2023.

OPENAI. **Fine-tuning:** Disponível em: <https://platform.openai.com/docs/guides/fine-tuning>. Acesso em: 20 out 2023.

OPENAI. **Usage policies.** Disponível em: <https://openai.com/policies/usage-policies>. Acesso em: 20 out 2023

OPENAI. **Models:** Disponível em: <https://platform.openai.com/docs/models/gpt-3-5>. Acesso em: 20 out 2023.