Faculdade de Computação e Informática

"Classificação de Textos Utilizando Redes Neurais para Análise de Artigos da Wikipédia com Python"

Gustavo Aragão Guedes, Gustavo Fernandes Costa, Júlia Rampani

Curso de Ciência da Computação
Universidade Presbiteriana Mackenzie – São Paulo, SP – Brasil

10376534@mackenzista.com.br, 10390377@mackenzista.com.br, 10395600@mackenzista.com.br

Resumo. *O projeto desenvolvido tem como objetivo criar um programa em Python que utiliza redes neurais para realizar a classificação de artigos extraídos da API da Wikipédia, categorizando-os por assunto. A tarefa de classificação de texto é um desafio comum em Processamento de Linguagem Natural (PLN), com aplicações significativas na organização e análise de grandes volumes de dados textuais. Para atingir esse objetivo, foi treinado um modelo de rede neural capaz de identificar padrões linguísticos presentes nos artigos, permitindo a atribuição automática de categorias predefinidas a cada texto. A solução implementada baseia-se em técnicas avançadas de PLN, como tokenização, vetorização e redes neurais profundas, garantindo alta eficiência na classificação dos artigos. O projeto não só contribui para a automação da organização de informações na Wikipédia, mas também oferece uma aplicação prática do uso de redes neurais em problemas reais de PLN.*

Palavras-chave: *Redes neurais; Classificação de texto; Processamento de Linguagem Natural (PLN); Análise de dados; Wikipédia*

1. Introdução

Nos últimos anos, a ciência de dados revolucionou a forma como organizações em todo o mundo gerenciam e analisam informações, proporcionando uma base sólida para decisões estratégicas mais precisas e eficazes.

Entre as diversas áreas de aplicação, o Processamento de Linguagem Natural (PLN) tem se destacado, permitindo a extração e análise de grandes volumes de texto de maneira automatizada. Essa evolução tecnológica é particularmente relevante em plataformas como a Wikipédia, que concentra uma enorme quantidade de informações organizadas em formato textual.

Dentre os desafios enfrentados nesse contexto, a classificação de texto emerge como uma tarefa crítica. Organizar dados textuais em categorias predefinidas não só otimiza a acessibilidade e compreensão das informações, mas também aprimora o uso estratégico desses dados em várias esferas.

Com foco nessa problemática, este trabalho propõe uma solução baseada em redes neurais para classificar automaticamente artigos extraídos da API da Wikipédia por assunto. Essa abordagem utiliza técnicas avançadas de PLN, como tokenização, vetorização e redes neurais profundas, para identificar padrões linguísticos e atribuir categorias de forma eficiente.

Além de contribuir para a automação na organização de informações, o projeto destaca a relevância das redes neurais como ferramenta prática para resolver problemas reais de PLN. Por meio da análise e categorização automatizada, busca-se não apenas explorar o potencial dessa tecnologia, mas também demonstrar sua aplicabilidade no contexto da ciência de dados, um campo que continua a enfrentar desafios no Brasil, principalmente no que diz respeito à sua implementação estratégica no ambiente corporativo.

2. Referencial Teórico

Este trabalho fundamenta-se em três pilares principais: o Processamento de Linguagem Natural (PLN), a utilização de redes neurais em tarefas de classificação de texto e as contribuições da API da plataforma de conhecimento Wikipédia para coleta de dados.

2.1. Processamento de Linguagem Natural (PLN)

O PLN é uma área interdisciplinar que combina linguística, ciência da computação e inteligência artificial para permitir que máquinas processem e compreendam linguagem humana. Segundo Russell e Norvig (2021), as técnicas de PLN são amplamente utilizadas para tarefas como análise de sentimento, tradução automática e classificação de texto. No contexto deste projeto, o PLN desempenha um papel central ao viabilizar a transformação de textos brutos em representações estruturadas que podem ser interpretadas por algoritmos de aprendizado de máquina.

O pré-processamento de texto, uma etapa crucial do PLN, inclui tarefas como tokenização, remoção de ruídos (HTML, pontuação e caracteres especiais) e vetorização. Essas etapas garantem que os dados textuais sejam convertidos em formatos numéricos utilizáveis pelas redes neurais, preservando características linguísticas essenciais.

2.2. Redes Neurais para Classificação de Texto

As redes neurais têm se destacado como ferramentas poderosas em tarefas de classificação de texto devido à sua capacidade de aprender padrões complexos a partir de dados não estruturados. Segundo Coppin (2010), redes neurais profundas, compostas por múltiplas camadas, são particularmente eficazes para problemas de classificação, pois conseguem capturar características de alto nível no texto.

A arquitetura empregada neste projeto inclui camadas de *embedding*, que transformam palavras em vetores densos de baixa dimensão, preservando relações semânticas entre elas. Adicionalmente, uma camada de *GlobalAveragePooling1D* permite a compactação das informações extraídas, enquanto camadas densas com funções de ativação ReLU e softmax realizam a classificação final. Essa abordagem é suportada por técnicas de otimização, como o algoritmo Adam, e funções de perda, como *sparse categorical crossentropy*, que maximizam a acurácia da predição.

2.3. Coleta de Dados via API

A utilização da API da Wikipédia para coleta de dados demonstra o potencial das interfaces de programação de aplicativos como fontes acessíveis e ricas de informações textuais. De acordo com Szeliski (2021), a automação no acesso a dados por meio de APIs é fundamental para projetos de aprendizado de máquina, pois permite a criação de conjuntos de dados extensos e diversificados.

No contexto deste trabalho, os dados extraídos da Wikipédia são categorizados em tópicos como Ciência, História e Tecnologia. A diversidade dos textos fornecidos pela API é essencial para o treinamento de um modelo robusto e generalizável. Apesar de serem amplamente utilizados, os snippets textuais coletados apresentam limitações semânticas que foram parcialmente mitigadas por meio de técnicas avançadas de PLN.

3. Metodologia

Para resolver esse problema, será necessário utilizar um conjunto de ferramentas e bibliotecas em Python focadas em análises de dados, machine learning e processamento de linguagem natural.

3.1. Ferramentas

3.1.1. Pandas: Utilizada para manipulação e estruturação dos dados extraídos da API. Será útil para carregar os dados, realizar limpeza textual, e organizar os artigos em um formato tabular adequado para o processamento.

3.1.2. NumPy: Necessário para realizar operações matemáticas, manipulação de arrays e vetorização dos dados textuais, garantindo eficiência no processamento dos dados de entrada para as redes neurais.

3.1.3. Matplotlib: Ferramenta de visualização para gerar gráficos que auxiliam no entendimento dos dados, como a distribuição dos artigos por categorias e a performance do modelo após o treinamento.

3.1.4. Scikit-Learn: Bibliotecas que serão utilizadas para tarefas como tokenização do texto (transformação do texto em números), divisão dos dados em

conjuntos de treino e teste, além de oferecer métricas de avaliação como acurácia e matriz de confusão.

3.1.5. TensorFlow: Principal ferramenta para construir e treinar as redes neurais. Será utilizada para desenvolver um modelo de classificação de texto com redes neurais profundas (Deep Learning). A arquitetura da rede pode incluir camadas densas e camadas de embedding para representar os textos como vetores e aprender a partir de seus padrões.

3.2. Métodos

3.2.1. Coleta de dados da API da Wikipédia: O programa fará requisições à API para baixar os artigos, utilizando a biblioteca Requests ou Wikimedia API. Serão coletados artigos de várias categorias, garantindo uma variedade suficiente para o treinamento.

3.2.2. Pré-processamento de texto:

- **Limpeza de texto:** Remoção de HTML, caracteres especiais, pontuação, e outros ruídos indesejados.
- **Tokenização:** Transformação dos textos em tokens (palavras ou sub-palavras)
- **Stopwords:** Remoção de palavras muito comuns que não agregam significado, como artigos e preposições.
- **Lematização:** Redução das palavras para suas formas base (Por exemplo, “correndo” para “correr”).

3.2.3. Criação do modelo de classificação:

- **Representação de textos:** Utilizando camadas de embedding no TensorFlow para transformar os textos em vetores de números, capturando as relações semânticas entre as palavras.
- **Estrutura da rede neural:** Construir uma rede neural composta por:
  - Camadas densas fully connected, responsáveis por extrair padrões latentes,
  - Função de ativação ReLu para introduzir não-linearidade.
  - Camada softmax na saída para classificar os artigos nas diferentes categorias.
- **Função de perda:** Usar categorical crossentropy para medir a divergência entre as previsões do modelo e as classes reais.
- **Otimização:** Algoritmo Adam será utilizado para ajustar os pesos da rede durante o treinamento.

3.2.4. Treinamento e validação

- Divisão do dataset em conjunto de treinamento e conjunto de teste, geralmente 80/20, para treinar o modelo e avaliar sua performance.
- Ajuste de hiperparâmetros como número de épocas, tamanho do batch, e taxa de aprendizado para otimizar o desempenho do modelo.

3.2.5. Avaliação do modelo

- **Acurácia:** Principal métrica para avaliar o sucesso da classificação.
- **Matriz de Confusão:** Para entender como o modelo performa em cada categoria, identificando quais classes está sendo corretamente classificadas e onde há confusão.

3.2.6. Visualização dos resultados

- Após o treinamento, a biblioteca Matplotlib será usada para gerar gráficos que mostram a evolução da acurácia e da perda ao longo das épocas.
- Geração de uma matriz de confusão para visualizar o desempenho do modelo por categoria.

4. Resultados e discussão

O código desenvolvido para a classificação de artigos de Wikipédia utilizando redes neurais foi eficaz na organização e categorização dos textos. A seguir, destacam-se os principais resultados e as métricas de desempenho do modelo:

4.1. Coletas de dados: O processo de coleta de artigos foi bem-sucedido, com 200 artigos extraídos para cada uma das categorias predefinidas, incluindo “Ciência”, “História”, “Tecnologia”, “Arte”, “Esportes”, “Literatura”, “Filmes” e “Música”. Cada artigo consistiu em um título, um snippet de texto e a categoria associada, obtendo-se um dataset equilibrado e diversificado.

4.2. Pré-processamento de texto: O pré-processamento foi realizado de forma eficiente, com a remoção de HTML, caracteres especiais, pontuação e a tokenização do texto. A limpeza e preparação dos dados possibilitaram a transformação dos textos em uma forma adequada para o modelo de rede neural.

4.3. Modelo de rede neural: O modelo foi construído com uma camada de embedding para transformar os textos em vetores, seguida de uma camada de pooling global para capturar características importantes do texto. A arquitetura final incluiu uma camada densa e uma camada de saída com softmax para classificação das categorias.

4.4. Treinamento e avaliação: O modelo foi treinado por 10 épocas, com um batch size de 32. Durante o treinamento, as métricas de acurácia e perda foram monitoradas, mostrando uma evolução positiva. A acurácia no conjunto de teste foi de aproximadamente 85%, indicando que o modelo foi capaz de classificar corretamente a maioria dos artigos.

4.5. Resultados de classificação: A avaliação do modelo, com base na matriz de confusão e na contagem de artigos classificados corretamente, mostrou que o modelo

teve um bom desempenho na classificação das categorias mais bem definidas como “Tecnologia” e “Ciência”. No entanto, categorias com sobreposição de tópicos, como “Arte” e “Literatura”, apresentaram um desempenho um pouco inferior.

4.6. Visualização de resultados: Gráficos de acurácia e perda ao longo das épocas demonstraram a eficácia do treinamento, com uma melhoria contínua na acurácia e uma redução na perda tanto no conjunto de treino quanto no de validação. A matriz de confusão forneceu uma visão detalhada do desempenho do modelo por categoria, destacando onde ocorreram os erros de classificação.

5. Conclusão

Os resultados confirmam que a combinação de redes neurais profundas com técnicas de pré-processamento de texto e tokenização é uma abordagem eficaz para a classificação automática de artigos. A implementação foi capaz de realizar a categorização de textos com alta precisão, mas há margem para melhorias, especialmente nas categorias com maior complexidade semântica. Modelos pré-treinados, como o BERT, poderiam ser explorados para melhorar ainda mais o desempenho, principalmente em tarefas mais desafiadoras de PLN. A visualização dos resultados e a análise da matriz de confusão foram fundamentais para identificar áreas de melhoria no modelo.

6. Referências bibliográficas

COPPIN, B. Inteligência artificial. Rio de Janeiro: LTC, 2010.

RUSSELL, S.; NORVIG, P. Inteligência artificial. Rio de Janeiro: LTC, 2021.

SZELISKI, R. Computer Vision: Algorithms and Applications. New York: Springer, 2021.

NEHA, S.; YADAV, Y.; GOYAL, Y. Introduction to machine learning. International Journal of Advanced Research in Science, Communication and Technology, Naksh Solutions, p. 100–105, mar. 2024. ISSN 2581-9429.

DE FREITAS, S. M.; MARQUES, A.; ARRUDA, D. R. ATUAL CENÁRIO DE PUBLICAÇÕES COM USO DE MACHINE LEARNING PARA DIAGNÓSTICOS EM SAÚDE. Disponível em: