Análise de Técnicas de Aumento de Dados para Processamento de Linguagem Natural

Deivid Gomes Silva
RAI - Robotics and Artificial Intelligence
Centro de Ciências Exatas e Tecnológicas
Universidade Federal do Recôncavo da Bahia
Cruz das Almas, Brasil
deividgomes@aluno.ufrb.edu.br

André Luiz Carvalho Ottoni
RAI - Robotics and Artificial Intelligence
Centro de Ciências Exatas e Tecnológicas
Universidade Federal do Recôncavo da Bahia
Cruz das Almas, Brasil
andre.ottoni@ufrb.edu.br

***Abstract*—O estudo do processamento de linguagem natural (PLN) é essencial para a interação humano-computador do futuro, pois fornece aos computadores a capacidade de compreender a Linguagem Natural. Entretanto, há uma escassez de pesquisas científicas, especialmente voltadas para a língua portuguesa, que abordem qualitativamente as técnicas de aumento de dados para PLN. Dessa forma, os objetivos deste trabalho são listar e descrever técnicas de aumento de dados para o processamento de linguagem natural e entender o impacto dessas técnicas em dois sistemas de conversação. Para isso, a metodologia proposta nesse trabalho abordou as seguintes etapas: uma revisão de literatura em artigos científicos relevantes sobre a temática; a implementação de dois sistemas de conversação com redes neurais treinadas nos cenários com aumento de dados e sem aumento de dados. Nos resultados, com a aplicação das técnicas de EDA e BackTranslation, obteve-se 94,43% e 92,14% de acurácia com os dados aumentados a partir do primeiro dataset e 86,92% de acurácia com os dados aumentados a partir do segundo dataset, além de melhorias gerais nas métricas de precisão e recall. Concluiu-se que as técnicas abordadas geram melhoria de desempenho satisfatória e importante diversificação dos textos de treino.**

***Index Terms*—Processamento de linguagem natural; Data augmentation; Sistema de conversação; Redes neurais artificiais;**

I. Introdução

Os avanços notados nos mais recentes modelos de processamento de linguagem natural (PLN) reafirmam o potencial que essa sub-área da inteligência artificial possui. Os estudos em PLN buscam possibilitar que os computadores reconheçam a linguagem natural [1]. Tal fato é animador por desenhar um futuro onde as interações humano-computador seriam tão práticas quanto as interações entres seres humanos [2].

Aplicações interessantes envolvendo PLN podem ser notadas em artigos como [3], [4] que desenvolvem algoritmos para coleta de conteúdos com opiniões e informações relevantes da rede social Twitter. No trabalho [5] foi realizada, com auxílio de PLN, uma mineração de dados em *logs* de servidores *Web* para detecção de robôs de mecanismos de busca. Já nos artigos [6] e [7], uma arquitetura inovadora chamada *Generative Pre-trained Transformer (GPT)* mostra o quão impactante as implementações em PLN podem se tornar, pois o *GPT* foi o primeiro passo para criação do famoso modelo de linguagem *ChatGPT*.

No entanto, um dos desafios para desenvolver sistemas em aprendizado de máquina para PLN é a necessidade de grandes volumes de dados para o treinamento dos modelos neurais. Nesse aspecto, a prática de *data augmentation* faz-se essencial para o desempenho satisfatório dos modelos em PLN. Em seu estudo sobre o aumento de dados textuais [8] afirma: "Sem aumento de dados ou regularização em geral, as Redes Neurais Profundas são propensas a aprender correlações espúrias e memorizar padrões de alta frequência que são difíceis de serem detectados pelos humanos". Pensando nisso, os autores de [8]–[11] se esforçam em sumarizar e investigar as técnicas para aumento de dados textuais existentes e em convidar a comunidade acadêmica a pesquisar sobre a temática.

A técnica de *data augmentation* tem sido amplamente utilizada em outras áreas, como visão computacional, para aumentar o tamanho do conjunto de dados e melhorar o desempenho dos modelos de aprendizado de máquina [12]–[14]. No entanto, o uso de técnicas de *data augmentation* em PLN ainda é pouco explorado e há uma escassez de pesquisas científicas, especialmente na Língua Portuguesa, que avaliem a eficácia e aplicabilidade dessas técnicas nessa área.

Dessa forma, os objetivos deste trabalho estão em: listar e descrever uma série de técnicas de aumento de dados para processamento de linguagem natural; aplicação de algumas técnicas de *data augmentation* em dois conjuntos de dados textuais. Esses conjuntos de dados são designados para criação de um sistema de conversação (*chatbot*) para auxílio ao aprendizado de Geometria Analítica (GA) e outro para o aprendizado de Inteligência Artificial (IA).

Para atingir esses objetivos serão realizadas as seguintes tarefas: uma pesquisa por técnicas de *data augmentation* para PLN; a construção dos conjuntos dados para os sistemas de conversação; a escrita do algoritmo em aprendizado de máquina com pré-processamento de texto, instanciação do modelo, treino e avaliação das técnicas.

Este artigo está dividido em cinco seções. A seção 2 apresenta a fundamentação teórica. A seção 3 discorre sobre a metodologia. A seção 4 apresenta os resultados da pesquisa. E, por fim, a seção 5 realiza a conclusão do trabalho.

II. Fundamentação Teórica

A. Processamento de Linguagem Natural

Entre os conceitos abordados neste trabalho está o processamento de linguagem natural. Esta é uma subárea da inteligência artificial que tem como objetivo permitir que as máquinas entendam, processem e gerem linguagem natural. O PLN envolve uma série de técnicas e algoritmos que permitem às máquinas realizar tarefas como análise de sentimentos, classificação de textos, tradução automática, entre outras [15]. Essas tarefas são importantes em diversos contextos, desde a análise de dados em redes sociais até a automação de tarefas em empresas [1], [16].

A linguagem humana é extremamente complexa, com várias nuances, ambiguidades, contextos e variações. O PLN aborda o desafio de lidar com essa complexidade, desenvolvendo algoritmos e técnicas que possibilitam aos computadores processar, analisar e extrair informações úteis a partir de dados textuais ou falados [17].

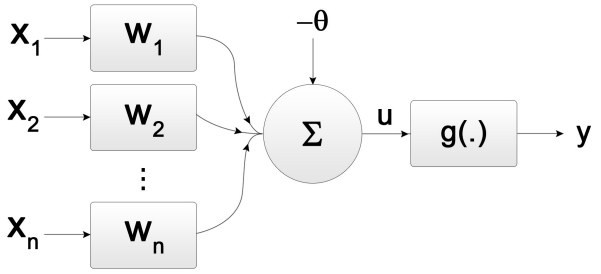
Essas são apenas algumas das muitas tarefas realizadas pelo PLN. O campo do PLN se beneficia do avanço das técnicas de aprendizado de máquina, como o uso de redes neurais profundas e algoritmos de aprendizado supervisionado e não supervisionado, que ajudaram a impulsionar o desempenho em várias tarefas de processamento de linguagem natural.

B. Redes Neurais Artificiais

As redes neurais artificiais são modelos computacionais inspirados no funcionamento do sistema nervoso humano. Elas são compostas por um conjunto de neurônios artificiais, ou nós, interconectados. Esses neurônios recebem entradas, realizam cálculos e produzem saídas. As conexões entre os neurônios são ponderadas e ajustadas durante o processo de treinamento da rede [18], [19]. Uma ilustração de neurônio artificial, baseado em [20], pode ser visto na Figura 1.

Aqui deve-se apresentar também a conceituação das Redes Neurais Densas (RNDs) utilizadas nos sistemas de conversação desenvolvidos. As RNDs são um tipo de arquitetura de redes neurais artificiais em que cada nó em uma camada está conectado a todos os nós na camada seguinte.

Fig. 1. Ilustração de neurônio artificial



Na Figura 1 tem-se x_1, x_2 e x_n como variáveis de entrada para o neurônio artificial, w_1, w_2 e w_n são os pesos sinápticos, Σ trata-se do combinador linear, θ é o limiar de ativação, u é o potencial de ativação, $g(.)$ é a função de ativação, e y é a variável de saída do neurônio.

Em termos matemáticos, o processamento realizado pelo neurônio artificial pode ser descrito pelas expressões (1) e (2):

$$u = \sum_{i=1}^{n} w_i.x_i - \theta \tag{1}$$

$$y = g(u). \tag{2}$$

III. Metodologia

A metodologia de pesquisa seguiu a seguinte ordem: (i) busca por técnicas de aumento de dados textuais na literatura, (ii) construção dos conjuntos de dados originais, (iii) ampliação dos conjuntos de dados com as técnicas encontradas e (iv) treinamento dos modelos para todos os *datasets* criados.

A. Pesquisa por técnicas de aumento de dados para PLN

Entre os objetivos definidos para este trabalho está a listagem de técnicas de aumento de dados para PLN. Para alcançar esse objetivo, uma revisão de literatura de artigos científicos relevantes foi feita. As buscas por artigos na temática foram feitas através do *Google Scholar*,*IEEE Xplore* e do Portal de Periódicos CAPES. Os trabalhos visitados nessa etapa foram publicados em revistas de alto fator de impacto ou apresentados em congressos nacionais importantes como o Congresso Brasileiro de Inteligência Computacional (CBIC).

Os dois artigos de maior contribuição para entendimento de *data augmentation* em PLN são: *A Survey of Data Augmentation Approaches for NLP* [9] e *EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks* [10]. O trabalho de [9] tem como objetivo servir como um guia para os pesquisadores de PLN decidirem sobre quais técnicas de aumento de dados usar, e para isso os autores realizam uma sumarização dos métodos para aumento de dados textuais existentes até 2021. Já o artigo [10] discorre sobre quatro técnicas que podem ser facilmente aplicadas. Essas técnicas são expostas ao longo deste artigo e utilizadas no treinamento dos *chatbots*.

B. Implementação dos chatbots

Cada um dos sistemas de conversação (*chatbots*) construídos para este artigo possui essencialmente três partes:

- arquivos de conjunto de dados de treinamento original e aumentados;
- arquivo com o pré-processamento de texto, além de estruturação e treinamento da rede neural artificial;
- arquivo com o processamento de entradas de perguntas por usuários e impressão de respostas pelo sistema treinado.

A linguagem de programação utilizada foi Python por conta de sua compatibilidade com as bibliotecas de *deep learning*, PLN e *data augmentation*. Basicamente, as principais bibliotecas de código utilizadas foram:

- *TensorFlow* [21] para construção de modelos de rede neural artificial;
- *Natural Language Toolkit (NLTK)* [22] para uso das funções comuns em PLN;

- *BackTranslation* para uso da técnica de *data augmentation* [23], [24];
- e *TextAttack* [25] para uso de técnicas frequentemente usadas em *data augmentation* para PLN.

Os *chatbots* desenvolvidos neste artigo estão centrados no ensino de conceitos básicos de geometria analítica e inteligência artificial, individualmente. Essa escolha foi feita para ressaltar que a análise realizada busca abranger a eficiência e aplicabilidade das técnicas de aumento de dados para PLN, e não o desempenho dos *chatbots* desenvolvidos. Dessa forma, os *chatbots* poderiam ter outra temática central ou serem substituídos por outra tarefa de classificação em PLN.

C. Conjunto de dados

Os arquivos com conjunto de dados de treinamento são do tipo JSON e têm a distribuição como apresentada na Figura 2.

Fig. 2. Exemplo do padrão de distribuição dos textos nos arquivos JSON para o conjunto de dados de IA

```
{"intents": [
{"tag": "artificial_intelligence",
   "patterns": [
        "What is artificial intelligence?",
        "What does artificial intelligence means?",
        "What AI stand for?"
        ],
   "responses": [
        "Artificial intelligence (AI) is the ability
        of a machine to perform human-like
        skills such as thinking, learning, understanding
        and creativity."
        ]
 }
]}
```

A lista presente nos arquivos tem o nome de *intents* e é responsável por agrupar as classes de classificação da rede neural artificial. Cada uma dessas classes é uma *tag* no arquivo JSON. Percebe-se que para cada classe existe um padrão (*pattern*) de textos de entrada e uma lista de possíveis respostas (*responses*) para esses textos de entradas. O modelo treinado deve ser capaz de informar qual a classe da pergunta (*pattern*) para que um algoritmo aleatório escolha uma das possíveis respostas (*responses*) à pergunta inserida.

Como exemplificação do funcionamento do *chatbot* pode-se analisar a Figura 2. A seguir estão as perguntas da Figura 2, relacionadas a classe de inteligência artificial, traduzidas do inglês para o português: ”O que é inteligência artificial?”, ”O que inteligência artificial significa?” e ”O que a sigla IA significa?”. Com o *chatbot* completo, o usuário final, ao realizar qualquer pergunta semelhante às apresentadas, deve ter como resposta: ”A inteligência artificial (IA) é a capacidade de uma máquina de executar habilidades semelhantes às humanas, como pensamento, aprendizado, compreensão e criatividade”.

Analogamente, para o *chatbot* GA, vê-se na Figura 3 as seguintes perguntas, traduzidas para o português, relacionadas a classe de equação do plano: ”Como determinar a equação do plano?” e ”Qual é a equação do plano?”. Com o sistema final, ao receber alguma pergunta semelhante às apresentadas, o *chatbot* deve fornecer como resposta: ”A equação do plano pode ser obtida com três pontos pertencentes ao plano ou com dois pontos pertencentes ao plano e um vetor paralelo”.

Fig. 3. Exemplo do padrão de distribuição dos textos nos arquivos JSON para o conjunto de dados de GA

```
{"intents": [
{"tag": "plane_equation",
   "patterns": [
     "How to determine the equation of the plane?",
     "What is the plane equation?"
      ],
   "responses": [
     "The equation of the plane
     can be obtained with three points belonging
     to the plane or with two points belonging to
     the plane and a parallel vector."
     ]
 }
]}
```

Os *chatbots* concluídos possuem diversas outras classes como essas apresentadas, cada uma com suas perguntas específicas. Para o *chatbot* IA, por exemplo, algumas das outras classes são ”Conjunto de dados”, ”Visão computacional” e ”Taxa de aprendizado”. E para o *chatbot* GA algumas das outras classes são ”Equação da reta”, ”Módulo de vetor” e ”Vetor unitário”, por exemplo.

Neste ponto destaca-se que todos os textos dos conjuntos de dados encontram-se na língua inglesa, dado que todas as bibliotecas utilizadas neste trabalho foram desenvolvidas para processar textos em inglês. Dessa forma, não apenas a implementação dos *chatbots*, mas também todo seu funcionamento, deverá ocorrer em inglês. Além disso, todos os textos do conjunto de dados foram elaborados pelo autor, e isso se dá por dois motivos:

- não se pretendia investir esforços em extensos algoritmos de pré-processamento de textos, os quais são geralmente necessários para os grandes, e bagunçados, *datasets* públicos;
- e, em principal, buscava-se perceber a diversificação de palavras gerada pelas técnicas de *data augmentation*. Sabe-se que os *datasets* públicos já dispõem de uma ampla gama de sinônimos e palavras diversas. Dessa forma, a aplicação de *data augmentation* nesses *datasets* seria eficiente para o aumento de dados mas pouco impactante na diversificação dos textos.

D. Data Augmentation

Para a etapa de aumento de dados, foram utilizadas as técnicas de *BackTranslation* (com Chinês como idioma intermediário) e todas as operações do *Easy Data Augmentation* [10]. Dois novos *datasets* (um com aumento feito por *BackTranslation* e outro com aumento feito por *Easy*

Data Augmentation) foram produzidos para cada um dos dois conjuntos de dados originais.

No total, seis conjuntos de dados foram gerados nessa etapa:

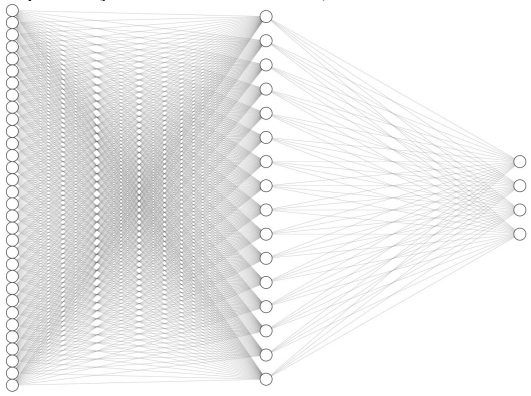
- *dataset* GA original;
- *dataset* GA aumentado por *BackTranslation*;
- *dataset* GA aumentado por *EDA*;
- *dataset* IA original;
- *dataset* IA aumentado por *BackTranslation*;
- *dataset* IA aumentado por *EDA*;

O trecho de código responsável por aplicar as técnicas de supracitadas começa lendo o arquivo JSON como um dicionário Python. Após isso, uma estrutura de repetição coleta todos os itens da lista de *patterns*, executa o *BackTranslation* e adiciona os textos resultantes do processo de volta à lista de *patterns*. Para finalização, o dicionário Python aumentado pela técnica de *data augmentation* é salvo em outro arquivo JSON.

E. Rede neural densa

Neste trabalho, foi proposta uma rede neural densa devido a simplicidade da tarefa de classificação e ao pequeno tamanho dos conjuntos de dados. A rede neural densa utilizada foi do tipo "Sequential" pela biblioteca da *TensorFlow* e possuía uma camada de entrada com 128 neurônios, uma camada escondida com 64 neurônios, ambas camadas com função de ativação relu, e uma camada de classificação/saída com 16 neurônios com função de ativação *softmax*. Entre as camadas citadas existem camadas de dropout com taxa de 0,5. A ilustração da arquitetura proposta pode ser vista na Figura 4.

Fig. 4. Representação da rede neural densa (número de neurônios ilustrativo)



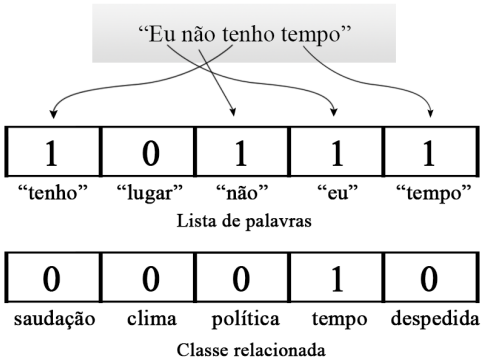
F. Treinamento dos modelos

A primeira ação realizada no pré–processamento do conjunto de dados é a tokenização e lematização dos textos presentes nas listas *patterns*. Todas as palavras que passaram por esses processos são armazenadas em uma lista que pode ser chamada de vocabulário ou apenas lista de palavras.

Para que os dados textuais fossem representados em formato numérico a técnica *bag of words* foi utilizada. Como explica Thakur [17] "Na *bag of words*, criamos uma enorme matriz esparsa que armazena contagens de todas as palavras em nosso corpus", sendo o corpus o agrupamento de todas as sentenças presentes no conjunto de dados.

O procedimento para a criação da *bag of words* é simples: para cada sentença do *dataset*, o algoritmo cria um vetor do mesmo tamanho que o vocabulário, com cada elemento representando se a palavra correspondente aparece ou não na sentença. Se uma palavra aparece na sentença, seu elemento correspondente no vetor é definido como 1 e, se não aparece, o elemento é definido como 0. Neste caso, tem-se uma *bag of words* binária. Posteriormente, basta associar o vetor obtido à uma classe como visto na Figura 5.

Fig. 5. Exemplo do procedimento de conversão de texto para número



Por fim, a junção do vetor previamente citado com o vetor que define a classe da sentença serve de entrada para a rede neural artificial. Todos os treinamentos aconteceram com os mesmos hiperparâmetros, sendo eles: 0,01 para taxa de aprendizado, 50 épocas, 10 para o tamanho do *batch* e otimizador de gradiente descendente com momentum de 0,9. Esses hiperparâmetros foram escolhidos de forma arbitrária devido a simplicidade da classificação em questão.

IV. Resultados

A. Técnicas de aumento de dados para PLN encontradas

Neste tópico são apresentadas as técnicas mais comuns e eficientes em *data augmentation* para a área de processamento de linguagem natural.

1) BackTranslation: Nesse método, a ideia central é traduzir os dados de texto para um idioma arbitrário (idioma intermediário) e então realizar a tradução de volta para o idioma original. Com isso gera-se sentenças novas que podem ser diferentes das originais mas com o mesmo contexto e sentido semântico. Os sistemas de tradução concentram-se em manter a semântica da sentença de entrada mesmo que algumas palavras sejam substituídas, excluídas ou realocadas na frase. Dessa forma, o método de *BackTranslation* é ótimo para aumentar a quantidade de dados de uma mesma classe. A seguir, na Figura 6, é possível observar o funcionamento do método.

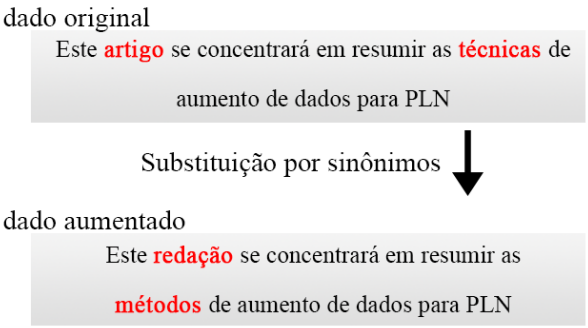
2) EDA: O aumento de dados fácil, ou como é mais conhecido *EDA: Easy Data Augmentation* [10], usa métodos de aumento de dados tradicionais e muito simples. EDA consiste em quatro operações simples que fazem um trabalho

Fig. 6. Funcionamento do *BackTranslation*

surpreendentemente bom na prevenção do *overfitting* e ajudam a treinar modelos mais robustos. As quatro operações citadas são: substituição por sinônimos, inserção aleatória, troca aleatória e exclusão aleatória.

3) Substituição por sinônimos: Na substituição por sinônimos são escolhidas, aleatoriamente, algumas palavras da frase que não sejam *stopwords*, após isso ocorre a substituição de cada uma dessas palavras por um de seus sinônimos escolhido ao acaso. Na Figura 7, é possível observar o funcionamento do método.

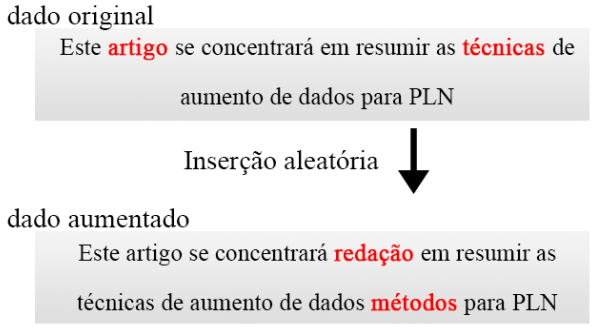
Fig. 7. Funcionamento da substituição por sinônimos



4) Inserção aleatória: Na inserção aleatória o código deve encontrar um sinônimo aleatório de alguma palavra aleatória na frase que não seja uma *stopword*, após isso inserir esse sinônimo em uma posição aleatória na frase. Todo esse processo pode ser feito mais de uma vez. Na Figura 8, é possível observar o funcionamento dessa técnica.

5) Troca aleatória: Na troca aleatória são escolhidas aleatoriamente duas palavras na frase cujas posições serão trocadas. Esse processo pode ser feito mais de uma vez.

Fig. 8. Funcionamento da inserção aleatória

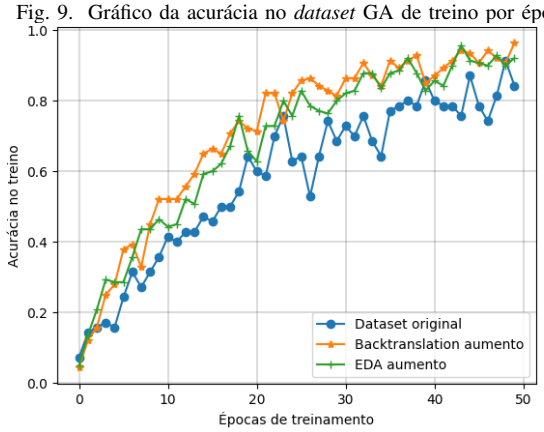


6) Exclusão aleatória: Na exclusão aleatória ocorre a remoção aleatória de alguma palavra da frase seguindo uma probabilidade pré-definida.

B. Desempenho do aumento de dados nos chatbots

Neste tópico são apresentados os resultados do desempenho do aumento de dados nos *datasets* construídos. Nas Figuras 9-12 pode-se ver os gráficos de acurácia por época de treinamento. Foram registradas as interações na parte de treino e de validação dos conjuntos de dados. Os *datasets* não possuiam dados o suficiente para que a parte de teste fosse construída e analisada adequadamente. Ressalta-se que os *datasets* possuíam um número de amostras por classe apropriado, portanto a avaliação por acurácia não é tendenciosa.

Fig. 9. Gráfico da acurácia no *dataset* GA de treino por época



Pode-se notar a melhoria no desempenho dos modelos nos *datasets* aumentados comparados com os *datasets* originais.

Na Tabela 1 vê-se os resultados finais de outras métricas no treino.

As melhorias em desempenho por parte dos *datasets* aumentados, notadas na Tabela 1 e nas Figuras 9-12, eram esperadas visto que esses *datasets* dispõem de um número expandido de amostras de treinamento. Entretanto, além de mostrar o impacto das técnicas de *data augmentation* nas métricas de aprendizado de máquina, busca-se reconhecer o papel fundamental de diversificação do conjunto de dados que as técnicas carregam.

Fig. 10. Gráfico da acurácia no *dataset* GA de validação por época

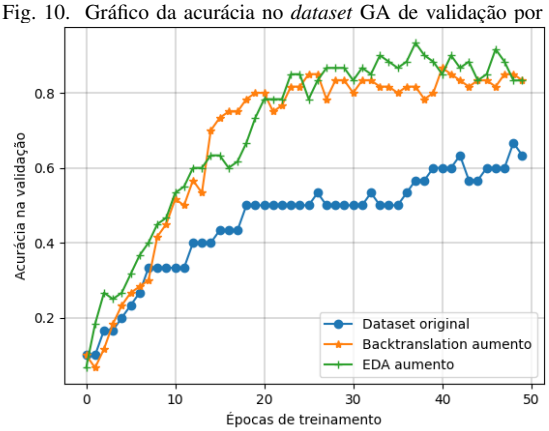
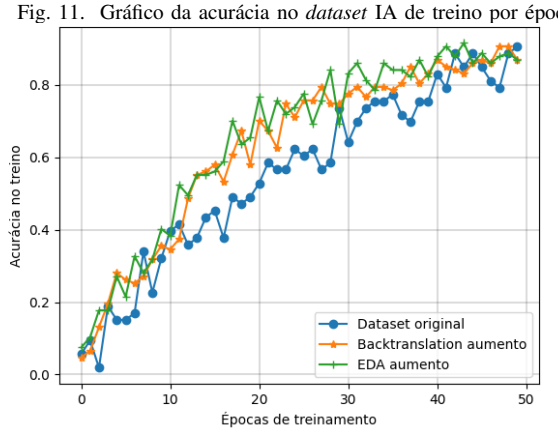


Fig. 11. Gráfico da acurácia no *dataset* IA de treino por época



Como exemplo pôde-se notar, utilizando uma comparação de arquivos JSON, que algumas palavras relacionadas com perguntas sobre equação geral do plano e equação da reta, como *flat* (plano) e *straight* (reto), possuem ocorrência no *dataset* de Geometria Analítica aumentado por *BackTranslation* mas não existem no *dataset* original. Também notou-se que a palavra *size* (tamanho), essencial para detecção de perguntas sobre módulo de vetores, aparece no *dataset* de GA aumentado por EDA mas não existe no *dataset* original. Ou seja, o modelo treinado não teria conhecimento da existência dessas palavras antes da aplicação de técnicas de *data augmentation*.

De maneira similar, a utilização das técnicas de aumento de dados também adicionou palavras diferentes aos *datasets* originais para o *chatbot* IA. Como exemplo, tem-se a ocorrência de *enhance* (aumentar) entre as amostras da classe de perguntas sobre *data augmentation* e *view* (visualizar) entre as amostras da classe de perguntas sobre visão computacional.

V. Conclusão

Neste artigo foram apresentadas técnicas recorrentes e eficazes no aumento de dados de textos para aplicações em processamento de linguagem natural. Além disso, foi realizada uma análise comparativa, em dois sistemas de conversação, do desempenho de modelos com conjunto de dados original

Fig. 12. Gráfico da acurácia no *dataset* IA de validação por época

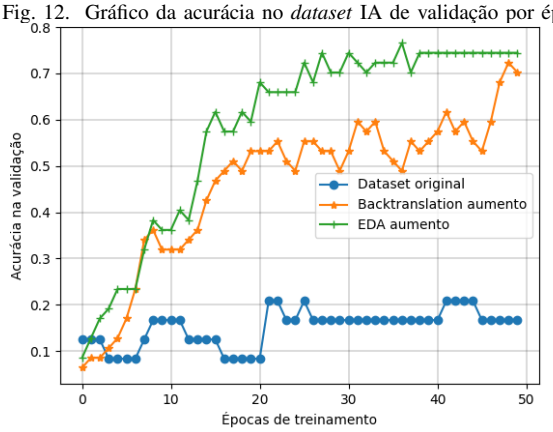


Tabela I
Métricas de treino para os *datasets* construídos.

Datasets	Acurácia (%)	*Recall* (%)	Precisão (%)
GA original	84,29	70,00	94,23
GA *BackTranslation*	94,43	90,71	97,69
GA EDA	92,14	90,71	96,95
IA original	90,57	67,92	92,31
IA *BackTranslation*	86,92	84,11	92,78
IA EDA	86,92	79,44	96,59

e aumentados. Ambos pontos foram definidos como objetivos desse trabalho e foram alcançados com êxito.

Este trabalho apresenta o impacto positivo das técnicas de *EDA* e *BackTranslation* com gráficos de treinamentos, melhorias gerais de desempenho, como observadas nos valores de acurácia 94,43% e 92,14% para o dataset GA aumentado, e exemplos da diversificação textual produzida. Dessa forma, afirma-se que as técnicas aqui descritas podem ajudar pesquisadores em PLN limitados a pequenos conjuntos de dados.

Para trabalhos futuros, pode-se estudar a abrangência e eficácia das mesmas técnicas e bibliotecas vistas neste trabalho em textos na língua Portuguesa, ou o efeito das etapas de pré-processamento de texto no desempenho final dos modelos de PLN.

Agradecimentos

Os autores agradecem ao apoio da Universidade Federal do Recôncavo da Bahia.

References

[1] D. Jurafsky and J. H. Martin, “Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition,” *Stanford University*, 2006.

[2] S. Barbosa and B. da Silva, *Interação Humano-Computador*, ser. Editora Campus. Elsevier, 2010.

[3] M. B. Pasinato, C. E. Mello, and G. Z. ao, “Acompanhamento de campanha eleitoral pelo twitter,” in *Anais do 12 Congresso Brasileiro de Inteligência Computacional*, C. J. A. Bastos Filho, A. R. Pozo, and H. S. Lopes, Eds. Curitiba, PR: ABRICOM, 2015, pp. 1–6.

[4] W. Y. Wang and D. Yang, "That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using #petpeeve tweets," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 2557–2563.

[5] C. G. Ramon Abilio and V. Fernandes., "Data mining applied on web robots detection: A systematic mapping," in *Anais do 15 Congresso Brasileiro de Inteligência Computacional*, C. J. A. B. Filho, H. V. Siqueira, D. D. Ferreira, D. W. Bertol, and R. C. L. ao de Oliveira, Eds. Joinville, SC: SBIC, 2021, pp. 1–8.

[6] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[7] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901.

[8] C. Shorten, T. Khoshgoftaar, and B. Furht, "Text data augmentation for deep learning," *Journal of Big Data*, vol. 8, 07 2021.

[9] S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, and E. Hovy, "A survey of data augmentation approaches for NLP," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, Aug. 2021, pp. 968–988.

[10] J. Wei and K. Zou, "Eda: Easy data augmentation techniques for boosting performance on text classification tasks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, vol. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Association for Computational Linguistics, 2019, pp. 6382–6388.

[11] X. Lu, B. Zheng, A. Velivelli, and C. Zhai, "Enhancing Text Categorization with Semantic-enriched Representation and Training Data Augmentation," *Journal of the American Medical Informatics Association*, vol. 13, no. 5, pp. 526–535, 09 2006. [Online]. Available: https://doi.org/10.1197/jamia.M2051

[12] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[13] A. L. C. Ottoni, R. M. de Amorim, M. S. Novo, and D. B. Costa, "Tuning of data augmentation hyperparameters in deep learning to building construction image classification with small datasets," *International Journal of Machine Learning and Cybernetics*, vol. 14, no. 1, pp. 171–186, 2023.

[14] P. Rici, S. O. S. Santos, and A. L. C. Ottoni, "Tuning of data augmentation hyperparameters to covid-19 detection in X-ray images with deep learning," *Learning & Nonlinear Models*, vol. 20, no. 2, pp. 5–20, 2022.

[15] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly, 2009. [Online]. Available: http://www.nltk.org/book

[16] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts: The MIT Press, 1999.

[17] A. Thakur, *Approaching (Almost) Any Machine Learning Problem*. Abhishek Thakur, 2020.

[18] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *The Visual Computer*, vol. 521, pp. 436–444, 2015.

[19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[20] I. N. d. Silva, D. H. Spatti, and R. A. Flauzino, *Redes neurais artificiais para engenharia e ciências aplicadas*. Artliber Editora, 2010.

[21] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[22] S. Bird and E. Loper, "NLTK: The natural language toolkit," in *Proceedings of the ACL Interactive Poster and Demonstration Sessions*. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 214–217.

[23] Z. Wu, "Backtranslation," 2021. [Online]. Available: https://github.com/hhhwwwuuu/BackTranslation

[24] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 86–96.

[25] J. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin, and Y. Qi, "Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 119–126.