CLaRiCe: Uma abordagem neural para a correção automática de redações

João Tavares[1], **Luiz Rodrigues**[1], **Diego Dermeval**[1,2]

[1]Núcleo de Excelência em Tecnologias Sociais - Universidade Federal de Alagoas
Maceió - AL - Brasil

[2]Graduate School of Education - Harvard University - USA

diego.dermeval@famed.ufal.br

Abstract. *Writing is an important skill that we acquire when starting our studies, being used on several occasions for acquisition, representation, evaluation of knowledge, as carried out in evaluation media such as ENEM, and entertainment as in the literary environment. Several previous works explored the automatic correction of essay-argumentative texts, but did not carry out an in-depth analysis and comparison of the use of neural models. Carrying out experiments with the Extended Essay-BR database demonstrated that convolutional models excel in the regression task, reaching an Mean Absolute Error that varies from 15.24 to 21.48 among the five skills, providing a model capable of performing a good simultaneous correction of 5 skills.*

Resumo. *Escrever é uma importante habilidade que adquirimos ao iniciar nossos estudos, sendo utilizada em diversas ocasiões para aquisição, representação, avaliação de conhecimentos, como realizado em meios avaliativos como o ENEM, e entretenimento como no meio literário. Diversos trabalhos anteriores realizaram uma exploração acerca de correção automática de textos dissertativos-argumentativos, mas não chegaram a realizar uma análise e comparação profundas acerca do uso de modelos neurais. A realização de experimentos com a base Extended Essay-BR demonstrou que os modelos convolucionais se sobressaem na tarefa de regressão, atingindo um Erro Absoluto Médio que varia de 15.24 a 21.48 dentre as cinco competências, proporcionando um modelo capaz de realizar uma boa correção simultânea das 5 competências.*

1. Introdução

Em diversos países é comum a utilização da escrita, sobretudo de textos dissertativos argumentativos, não só como avaliação do conhecimento mas como parte importante de exames de admissão em instituições de nível superior. No cenário brasileiro, sobretudo, o Exame Nacional do Ensino Médio *(ENEM)* se utiliza de 5 avaliações de diferentes áreas do conhecimento para a composição das notas dos candidatos, sendo um destes a redação, cuja nota é a soma das notas do candidato em cada uma das cinco competências avaliadas. Textos dissertativos argumentativos também são amplamente utilizados no contexto escolar para a composição de currículos e preparação para o ENEM, nesse contexto, esse tipo de texto pode ser utilizado para a geração de Analíticas de Aprendizado (*Learning Analytics*), como descrita em [Freitas et al. 2020], que podem ser utilizadas para entender

melhor os diferentes níveis de proficiência em escrita disponíveis em uma determinada sala de aula ou até mesmo numa população de candidatos a um exame admissional.

Ademais, a correção de redações é uma tarefa de extrema importância, além de ser extremamente trabalhosa, uma vez que a leitura deve ser feita por um agente de correção. Com isso, esse agente está sujeito à fadiga, seja esta visual ou intelectual, e passível de cometer erros, além de levar uma janela de tempo consideravelmente grande para a realização destas correções. Esses desafios podem acarretar em atrasos na disponibilização de notas e até mesmo na geração das Analíticas de Aprendizado, que poderiam auxiliar num melhor direcionamento acerca da política de ensino da instituição ou das instituições que dela se utilizem, tornando-se uma problemática para a verificação de tais estatísticas.

Entretanto, nos últimos anos observou-se um grande desenvolvimento e afloramento das pesquisas em aprendizado de máquina. Vários desses algoritmos de aprendizado de máquina se mostraram eficazes em lidar com problemas no cenário não só de geração de Métricas de Aprendizado, mas em tarefas como reconhecimento facial, classificação de textos e até mesmo gerando textos e imagens. Graças a esses resultados, profissionais da área e até mesmo autores têm sugerido a implantação de tais algoritmos no contexto da educação, principalmente no cenário do ensino gramatical e ortográfico, onde técnicas como as de *Processamento de Linguagem Natural(PLN)* podem auxiliar na automação de diversos processos, como visto em [Filho et al. 2023] e descrito em [Barbosa and Campelo 2020]. Nesse contexto, uma das principais possibilidades é o uso de *Redes Neurais Artificias(ANNs)*, que nos últimos anos se provaram extremamente eficazes em lidar com dados não estruturados, como texto.

Diante deste cenário, o objetivo deste artigo é avaliar a capacidade de modelos de redes neurais para a realização da tarefa de correção de redações do ENEM através da testagem de diferentes topologias e tipos de camadas de redes neurais. Apesar de diversas pesquisas semelhantes, como [Oliveira et al. 2022] e [Barbosa et al. 2022], estudos anteriores não realizaram uma comparação direta entre diversos modelos de *Aprendizagem Profunda*, principalmente no que tange à usabilidade de modelos de sequência e não chegaram a realizar uma abordagem de tentativa de *regressão* das 5 competências com um único modelo, explorando somente a tarefa de *regressão* considerando apenas uma única competência, além de aplicarem técnicas mais básicas de pré-processamento e vetorização de palavras como a Frequência do Termo - Inverso da Frequência no Documento*(TF-IDF)*. Assim, este artigo expande a literatura ao realizar uma comparação entre diferentes tipologias de Redes Neurais Artificiais, utilizando técnicas de pré-processamento e vetorização mais robustas, além de propôr uma abordagem de um modelo capaz de realizar a correção das 5 competências simultaneamente.

Os resultados dos experimentos realizados demonstraram que o modelo de *Convolução 1D* se sobressaiu à modelos de recorrência na tarefa de correção, demonstrando os melhores resultados para o Erro Absoluto Médio*(MAE)* e para a Raiz do Erro Quadrático Médio*(RMSE)* nas 5 competências. Além disso, resultados adicionais obtidos após a otimização dos *hiperparâmetros* sugerem que modelos mais profundos podem obter uma capacidade mais elevada de obter um alto desempenho nesta tarefa, desde que acrescidos de um determinado valor de regularização.

2. Fundamentação Teórica

Esta seção apresenta uma breve introdução ao *Aprendizado de Máquina*, *Aprendizado Profundo* e ao *Processamento de Linguagem Natural(PLN*, descrevendo brevemente o funcionamento de algumas técnicas aqui empregadas.

2.1. Aprendizado de Máquina e Aprendizado Profundo

O *Aprendizado de Máquina(AM)* consiste na aplicação de diversos algoritmos, conhecidos como *algoritmos de aprendizado*, que se utilizam de cálculo, álgebra linear e estatística para gerar uma representação matemática de um determinado problema, com o intuito de aproximar uma função desconhecida a partir dos seus dados. Suas principais abordagens consistem em:

1. Aprendizado Supervisionado: Tarefas de Classificação e Regressão, tarefas em que há rótulos disponíveis.
2. Aprendizado Não-Supervisionado: Tarefas de Agrupamento, Extração de Regras e Identificação de Padrões.
3. Aprendizado Por Reforço: Tarefas em que dados não estão disponíveis, devendo o agente aprender por meio de sua interação com o ambiente e um sistema de recompensas e punições por meio destas interações.

O *Aprendizado Profundo(AP)* diferencia-se do *Aprendizado de Máquina* clássico por se tratar apenas da utilização de *Redes Neurais Artificiais Profundas(DNNs)*, de forma que diferentes arquiteturas de *DNN* são construídas para a resolução do problema, podendo ser estas divididas em:

- Redes Neurais Convolucionais(CNNs)
- Redes Neurais Recorrentes(RNNs) e suas variantes como Long Short-Term Memory(LSTM) e Gated Recurrent Unit(GRU)
- Autoencoder, Variational Autoencoder
- Redes Neurais Generativas Adversariais(GANs)
- Transformadores(Transformers)

Outra forma de se aplicar AM ou AP é a partir da *Aprendizagem de Multiplas Tarefas(Multi-Task Learning)*, no qual um mesmo modelo é treinado em diferentes tarefas de Aprendizado ao mesmo tempo.

2.2. Processamento de Linguagem Natural(PLN)

O *PLN* trata-se de um subcampo da *Inteligência Artificial(IA)* responsável por lidar com problemáticas referentes à realização de aprendizado utilizando linguagem natural, ou seja, a realização de tarefas de AM utilizando dados de texto e até mesmo áudio para a classificação, regressão, agrupamento, etc em dados dispostos nesses formatos.

Diversas técnicas de *PLN* são utilizadas em diferentes cenários, porém o fluxo comum do treinamento de algoritmos nesse contexto segue a seguinte ordem:

1. Remoção de palavras que não possuem tanto impacto semântico ou são muito frequentes, como ”a”e ”o”, conhecidas como *Stopwords*.

2. Normalização das palavras por meio de técnicas como *Lemmatização* e *Stemming*, que buscam reduzir as palavras de uma base de dados de texto(corpus) em raízes comuns, ao contrário da lemmatização o stemming é mais rápido, porém não garante que a raíz extraída existe.
3. Dividir as frases em palavras isoladas de acordo com um critério de separação, criação de *Tokens*.
4. Converter as palavras em números que possam ser interpretados pelos algoritmos, conhecidos como*Vetores de Palavras(Word Embbedings)*, sendo poderosa por ser capaz de proporcionar uma melhor representação semântica das palavras, uma vez que palavras similares ficam próximas no espaço.
5. Treinamento de um algoritmo de AM.

3. Trabalhos Relacionados

As abordagens de Aprendizado de Máquina têm se consolidado como solução para diversos problemas de pesquisa, sobretudo no cenário educacional brasileiro. Nesse contexto, esta seção apresenta artigos relacionados que também utilizaram técnicas de *PLN* para automatizar a avaliação de redações do ENEM.

Em [Filho et al. 2023]os autores realizaram um ensaio acerca da utilização de algoritmos clássicos de AM para a geração de *Analíticas de Aprendizagem* de textos narrativos, realizando comparações de menos de 10 modelos clássicos, além de descreverem métricas e métodos para extração de características suplementares às tarefas de aprendizado de máquina. Dentre as limitações deste artigo têm-se a não testagem de qualquer modelo baseado em DNNs, tendo explorado somente modelos clássicos sem a utilização de representações neurais ou *modelos de linguagem neural* para a representação das palavras.

Em [Oliveira et al. 2022] os autores apresentam questionamentos e possibilidades para o emprego de algoritmos de aprendizado para a correção de redações em exames de admissão, como o ENEM, além de apresentarem um impasse para a melhora do desempenho de modelos de aprendizado nesse sentido que consiste no desbalanceamento da base de dados, além da pouca disponibilidade de dados para um melhor refinamento desses modelos, simbolizando um problema principalmente para modelos neurais. Entretanto, neste artigo os autores trataram apenas da pontuação referente a uma única competência dentre as 5 empregadas pelo *ENEM*, de forma que não foram avaliadas as correlações dos modelos testados com as demais 4 notas que compõem a nota final da redação do exame, e, apesar da utilização de um modelo básico de rede neural, não realizaram a utilização de um modelo de transformador para a extração dos vetores de palavras, o que poderia providenciar uma melhor representação semântica, com a possibilidade de melhorar os resultados da competência 4, referente à coesão textual.

Em [Rosa and Mello 2022] os autores propõem a utilização do gênero textual para a tarefa de classificação, o que pode ser um indicativo de boa coesão textual. Ademais, os mesmos evidenciam ainda que a utilização de Aprendizagem de Máquina para a correção e análise de coesão textual e de demais características textuais já foram empregadas em outras línguas, afirmando ainda que no caso particular da língua portuguesa essa área ainda estaria dando seus passos iniciais.

Em resumo, a revisão dos trabalhos relacionados mostra que por mais que modelos

neurais tenham sido testados, os mesmos foram testados consideram ténicas clássicas de vetorização, de forma que uma forma mais sofisticada como a utilização de modelos pré-treinados somente para a etapa de extração de características e geração dos vetores de palavras, além disso modelos mais profundos e com diferentes tipos de camadas como CNNs e RNNs não chegaram a ser avaliados para verificar o seu comportamento no que tange à tarefa de regressão. O mesmo cenário ocorreu com *Aprendizagem de Múltiplas Tarefas*, que não chegou a ser testada, uma vez que os trabalhos focaram em avaliar a geração de novas características ou focaram seus esforços na avaliação de somente uma das 5 competências presentes na nota final.

O presente artigo contribui para a literatura ao se utilizar não somente de um modelo pré-treinado para extração dos vetores de palavras, mas também na avaliação de diferentes modelos, utilizando os diferentes tipos de ANN, que são capazes de realizar a correção das 5 competências de um único texto, o que representa um passo importante uma vez que os modelos anteriores não realizavam a correção das 5 competências com um único texto.

4. Método

Esta seção apresenta o método do presente artigo, descrevendo a base de dados utilizada, bem como o processo de preprocessamento dos dados, seleção e avaliação dos modelos.

4.1. Base de dados

A base de dados empregada no uso deste estudo foi a base de dados *Extended Essay-Br*, descrita em [Marinho et al. 2021], contendo até a data deste estudo um total de 6578 redações corrigidas por profissionais humanos de acordo com as competências do ENEM. Essas bases de dados dispõe das notas de cada aluno em cada uma das 5 competências, além de dispôr da nota final total do mesmo, conforme detalhado na Tabela 1.

Tabela 1. Estatísticas das notas da base

#	Competência	Média	Desvio Padrão
1	norma-padrão da língua portuguesa	133,25	33,17
2	gênero, tema e repertório	132,75	39,59
3	autoria	115,56	38,17
4	coesão textual	134,99	45,38
5	proposta de intervenção	115,71	51,79
	Nota Total	632,25	179,81

4.2. Preparação dos Dados

Para atingir o objetivo deste artigo, foi recorrido ao teste e avaliação de diferentes configurações de modelos neurais. A primeira etapa para o treinamento dos modelos foi a execução de processos de limpeza no *corpus*, dessa forma foram aplicadas as seguintes etapas de pré-processamento:

1. Remoção de stopwords.
2. Remoção de pontuação e acentuação.

3. Normalização dos textos por meio de Lemmatização.

Tais técnicas foram empregadas com o intuito de tornar o texto presente na base o mais limpo possível, proporcionando aos modelos neurais uma melhor extração do conhecimento ali presente e melhor desempenho final na tarefa de regressão. Por exemplo, a remoção de stopwords, pontução e acentuação foram empregadas com o intuito de reduzir a quantidade de palavras que não apresentam tanta importância semântica, uma vez que estamos utilizando um modelo baseado em atenção, o BERTimbau, para a extração das características e as muitas ocorrências dessas palavras podem fazer com que o algoritmo atribua maior atenção as mesmas.

Quanto a utilização de Lemmatização, ocorreu-se a partir de uma necessidade de tornar o texto padronizado de forma que o BERTimbau tivesse um melhor desempenho na extração dessas características, além de gerar uma padronização que carrega uma melhor informação semântica e contextual. O emprego de tais técnicas é notado também em outros trabalhos como em [Filho et al. 2023], [Barbosa and Campelo 2020] e [Oliveira et al. 2022], sendo utilizados para a contabilização de ocorrências de palavras, contabilização de características e a utilização das formas base(lemmas) para a análise de melhoria de desempenho em tarefas de correção de textos.

Para a extração das características foi utilizado um modelo pré treinado BERT adaptado para a língua portuguesa, o BERTimbau descrito em [Souza et al. 2020], realizando a extração de vetores de palavras (Word Embeddings) para alimentar as camadas posteriores nas topologias de Redes Neurais avaliadas. A utilização do BERTimbau foi feita para que não fosse necessário o retreinamento dos modelos a serem testados na tarefa de representação e modelagem da língua portuguesa, proporcionando um maior foco no treinamento e ajustes de hiperparâmetros dos regressores.

4.3. Seleção de Modelos

A seleção dos modelos se deu através do treinamento dos modelos na base de dados anteriormente citada com uma divisão de 75/10/15 em treino, validação e teste, realizada após a limpeza dos textos da base. Para definição do melhor modelo, foram realizados testes usando camadas LSTM, GRU, Convoluções 1D, camadas Feed-Forward e camadas LSTM e GRU bidirecionais.

Também foram realizadas alterações na forma que se era realizada a tokenização e geração da sequência de palavras para vetorização pelo BERTimbau, sendo testada inicialmente uma sequência de tamanho de 64 tokens e posteriormente uma sequência de 128 tokens, utilizando como base as métricas do dataset disponibilizadas em [Marinho et al. 2021], demonstrando as métricas de média de tokens por texto, tokens por parágrafo e tokens em cada sentença, além de realizar essa variação para verificar se a utilização de sequências de entrada maiores poderia proporcionar uma melhora no desempenho final dos modelos testados.

Ademais, também foi utilizada a biblioteca *optuna*[1] para realizar ajustes de hiperparâmetros nos modelos e selecionar aqueles que apresentaram o melhor desempenho. O *Optuna* é um framework de otimização de hiperparâmetros, apresentado e descrito em [Akiba et al. 2019], que possui implementações eficientes para diversos tipos de métodos

[1]https://optuna.readthedocs.io/en/stable/

de otimização de hiperparâmetros, como *Grid Search*, *Random Search* e *Otimização Bayesiana*, ver [Mockus et al. 1978], sendo o método de otimização aqui utilizado o *Tree-Structured Parzen Estimator*, um método de Otimização Bayeasiana descrito em [Watanabe 2023].

Devido a natureza computacionalmente intensiva das redes neurais, a primeira etapa do experimento foi verificar a integração de um modelo pré-treinado com as camadas subsequentes para que não fosse necessário recorrer ao treinamento de um modelo que tivesse que aprender o vocabulário antes de poder ser utilizado para testagem.

Em seguida foram realizados alguns testes iniciais com a utilização de um modelo com 5 saídas de regressão, para verificar a viabilidade do aprendizado e o desempenho do modelo num momento inicial, e, por ser tratar de um modelo de aprendizado profundo com 5 saídas de regressão independentes. Fez-se necessário o uso do *Optuna* para a otimização dos hiper parâmetros da rede, na tentativa de se obter uma melhor configuração do modelo e poupar tempo de treinamento com a otimização de tais parâmetros que pode se fazer exaustiva.

Dentre os parâmetros utilizados como espaço de busca, foram utilizadas os seguintes valores:

1. Taxa de dropout: Variando de 0.1 até 0.5
2. Regularização L2: Variando de 1e-10 até 1e-1
3. Tipos de camada: Feed-Forward, Convolução 1D, LSTM, GRU e camadas LSTM ou GRU Bidirecionais.
4. Número camadas: Variando de 1 até 3
5. Número de filtros convolucionais: Variando de 16 até 64
6. Número de células recorrentes: de 8 até 32
7. Otimizadores: Adam, RMSProp e SGD
8. Learning rate: Variando de 1e-3 até 1e-1
9. Momentum: Variando de 1e-3 até 1e-1

Além disso, também foi configurada uma função de callback personalizada para realizar o EarlyStopping, uma vez que o Early Stop padrão do Keras só consegue se atentar a uma métrica por vez, nesta função personalizada o Early Stopping era realizado baseando-se na mediana dos 5 *Root Mean Squared Error(RMSE's)* da rede no conjunto de validação, devido ao desbalanceamento da base nas notas das competências fornecidas, juntamente com o *Mean Absolute Error(MAE)* como função de perda. Juntamente com a função de Early Stop também foi definido o valor de *patience=3*, pois em testes iniciais do experimento observou-se que com o valor utilizado neste parâmetro sendo 7, alguns modelos atingiam uma etapa de treinamento sem nenhuma alteração significativa nas métricas avaliadas.

5. Resultados e Discussão

Ao longo do trabalho desenvolvido, foram executados testes ao todo em 40 modelos diferentes se utilizando do espaço de busca acima citado, sendo os 5 melhores modelos os seguintes:

- Convolução 1D
- LSTM Bidirecional

- 2 modelos GRU Bidirecional
- Feed-Forward

Para fins de completude, a tabela com os resultados de todos os 40 modelos pode ser encontrada em nosso material suplementar[2].

Dentre os modelos acima, com exceção do modelo convolucional, os demais apresentaram métricas medianas das 5 competências para o RMSE em torno de 35,4 a 35,91, enquanto os 5 modelos que obtiveram os piores valores nessa métrica tiveram valores observados em torno de 107,03 a 126,44, sendo o modelo com pior desempenho também um modelo convolucional.

Apesar de se obter aumento no tempo de treino com o aumento do tamanho da sequência, foi perceptível que esse aumento proporcionou um benefício significativo no desempenho final do modelo, ficando evidente nos valores do MAE, que apresentaram uma convergência melhor na segunda rodada de otimização através do ***Optuna***, conseguindo atingir os valores que foram verificados no modelo convolucional.

A tabela 2 apresenta os valores atingidos para o MAE em cada competência:

Tabela 2. MAE de cada competência nos modelos

#	Modelo	Comp. 1	Comp. 2	Comp. 3	Comp. 4	Comp. 5
1	Convolução 1D	15,24	16,34	17,32	19,8	21,48
2	LSTM Bidirecional	14,69	22,51	26,24	27,84	33,28
3	GRU Bidirecional	23,05	24,81	25,57	25,54	33,39
4	GRU Bidirecional 2	18,72	24,17	24,36	27,39	30,74
5	Feed-Forward	21,29	26,24	26,69	28,87	32,88

Apesar de o modelo convolucional ter se sobressaído quanto aos demais modelos, é importante se observar que todos os 5 modelos atingiram métricas relativamente próximas em cada uma das competências analisadas, obtendo-se uma maior discrepância na competência 5, onde o modelo convolucional conseguiu obter resultados com pouco mais de 10 pontos de diferença em relação ao segundo melhor modelo avaliado.

A tabela 3 apresenta os valores atingidos para o RMSE em cada competência:

Tabela 3. RMSE de cada competência nos modelos

#	Modelo	Comp. 1	Comp. 2	Comp. 3	Comp. 4	Comp. 5
1	Convolução 1D	22,71	23,89	24,38	27,56	31,03
2	LSTM Bidirecional	27,88	33,92	35,40	38,45	46,06
3	GRU Bidirecional	31,28	34,90	37,71	35,58	43,60
4	GRU Bidirecional 2	29,97	33,36	35,82	38,58	41,14
5	Feed-Forward	29,48	35,31	35,92	38,61	43,77

[2]https://osf.io/kqdbc/?view$_onl y = 005598998a074f31bda022102a913e66$

O resultado apresentado pelo modelo convolucional mais uma vez foi superior em relação aos demais, conseguindo manter quase todas as suas métricas em torno de 20, o que implica que modelos convolucionais utilizados em conjunto com transformadores para a extração de características possam ser uma abordagem ideal para a representação das notas em contextos textuais, dado que as notas no contexto do ENEM para cada competência da redação tendem a ser incrementadas ou decrescidas de 20 em 20 pontos, enquanto que nosso modelo alcançou, em média um erro de 25.91 no ***RMSE*** e conseguiu atingir um erro médio de 18.04 no ***MAE***.

Abaixo estão os gráficos de convergência do MAE dos 2 modelos com melhor desempenho:

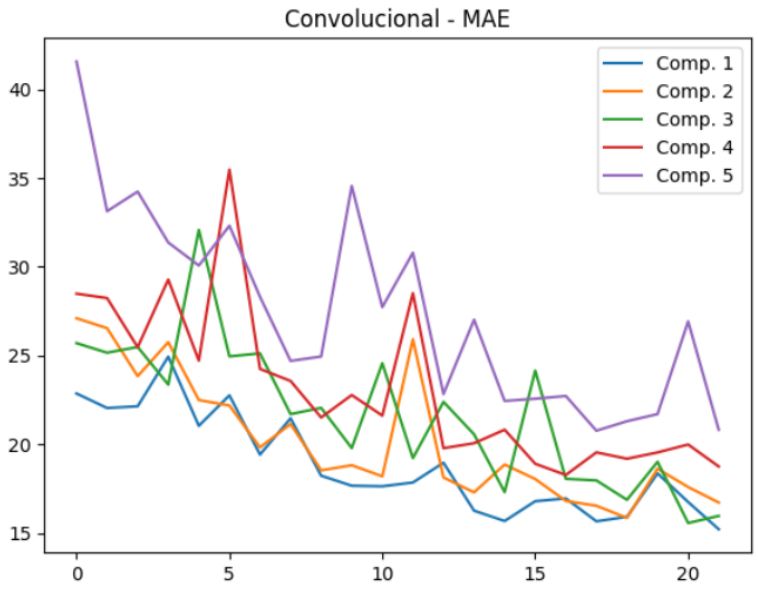

Figura 1. Gráfico de convergência do modelo convolucional

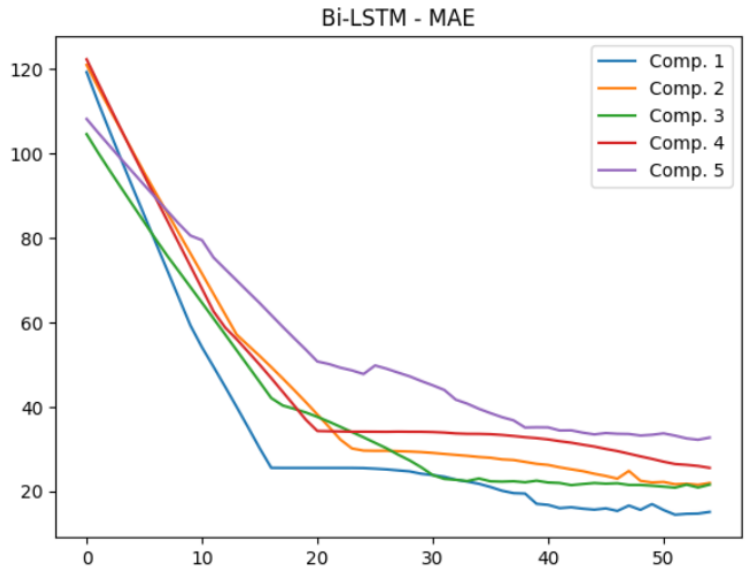

Figura 2. Gráfico de convergência do modelo recorrente LSTM Bidirecional

Os dois gráficos apresentaram uma disparidade na curva de aprendizado. Enquanto o gráfico do modelo convolucional apesar de possuir mais flutuações na curva de perda conseguiu obter resultados mais satisfatórios, o modelo recorrente apresenta um gráfico típico que pode simbolizar uma boa convergência do modelo. No entanto, ele não foi capaz de obter valores menores para a métrica utilizada como perda; uma suposição plausível é de que, com alguns ajustes nos hiperparâmetros, e com mais tempo de treinamento, o modelo recorrente possa atingir resultados ainda mais satisfatórios que o modelo convolucional.

Por fim, as tabelas 4 e 5 apresentam respectivamente os hiperparâmetros obtidos pelas duas melhores configurações convolucional e recorrente:

Tabela 4. Hiperparâmetros dos regressores do modelo convolucional

Parâmetro	Comp. 1	Comp. 2	Comp. 3	Comp. 4	Comp. 5
Total de camadas	7	5	5	7	5
Filtros/convolução	28/39	22	53	51/21	28
Neurônios na camada densa	90	20	126	39	123
Otimizador	Adam				
Learning Rate	7,06e-03				
Dropout rate	3,03e-01				
Weight Decay(L2)	6,61e-04				

Tabela 5. Hiperparâmetros dos regressores do modelo recorrente

Parâmetro	Comp. 1	Comp. 2	Comp. 3	Comp. 4	Comp. 5
Total de camadas	3	2	2	1	2
Unidades/Célula	10/26/21	26/16	22/13	18	29/11
Otimizador	Adam				
Learning Rate	1,01e-03				
Dropout rate	1,64e-01				
Weight Decay(L2)	4,67e-06				

Apesar de serem modelos que possuem abordagens diferentes, a diferença se dá em duas principais características:

1. O modelo recorrente não apresenta uma camada densa nem camada de achatamento (*flatten*) antes de enviar os resultados de suas iterações para a camada de saída do regressor.
2. O número total de camadas do modelo convolucional foi maior, sobretudo nas competências 1 e 4, o que pode significar que um modelo recorrente que possua uma abordagem com maior profundidade na rede, além de acrescentar uma última camada densa possa vir a beneficiar os resultados finais das notas de cada competência.

Também é importante destacar que em todas as camadas foi aplicada a regularização L2, além de possuir camadas de dropout entre cada camada da rede para melhorar o desempenho de generalização das camadas intermediárias da rede.

Em resumo, os resultados mostram o potencial das redes neurais baseadas em embeddings para avaliar as cinco competências do ENEM. Pesquisas anteriores usaram outros formatos de vetorização, como TF-IDF, e se utilizaram, também, de características artificiais para modelos de regressão, focando em uma única competência ou na relação e impacto das características criadas artificialmente com uma competência específica ou com a nota final dada atribuída pela modelo. Neste artigo, exploramos o uso de embeddings gerados por um transformador pré-treinado e a construção de um modelo capaz de corrigir simultaneamente as cinco competências da redação do ENEM. Demonstramos que os modelos baseados em ANN, utilizando embeddings, oferecem vantagens por carregar mais informações semânticas. Esta abordagem é relevante para a criação de ferramentas de correção automática de textos, beneficiando a geração de Analíticas de Aprendizado e auxiliando no aprendizado por meio de correções mais rápidas e contextualizadas, além de proporcionar uma possibilidade para a identificação de pontos de melhoria no ensino da escrita de textos neste formato.

6. Considerações Finais e Trabalhos Futuros

No presente trabalho foi realizada uma investigação acerca de abordagens de aprendizado profundo como uma técnica que possa ser utilizada no desenvolvimento de ferramentas com a finalidade e auxiliar no processo de correção de textos. É perceptível que abordagens baseadas em PLN para a realização de correção de métodos avaliativos baseados em textos discursivos se mostra com um grande potencial de desenvolvimento, como vem sendo explorado no atual cenário de pesquisa brasileiro, e agora com a popularização de tecnologias baseadas em inteligência artificial surge uma tendência natural para que isso seja cada vez mais utilizado e aplicado em diversos contextos do nosso cotidiado.

Apesar de abordagens tradicionais de Aprendizado de Máquina se mostrarem promissoras e obterem um desempenho satisfatório nessa problemática, pudemos notar, a partir dos experimentos realizados no presente estudo, que uma abordagem neural parece ser um caminho natural a ser seguido para tratar de questões que envolvam dados textuais, uma vez que, como supracitado, modelos de aprendizado profundo tendem a obter um desempenho superior aos modelos tradicionais sobretudo no quesito de dados não estruturados como imagens e textos. No entanto, é necessário que uma quantidade considerável de dados seja utilizada para seu treinamento, de modo a torná-lo mais sofisticado e refinado e, assim, capaz de apoiar a tomada de decisão humana não só no atual modelo de correção, cujos critérios se baseiam no tipo dissertativo-argumentativo, mas também em outros gêneros e tipos textuais que possam ser utilizados como critérios de seleção e avaliação.

Ademais, os resultados obtidos no presente trabalho evidenciam que a utilização de modelos neurais se mostra como uma abordagem bastante promissora neste caso de uso, uma vez que a correção de redações no contexto do ENEM é realizada por dois avaliadores diferentes, sendo a nota final uma média aritmética dentre esses dois avaliadores, em caso de uma disparidade maior que 80 pontos entre esses dois avaliadores a correção é feita por um terceiro avaliador, sendo a nota final uma média aritmética entre os 3 ava-

liadores. No contexto deste trabalho, como supracitado e apresentado anteriormente, os modelos aqui demonstrados foram capazes de realizar uma tarefa de regressão para cada uma das competências de forma que os mesmos obtiveram uma taxa de erro bastante satisfatória e que poderia ser empregada para um possível teste de correção com avaliações de alunos em preparação para o exame nacional do ensino médio conjuntamente considerando as notas dadas por avaliadores humanos para verificar as métricas de desempenho do modelo.

No entanto, ainda existem alguns fatores limitantes na ampla utilização e implantação desta abordagem em uma ferramenta para gerar tais métricas, como a quantidade de dados disponíveis para realizar o treinamento desses modelos, uma vez que o verdadeiro desempenho de modelos tende a ficar cada vez mais refinado quando se utiliza uma quantidade maior de dados para o treinamento, o que representa um fator limitante no cenário atual uma vez que o conjunto de dados aqui utilizado, Extended Essay-BR apresenta uma quantidade de apenas 6577 redações, com essa quantidade de dados os modelos aqui demonstrados foram capazes de obter métricas satisfatórias, mas espera-se que com uma maior quantidade de dados no futuro seja possível verificar novamente o desempenho desses modelos para a utilização neste cenário.

Além disso, a criação de ferramentas que se utilizam de Redes Neurais atualmente são mais facilmente implementadas e disponibilizadas em ambientes de nuvem devido à natureza computacionalmente intensiva de redes neurais citada anteriormente, exigindo que a ferramenta que por ventura venha a ser utilizada necessite de uma conexão ativa com a internet para poder realizar a inferência da nota de textos a serem avaliados, ocasionando dificuldade de implantação de tal tecnologia em cenários mais remotos do mundo em que a conectividade à rede acaba sendo de baixa qualidade ou onde muitas vezes inexiste, dificultando o acesso da população local a tais avanços tecnológicos, além de possíveis questões acerca da privacidade de dados uma vez que os dados enviados podem possuir um certo grau de confidencialidade, principalmente em casos de aplicação e utilização em eventos de larga escala como o Exame Nacional do Ensino Médio, cujos dados são extremamente confidênciais antes, durante e após aplicação das provas e apuração dos resultados.

Por isso, como trabalhos futuros, vislumbram-se:

1. Enriquecimento de Bases de Dados Textuais: Melhoria das bases de dados com corpora textuais para criar soluções mais eficientes e integradas, facilitando a avaliação de textos além dos dissertativo-argumentativos e agilizando exames que utilizam este método.
2. Investigação de Aprendizado de Ranqueamento: Transformação de problemas de regressão em problemas de classificação para melhorar o ranqueamento de textos com múltiplas métricas, avaliando se modelos neurais oferecem melhores abordagens em textos narrativos.
3. Aplicação de Aprendizagem de Máquina em Sistemas Embarcados: Exploração do uso de modelos profundos em dispositivos de baixo custo para melhorar a assertividade e o desempenho em cenários com baixa conectividade, visando a aplicação de ferramentas de aprendizagem profunda.

Referências

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Barbosa, A. and Campelo, C. (2020). Processamento de linguagem natural em artefatos textuais educacionais: Um mapeamento sistemático no contexto brasileiro. In *Anais do XXXI Simpósio Brasileiro de Informática na Educação*, pages 1433–1442, Porto Alegre, RS, Brasil. SBC.

Barbosa, G., Batista, H., Miranda, P., Santos, J., Isotani, S., Cordeiro, T., Bittencourt, I. I., and Mello, R. F. (2022). Aprendizagem de máquina para classificação de tipos textuais: Estudo de caso em textos escritos em português brasileiro. In *Anais do XXXIII Simpósio Brasileiro de Informática na Educação*, pages 920–931, Porto Alegre, RS, Brasil. SBC.

Filho, M. S., Nascimento, A., Miranda, P., Rodrigues, L., Cordeiro, T., Isotani, S., Bittencourt, I., and Mello, R. (2023). Automated formal register scoring of student narrative essays written in portuguese. In *Anais do II Workshop de Aplicações Práticas de Learning Analytics em Instituições de Ensino no Brasil*, pages 1–11, Porto Alegre, RS, Brasil. SBC.

Freitas, E., Falcão, T. P., and Mello, R. F. (2020). Desmistificando a adoção de learning analytics: um guia conciso sobre ferramentas e instrumentos. *Sociedade Brasileira de Computação*.

Marinho, J., Anchiêta, R., and Moura, R. (2021). Essay-br: a brazilian corpus of essays. In *Anais do III Dataset Showcase Workshop*, pages 53–64, Porto Alegre, RS, Brasil. SBC.

Mockus, J., Tiesis, V., and Zilinskas, A. (1978). The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2(117-129):2.

Oliveira, H., Miranda, P., Isotani, S., Santos, J., Cordeiro, T., Bittencourt, I. I., and Mello, R. F. (2022). Estimando coesão textual em redações no contexto do enem utilizando modelos de aprendizado de máquina. In *Anais do XXXIII Simpósio Brasileiro de Informática na Educação*, pages 883–894, Porto Alegre, RS, Brasil. SBC.

Rosa, B. A. and Mello, R. F. (2022). Análise automatizada de coesão em redações do ensino fundamental por meio de técnicas de processamento de linguagem natural. In *Anais Estendidos do XI Congresso Brasileiro de Informática na Educação*, pages 144–149, Porto Alegre, RS, Brasil. SBC.

Souza, F., Nogueira, R., and Lotufo, R. (2020). BERTimbau: pretrained BERT models for Brazilian Portuguese. In *9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23 (to appear)*.

Watanabe, S. (2023). Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance.