Automated Essay Scoring (AES) para o modelo ENEM

Cinthia Costa, Laura Seto, Leonardo Oliveira, Raul Komai, Renan Almeida
CCGT - Departamento de Computação
Universidade Federal de São Carlos
Sorocaba, Brasil
{ *ccosta, laura.naomi, leonardooliveirapedro, raulkomai, renan.almeida* } *@estudante.ufscar.br*

I. Introdução

O projeto explora a tarefa de pontuação automatizada de redações, *Automated Essay Scoring* (AES), alinhada aos critérios estabelecidos para o Exame Nacional do Ensino Médio (ENEM) pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP), inferindo uma nota para a produção textual do aluno como em uma avaliação humana profissional.

No ENEM, as redações recebem uma nota final entre 0 e 1000, calculada pela média da avaliação de dois corretores. Cada corretor aponta a nota como a somatória das pontuações, que variam de 0 a 200, obtidas em cinco competências distintas [1]. Como cada uma das competências aborda diferentes aspectos do texto e reflete problemas particulares de PLN, diversas técnicas podem ser exploradas de forma modular neste trabalho.

- **Competência I:** Demonstrar domínio da modalidade escrita formal da língua portuguesa.
- **Competência II:** Compreender a proposta de redação e aplicar conceitos das várias áreas de conhecimento para desenvolver o tema dentro dos limites estruturais do texto dissertativo-argumentativo em prosa.
- **Competência III:** Selecionar, relacionar, organizar e interpretar informações, fatos, opiniões e argumentos em defesa de um ponto de vista.
- **Competência IV:** Demonstrar conhecimento dos mecanismos linguísticos necessários para a construção da argumentação.
- **Competência V:** Elaborar proposta de intervenção para o problema abordado, respeitando os direitos humanos.

Em suma, o objetivo do projeto é avaliar um texto de entrada sob as regras determinadas pelo INEP para o modelo ENEM e gerar uma nota composta pelo grau de atendimento às competências mencionadas, sendo que este último deve ser mensurado utilizando técnicas de PLN.

II. Base de dados

A base de dados Essay-BR [2], considerada para o projeto, é um conjunto de 4.570 redações dissertativas-argumentativas produzidas por alunos brasileiros do ensino médio sobre 86 propostas diferentes. As amostras, coletadas entre dezembro de 2015 e abril de 2020 nos *websites* públicos Vestibular UOL e Educação UOL, são anotadas por especialistas humanos e normalizadas para aderir ao sistema de pontos do ENEM.

Cada amostra possui os atributos: *Prompt* - descrição do tema e um conjunto de textos motivadores; Score - nota final total da produção; *Title* - título da redação (pode ser nulo); *Essay Content* - a redação como uma lista de parágrafos; e C1, C2, C3, C4 e C5 - pontuações referentes às competências I, II, III, IV e V, respectivamente. Ademais, o corpus é dividido em amostras de treino (70%), validação (15%) e teste (15%) de forma a obter uma distribuição justa do atributo *Score* entre os conjuntos.

III. Experimentos e resultados

Como cada competência avalia características distintas da redação, elas são abordadas separadamente nas subseções a seguir.

A. Competência I

A Competência I envolve o domínio do candidato da linguagem formal da Língua Portuguesa. O modelo deve verificar se o texto recebido e dar uma nota de acordo com a adequação à Língua Portuguesa formal a partir de uma análise ortográfica e gramatical do texto.

1) Análise ortográfica: O primeiro aspecto analisado foi a ortografia do texto, isto é, se o texto possui ou não erros na escrita das palavras. Para isso, foi utilizado um algoritmo de similaridade para calcular a semelhança entre as palavras do texto com um dicionário de palavras. A biblioteca *spellchecker* configurada para português foi utilizada para esse fim.

2) Análise gramaticla: O segundo aspecto a ser avaliado é a gramática do texto. Isso pode ser verificado utilizando um modelo baseado em regras, utilizando a biblioteca *language_tool_python* também configurada para o português.

3) Pontuação c1: Para o cálculo da pontuação, inicialmente foi calculada a média aritmética entre as duas análises. Em seguida, foi feita uma normalização das médias para resultar nas notas utilizadas no ENEM. Graças a essa estratégia, a precisão atingida no corpus adotado foi de quase 37,5

B. Competência II

1) Adequação do texto ao gênero dissertativo-argumentativo: Para esta tarefa, que é uma classificação de gênero textual, focamos na identificação de características textuais que pudessem ser extraídas e utilizadas, posteriormente, como *features* na execução de um algoritmo simples de classificação.

Primeiramente, criamos um atributo alvo binário que indica a adequação ao gênero (1) e não adequação ao gênero (0), sendo que redações com pontuações c2 maior ou igual à 160 foram rotuladas com o primeiro e as redações com pontuações inferiores à 160 com o último. Com a intenção de auxiliar o classificador, extraímos atributos como o número de parágrafos por redação e o tamanho médio dos parágrafos por redação, que apresentavam uma correlação de 0.0305 e 0.282 com o atributo alvo, respectivamente.

Como modelo de classificação, treinamos de forma inicial uma Regressão Logística para obter não somente o rótulo binário, mas também a probabilidade da inferência, a fim de utilizá-la como uma *feature* no cálculo da pontuação c2. Computando as probabilidades para o conjunto de treinamento, observamos uma correlação de 0.552 entre a probabilidade obtida e a pontuação c2.

2) Cobertura do recorte temático: A cobertura do tema, por sua vez, envolveu a identificação de palavras chaves da redação e do *prompt* com suas representações TF-IDF e o cálculo das *embeddings* dessas palavras com o modelo 'paraphrase-multilingual-MiniLM-L12-v2' do SentenceTransformer que foram utilizadas para computar a similaridade do cosseno entre elas.

É interessante apontar que a adição da etapa de identificação das 10 palavras mais relevantes com TF-IDF trigrama aumentou a correlação entre a similaridade calculada e a pontuação c2 de 0.05 para 0.156 no conjunto de treinamento.

3) Uso de repertório sociocultural: E, para o último aspecto, o REN com o modelo 'pt_core_news_md' do spaCy e com um repertório personalizado foi utilizado para identificar menções a pessoas, organizações, obras, eventos históricos e conceitos relevantes. O número de entidades identificadas foi contabilizado para cada redação do conjunto de treino, e constatamos uma correlação de 0.306 com a pontuação c2.

4) Pontuação c2: Apesar da estratégia inicial apresentar a pontuação c2 como resultado de um mapeamento das *features* extraídas nas subseções anteriores com base em um sistema de regras, as correlações fracas acarretam uma dificuldade expressiva em encontrar limiares que possibilitam a definição de regras claras para este problema. Por esse motivo, o treinamento de um modelo de aprendizado para inferir a pontuação é uma solução interessante.

Executando o KNN, o *Naive Bayes*, a Regressão Logística e o *Random Forest* para avaliações iniciais em uma busca em grade, o modelo que apresentou o maior desempenho foi a Regressão Logística, com uma acurácia de 0.5401 no conjunto de teste.

Como os conjuntos de treino, validação e teste são estratificados, observando a distribuição das redações por pontuação c2, verificamos que uma solução que sempre infere a nota 120 para a competência teria uma acurácia de aproximadamente 0.37, e o nosso modelo treinado para classificar redações com base na probabilidade de adequação ao gênero textual, na similaridade semântica entre a redação e o tema expresso pelos textos motivadores e no número de repertório sociocultural identificados na produção não é significativamente melhor do que um mero chute.

Tabela I
CORRELAÇÃO DOS ATRIBUTOS EXTRAÍDOS COM A PONTUAÇÃO C2.

Atributos extraídos	Correlação com c2
similarity_score	0.552
prob_dissertativo	0.156
num_entities	0.306

C. Competência III

1) Selecionar, relacionar , organizar e interpretar informações: features: Para a componente de avaliação da competência cada redação foi processada em parceria com o conjunto de textos motivadores para a geração de um arcabouço de *features* referentes a estrutura do texto, ao léxico, sintática e semântica da redação para a construção de abordagens baseadas em *features*, para definir o esqueleto do projeto do texto, qualidade do vocabulário, coerência e complexidade da argumentação, assim como fluxo lógico e aderência ao tema [3]. Para isso foram utilizadas heurísticas para caracterização de parágrafos e sentenças entre argumentação, tese e conclusão; técnicas de cálculo de similaridade entre sentenças parágrafos e textos; extração de métricas quantitativas em número de palavras, sentenças, parágrafos, argumentos; utilização de *POS tagging* para extração de quantidade em determinados tipos de palavras; profundidade de árvore sintática; quantidade e diversidade de entidades.

A grande maioria das *features* teve uma correlação baixa a moderada com a nota da competência 3, sendo a *feature* com maior correlação um valor modular de 0.4413 e a menor com o valor 0.0071.

2) Modelo baseado em regras: Um modelo de classificação foi definido baseado em regras. Esse modelo funciona criando um perfil de valores distinto para cada valor de nota possível, com base nos dados de treinamento. Primeiramente, para construir esses perfis, o modelo analisa cada *feature* de forma individual para cada classe e define uma faixa de valores típica. Essa faixa representa o comportamento mais comum dos dados, pois é determinada pelo intervalo entre o percentil 20% e o 80%, faixa que agrupa a maior parte dos valores de cada classe. Esse intervalo foi escolhido por agrupar a maior faixa de

dados, com poucas amostras tendo valores extremos fora desse intervalo, e para evitar remoção ou suavização dos valores das *features* no tratamento de *outliers*.

Uma vez que os perfis estão estabelecidos para todas as características, o modelo está pronto para classificar novas amostras. Quando uma amostra de teste é introduzida, o sistema a compara com os perfis criados, verificando, para cada uma de suas características, se o valor se encaixa na faixa típica das classes de notas. A classificação final é decidida por uma contagem de correspondências: a amostra é rotulada com a classe para a qual um número maior de suas características foi compatível.

3) Abordagens: Afim de avaliar diferentes abordagens para predição da nota, foi conduzida uma análise comparativa de seis modelos distintos: Regressão Linear (RL), *Support Vector Machine* (SVM), *Random Forest* (RF), *Gradient Boosting* (GB), *XGBoost* e o Modelo Baseado em Regras (MR) [4], [5], [6]. Para os modelos de aprendizado de máquina, foram testadas combinações aleatórias de hiperparâmetros, selecionando a melhor configuração com base no desempenho em validação. A avaliação foi estruturada em dois cenários: o primeiro utilizando as 46 *features* disponíveis, e o segundo com 15 *features* de maior correlação com o alvo.

No Grupo 1, utilizando o conjunto completo de *features*, os modelos de aprendizado de máquina apresentaram uma performance superior, com acurácia entre 48% e 54%, F1-Score de 25% a 32% e RMSE entre 34 e 39. Em contraste, o Modelo Baseado em Regras alcançou 27,48% de acurácia, com 21,54% em F1-Score e 53,87 de RMSE. No entanto, para a métrica QWK, enquanto os modelos de ML registraram valores de 0,32 a 0,38 , o modelo de regras obteve um QWK de 0,3. Isso sugere que, mesmo errando a classe exata com mais frequência, o modelo simples demonstrou uma capacidade maior de preservar a ordem correta das classificações.

Na análise do Grupo 2, com dimensionalidade reduzida, a acurácia dos modelos de aprendizado de máquina caiu para a faixa de 29% a 35%, mas, outras métricas melhoraram: o F1-Score subiu para 28% a 35% e o QWK teve um salto expressivo para a faixa de 0.48 a 0.58. Essa mudança indica que a seleção de *features*, embora tenha sacrificado a precisão geral, melhorou a coerência ordinal das previsões. Os modelos passaram a "errar menos drasticamente", classificando os exemplos em classes mais próximas da correta. O Modelo Baseado em Regras se manteve estável com acurácia em 30.63%, F1-Score de 21.97%, RMSE de 53,41 e QWK de 0,39.

D. Competência IV

1) Correlações: Para a avaliação dessa competência, que analisa a aplicação de mecanismos coesivos, foi realizada inicialmente uma investigação de correlações de *features* linguísticas. Essa análise exploratória identificou a diversidade de conectivos como a *feature* com a correlação mais expressiva com a nota (0.39), superando o número total de conectivos na redação (0.32).

Elas foram calculadas identificando a presença desses elementos por meio de uma extensa lista pré-definida, como proposto anteriormente. Para identificar repetições na redação, é definida uma janela deslizante, calculando a taxa de repetição local de palavras. A sua correlação com a nota obtida foi -0.21.

Após isso, mede-se também a relação do tamanho médio dos parágrafos (0.20), indicando que redações maiores no geral obtêm maiores notas. Identifica-se também que o uso de pronomes referenciais foram pouco relevantes (correlação 0,1).

Em contraste à estratégia definida anteriormente, a métrica de adequação contextual dos conectivos via BERT, mostrou correlação de 0.06, ou seja, irrelevante. O processamento mascara os conectivos e gera sugestões para checar se alguma delas corresponde com o elemento mascarado (ou a algum da mesma categoria). Conclui-se então que, para essa competência, o uso de conectivos é relevante, já o uso inadequado não é evidente em consonância com a nota.

2) Avaliadores: A partir dessas métricas, são estruturados avaliadores baseados em regras, em que todas essas identificações calculadas para a análise anterior são realizadas para gerar a pontuação final. Com ele, é obtida uma taxa de acerto exata de apenas 38.4%, e 85.2% de acerto com margem de um nível de nota (±40 pontos), indicando que pelo menos a maioria das suas previsões estava próxima da avaliação correta.

No entanto, o sistema demonstrou uma forte tendência a centralizar as notas, com dificuldade em identificar os extremos da pontuação (0, 40 e 200), superestimando textos fracos e subestimando os de nota máxima.

3) LightGBM: Para tentar superar as limitações das regras, a abordagem foi refinada para o modelo *Light Gradient Boosting Machine*, *framework* de aprendizado de máquina baseado em árvores de decisão que utiliza o algoritmo de *gradient boosting*.

A engenharia de *features* foi aprimorada, segmentando as redações em introdução, desenvolvimentos e conclusão, e extraindo métricas de coesão, vocabulário e estrutura para cada parte, além de calcular a similaridade semântica entre elas. Para tentar mitigar o forte desbalanceamento de notas no *dataset*, a técnica de oversampling SMOTE foi aplicada durante o treinamento do classificador.

O modelo demonstrou uma performance um pouco superior ao sistema de regras, com acurácia de 42,3%. Nota-se um avanço principalmente na capacidade de identificar redações de nota máxima. A análise de importância de *features* do modelo final, no entanto, revelou uma hierarquia clara: as *features* mais preditivas foram as estruturais, como a contagem de palavras por parágrafo.

Embora as *features* de conectivos mostraram-se importantes em isolamento, sua relevância diminuiu quando combinadas com os fortes sinais estruturais. A conclusão final é que a principal barreira para uma maior precisão permanece sendo a natureza desbalanceada do *dataset*, um desafio que persiste mesmo com a aplicação de técnicas de mitigação.

E. Competência V

A competência 5 compreende a avaliação da etapa final da redação. Após introduzir a problemática, contextualizá-la, argumentar os pontos de vista e informações relacionadas, é necessário que o escritor desenvolva um plano de intervenção a ser seguido como solução para o problema. Segundo a cartilha do participante, essa proposta elaborada pelo candidato deve apresentar alguns elementos essenciais que evidenciem de maneira clara o curso de ação a ser seguido e os resultados esperados. Esses elementos podem ser simplificados na forma de quatro questões: "o que será feito?", "quem fará?", "como será feito?", "qual o objetivo a ser alcançado?". Isso oferece um guia sobre quais informações presentes no corpo da redação têm maior relevância para a avaliação e pontuação do texto elaborado pelo candidato.

A estrutura de uma redação é outra característica que pode ser utilizada a fim de identificar onde se concentra o maior volume de informações necessárias para a competência em questão. Muitos modelos de redação propõem uma estrutura em 4 parágrafos, dedicando o primeiro parágrafo a introdução do tema, seguido de dois parágrafos para o desenvolvimento da argumentação e por fim o último sendo reservado para a conclusão. Para a competência 5, este último parágrafo tem maior probabilidade de oferecer relevância para a decisão de pontuar o texto do candidato, pois é na conclusão que se espera a descrição da proposta de intervenção elaborada. Contudo, apesar da maioria dos textos seguirem essa estrutura que os modelos recomendam, existe ainda uma certa diversidade estrutural na formatação dos textos encontrada nos dados. Isso dificulta a identificação do parágrafo onde o escritor desenvolveu suas ideias de conclusão sobre o assunto, e tal dificuldade deve ser considerada na etapa de extração dessas informações relevantes.

Baseando-se nessas características descritas acima que os textos apresentam, foram elaboradas três estratégias diferentes de processamento das redações com o objetivo de extrair informações relevantes para posteriormente treinar um classificador capaz de pontuar o texto de maneira precisa e consistente.

1) Análise sobre o texto completo: A primeira abordagem proposta para o desenvolvimento de uma solução foi uma análise sobre o texto em sua totalidade. Sem nenhuma etapa para avaliar e selecionar as informações com relação à relevância e utilidade para predição da pontuação, a ideia desta proposta é que o modelo treinado tenha a responsabilidade de realizar essa ponderação durante o processo de treinamento. Assim, o texto que se encontra na forma de parágrafos dentro do conjunto de dados, foi seccionado em sentenças com a ajuda da biblioteca NLTK [7]. Em seguida, as sentenças foram processadas pelo modelo BERTimbau, um modelo de linguagem em larga escala da arquitetura BERT treinado na língua portuguesa, a fim de se extrair uma representação vetorial das sentenças [8], [9].

A partir da representação em *embeddings*, os dados foram submetidos a uma cabeça de classificação neural rasa. Para realizar o treinamento, foi escolhido o otimizador AdamW por permitir aplicar regularização por decaimento de pesos mais diretamente [10]. Após as diversas iterações de treinamento variando os hiperparâmetros, essa abordagem obteve um desempenho mediano. O melhor modelo atingiu uma taxa de 48% de acertos na partição de testes que continha 20% da totalidade dos dados. Contudo, ao analisar o desempenho pela ótica da métrica QWK, a penalização dos erros pela distância demonstra uma taxa de assertividade de 30% [11].

2) Análise sobre as sentenças relevantes: A próxima abordagem para com o problema de pontuação automática foi de utilizar as características do texto e da língua para extrair informações mais relevantes. Seguindo os elementos da cartilha do participante, foram definidas regras baseadas nos conectivos e expressões comumente utilizados na língua para construir sentenças que respondam as perguntas guias. Para o processo de identificação das entidades, uma etapa de pré -processamento foi realizada para aplicar técnicas de Reconhecimento de Entidades Nomeadas e o nome das entidades obtidas foi adicionado ao conjunto de regras que identifica as sentenças relevantes.

Com as sentenças identificadas, seguiu o processo da solução de maneira relativamente similar à abordagem anterior. Utilizando do modelo BERTimbau para extrair uma representação vetorial referente às sentenças selecionadas e agrupá-las para então definir o que representaria o texto [8]. O mesmo procedimento de treinamento foi empregado para poder comparar a eficácia de fazer a seleção manual das informações relevantes. Entretanto o modelo treinado a partir dessa abordagem obteve um desempenho inferior, atingindo 45% de taxa de acerto e apenas 22% de assertividade pela métrica QWK [11].

3) Análise sobre os parágrafos relevantes: Outra ótica pela qual a tarefa de pontuar as redações pode ser abordada é avaliar o parágrafo de conclusão, que contém a proposta de intervenção alvo da análise para a competência. Devido a diversidade estrutural que os textos apresentam, para extrair o parágrafo que contém as ideias relativas à proposta de intervenção foi elaborado um conjunto de regras. Esse conjunto identifica as informações alvo a partir da comparação de conectivos comumente utilizados para finalizar fluxos de ideias na construção da texto. Dessa forma, é possível identificar os trechos relevantes que podem permitir a análise

das informações relativas à proposta do candidato.

Seguindo o protocolo já estabelecido nas outras abordagens, as sentenças que pertencem aos parágrafos selecionados foram representadas na forma de *embeddings*. Na sequência, essas representações foram utilizadas para o treinamento da um modelo neural raso para pontuar as redações nos quesitos da competência 5. Essa abordagem obteve uma taxa de acerto similar à seleção de sentenças de 45%, mas não superou o desempenho da análise sobre o texto completo. Em relação a métrica QWK, ela teve uma assertividade superior a seleção das sentenças, chegando a 25% nesse quesito [11].

IV. Considerações Finais

É possível concluir que o desempenho das soluções apresentadas não são significativamente melhores que um mero chute na classe majoritária. As *features* extraídas para os cálculos das pontuações em cada competência apresentaram correlações fracas com o atributo alvo, dificultando sua inferência, e o desbalanceamento das classes no conjunto de treinamento acarretou em um desempenho lamentável para as classes minoritárias, que apresentaram, na maioria das competências, métricas como precisão e revocação iguais a zero. Ademais, constatamos a existência de amostras inválidas no conjunto de dados, com formatos inadequados e avaliações incoerentes, como produções textuais de outros gêneros, anotadas com notas relativamente altas.

Considerando a natureza da tarefa, a qualidade das anotações e correlações encontradas, o ideal para o desenvolvimento seria obter auxílio de especialistas que pudessem contribuir para anotações mais ricas e qualitativas das amostras, como marcação de argumentos e tese, além de atuar como norteadores na escolha de *features* que tenham mais impacto na geração das notas.

Referências

[1] INSTITUTO NACIONAL DE ESTUDOS E PESQUISAS EDUCACIONAIS ANÍSIO TEIXEIRA (INEP), "A redação no enem 2024: cartilha do participante," Brasília, DF, 2024, acesso em: 4 jun. 2025. [Online]. Available: https://download.inep.gov.br/publicacoes/institucionais/avaliacoes_e_exames_da_educacao_basica/a_redacao_no_enem_2024_cartilha_do_participante.pdf

[2] "Essay-br: a brazilian corpus of essays," *Dataset Showcase Workshop (DSW)*, pp. 53–64, 10 2021. [Online]. Available: https://sol.sbc.org.br/index.php/dsw/article/view/17414

[3] X. Wang, Y. Lee, and J. Park, "Automated evaluation for student argumentative writing: A survey," 5 2022. [Online]. Available: https://arxiv.org/pdf/2205.04083

[4] T. Evgeniou and M. Pontil, "Support vector machines: Theory and applications," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2049 LNAI, pp. 249–257, 2001. [Online]. Available: https://www.researchgate.net/publication/221621494_Support_Vector_Machines_Theory_and_Applications

[5] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 10 2001. [Online]. Available: https://link.springer.com/article/10.1023/A:1010933404324

[6] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 13-17-August-2016, pp. 785–794, 3 2016. [Online]. Available: https://arxiv.org/pdf/1603.02754

[7] E. Loper and S. Bird, "Nltk: The natural language toolkit," *COLING/ACL 2006 - 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Interactive Presentation Sessions*, pp. 69–72, 5 2002. [Online]. Available: https://arxiv.org/pdf/cs/0205028

[8] F. Souza, R. Nogueira, and R. Lotufo, "Bertimbau: Pretrained bert models for brazilian portuguese," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12319 LNAI, pp. 403–417, 2020.

[9] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, vol. 1, pp. 4171–4186, 10 2018. [Online]. Available: https://arxiv.org/pdf/1810.04805

[10] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *7th International Conference on Learning Representations, ICLR 2019*, 11 2017. [Online]. Available: https://arxiv.org/pdf/1711.05101

[11] A. Doewes, N. A. Kurdhi, and A. Saxena, "Evaluating quadratic weighted kappa as the standard performance metric for automated essay scoring," 7 2023. [Online]. Available: https://zenodo.org/records/8115784