Automated Essay Scoring: An approach based on ENEM competencies

Jeziel C. Marinho[1], **Fábio Cordeiro**[2], **Rafael T. Anchiêta**[3], **Raimundo S. Moura**[2]

[1]Federal Institute of Maranhão (IFMA) – Barra do Corda, MA – Brazil

[2]Federal University of Piauí (UFPI) – Teresina, PI – Brazil

[3]Federal Institute of Piauí (IFPI) – Picos, PI – Brazil

jeziel.marinho@ifma.edu.br, rta@ifpi.edu.br,

fabiocordeiro@gmail.com, rsm@ufpi.edu.br

Abstract. *This work presents strategies for Automatic Essays Scoring (AES) written in Portuguese through an approach based on the definition of features and specific AES models for each competence of the ENEM reference matrix. We investigate methods based on features engineering, embeddings, and Recurrent Neural Networks. Although the results obtained are better than related works, further studies should be conducted in order to improve the performance of AES models for the Portuguese language.*

Resumo. *Este trabalho apresenta estratégias para Avaliação Automática de Redações (AAR) escritas em português por meio de uma abordagem baseada na definição de features e modelos de AAR específicos para cada competência da matriz de referência do ENEM. Foram investigados métodos baseados em engenharia de features, embeddings e Redes Neurais Recorrentes. Apesar dos resultados obtidos serem melhores do que trabalhos relacionados, novos estudos devem ser conduzidos a fim de melhorar o desempenho dos modelos de AAR para a língua portuguesa.*

1. Introdução

No Brasil, o Exame Nacional do Ensino Médio (ENEM) contém, atualmente, a maior prova de redação do país em número de participantes [INEP 2021]. Em 2020, foram quase 2,8 milhões de candidatos inscritos e apenas 28 alcançaram a nota máxima na redação do ENEM. Destaca-se que 3,22% dos participantes obtiveram nota 0 [INEP 2021]. De acordo com [Gonçalves and Carvalho 2010], parte da problemática por traz de resultados tão ruins está relacionada ao baixo volume de produções textuais dos candidatos durante o período de estudos para exame. Além do mais, a evolução no desempenho dos candidatos para elaboração de boas redações depende de um *feedback* de profissionais da área de Letras/Língua Portuguesa ou Linguística em relação ao que deve ser melhorado, o que pode acarretar em uma sobrecarga de trabalho dos profissionais no acompanhamento destes candidatos.

A Avaliação Automática de Redações (AAR), por meio de técnicas de Processamento de Linguagem Natural (PLN) e Aprendizado de Máquina (AM), busca avaliar e pontuar textos em prosa escrita [Dikli 2006]. A AAR como área de pesquisa teve início

com o *Project Essay Grader* (PEG) [Page 1966] e, apesar da importância da área de AAR, e de haver muitas pesquisas relacionadas a AAR em vários idiomas, principalmente para o inglês [Ke and Ng 2019], o número de pesquisas voltadas o português ainda é pequeno.

[Ke and Ng 2019] apresentam uma visão geral sobre a área de AAR e citam que a grande maioria dos sistemas de AAR existentes foi desenvolvida para pontuação holística. A pontuação de dimensões específicas, ou competências, como esclarecem os autores, ainda é pouco explorada. Eles citam que praticamente todos as pesquisas abordaram a tarefa de AAR como uma tarefa de regressão [Izbicki and dos Santos 2020]. [Beigman Klebanov and Madnani 2020] mencionam o fato de os sistemas de AAR ainda não serem capazes de avaliar a originalidade de um aluno ao produzir uma redação, ou mesmo avaliar o seu nível de conhecimento em relação ao conteúdo abordado na redação.

Neste contexto, esta pesquisa tem como objetivo implementar e analisar estratégias para avaliação automática de redações escritas na língua portuguesa, seguindo os critérios estabelecidos para as cinco competência da matriz de referência da redação do ENEM. Nossa abordagem faz uso de técnicas de PLN e algoritmos de AM supervisionados. Investigaram-se métodos baseados em engenharia de *features* [Sarkar 2019], *embeddings* [Le and Mikolov 2014] e rede neurais recorrentes (do inglês, *Recurrent Neural Network* - RNN) do tipo *Long Short-Term Memory* (LSTM) [Hochreiter and Schmidhuber 1997] para implementar cinco modelos de AAR independentes e capazes de predizer as notas das competências da redação do ENEM. Esses modelos foram avaliados em um *corpus* publicamente disponível [Marinho et al. 2022], alcançando resultados superiores aos trabalhos relacionados.

O restante deste artigo está organizado da seguinte forma. A Seção 2 apresenta conceitos sobre o ENEM, focando na avaliação da redação. Na Seção 3, são descritos os principais trabalhos relacionados. Na Seção 4, são detalhados os métodos desenvolvidos. A Seção 5 apresenta e discute os resultados obtidos; Finalmente, a Seção 6 conclui o artigo, indicando trabalhos futuros.

2. ENEM

O ENEM possui 180 questões objetivas de múltipla escolha e uma proposta de redação. De acordo com a cartilha do participante do ENEM 2020 [INEP 2020], a prova de redação exige a produção de um texto em prosa, do tipo dissertativo-argumentativo, sobre um tema de ordem social, científica, cultural ou política.

O termo Matriz de Referência (MR), de acordo com os manuais de correção da redação do ENEM [INEP 2019], é utilizado especificamente no contexto das avaliações em larga escala para orientar a elaboração de itens de testes e provas, bem como a construção de escalas de proficiência. Durante o processo de correção das redações, são levados em consideração os critérios presentes nas seguintes competências definidas na Matriz de Referência da redação do ENEM:

1. Demonstrar domínio da modalidade escrita formal da língua portuguesa.
2. 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.
3. Selecionar, relacionar, organizar e interpretar informações, fatos, opiniões e argumentos em defesa de um ponto de vista.

4. Demonstrar conhecimento dos mecanismos linguísticos necessários para a construção da argumentação.
5. Elaborar proposta de intervenção para o problema abordado, respeitando os direitos humanos.

Durante o processo de avaliação das redações, pelo menos dois professores formados em letras ou linguística fazem as correções levando em consideração os critérios presentes nas competências definidas na matriz de referência. Cada avaliador atribui uma nota que pode ser 0, 40, 80, 120, 160 ou 200 pontos para cada uma das cinco competências. A soma desses pontos compõe a nota total de cada avaliador, que pode chegar a 1.000 pontos. A nota final do participante será a média aritmética das notas totais atribuídas pelos dois avaliadores.

Caso haja uma discrepância de mais de 100 pontos na nota final ou uma diferença de mais de 80 pontos entre as notas atribuídas pelos dois avaliadores em qualquer uma das competências, a redação será avaliada por um terceiro avaliador e a nota final será a média aritmética das duas notas que mais se aproximarem. Em ainda havendo discrepância, a redação será avaliada por uma banca composta por três professores, que atribuirá a nota final do participante.

Por fim, o [INEP 2019] deixa claro o caráter independente da avaliação de cada competência, pelo qual define-se que a correção de cada uma das cinco competências e a consequente atribuição de nota se dá de forma que a nota de uma competência não influencia nas notas das outras quatro.

3. Trabalhos relacionados

De acordo com [Shermis and Hamner 2013], existem vários trabalhos de pesquisa e sistemas comerciais de Avaliação Automática de Redações (AAR) para a língua inglesa como por exemplo o *Intelligent Essay Assessor* (IEA) de Pearson[1], o *Graduate Record Examination* (GRE)[2], ou o *Test of English as a Foreign Language* (TOEFL)[3]. Aqui, serão destacados os principais trabalhos para a língua portuguesa.

[Amorim et al. 2018] propuseram um sistema de AAR para o português brasileiro utilizando uma base de dados com 1840 redações sobre 96 assuntos diferentes extraídas do site da web. Durante pesquisa, os autores avaliaram o desempenho da previsão da nota final bem como das notas para cada uma das cinco competências do ENEM por meio do índice Kappa Quadrático Ponderado (do inglês, *Quadratic Weighted Kappa* - QWK) [Cohen 1968]. Um segundo experimento realizado pelos autores foi a análise do papel de cada *feature* na tarefa de previsão de notas. Os autores observaram que as *features* mais relevantes para a nota final não são necessariamente uma mistura das *features* mais relevantes para a pontuação das competências.

[Fonseca et al. 2018] seguiram duas abordagens para AAR em português: Redes neurais profundas e sistemas baseados em *features*. Os autores utilizaram um *dataset* com 56.644 redações escritas em uma plataforma online e anotadas por avaliadores humanos de acordo com as cinco competências do ENEM. O modelo foi composto por duas camadas de redes neurais recorrentes do tipo LSTM bidirecionais (BiLSTM) treinadas por

[1]https://pearsonpte.com/the-test/about-our-scores/ how-is-the-test-Score

[2]https://www.ets.org/gre/revised_general/scores/how

[3]https://www.ets.org/toefl/ibt/scores/understand

apenas duas épocas, com lotes de 8 redações. No segundo experimento, os autores utilizaram 681 *features* para treinar cinco regressores separadamente. Conforme esclarecem os autores, apesar dos métodos de aprendizado profundo terem alcançado um sucesso considerável, sistemas baseados em *features* ainda valem a pena investigar para AAR pois eles apresentam uma explicação mais transparente e são mais fáceis de treinar.

[Júnior 2020] apresentou um estudo sobre o uso de redes neurais profundas para atarefa de AAR escritas em português seguindo os critérios do ENEM por meio de uma arquitetura baseada em aprendizado multitarefa [Caruana 1998] para múltiplos temas. O autor utilizou redes Elman RNN [Elman 1990], redes GRUs [Cho et al. 2014] e redes LSTM combinadas com diferentes mecanismos de agregação [Luong et al. 2015] e diferentes combinações de representações de palavras. O autor implementou ainda uma rede neural de aprendizado multitarefa em que cada um dos temas foi considerado como uma tarefa diferente. Apesar do *corpus* utilizado por pelo autor conter as notas das 5 competências, os modelos desenvolvidos foram treinados apenas para a predição da nota final.

Fazendo uso do *corpus* criado por [Amorim et al. 2018], a pesquisa de [de Almeida Júnior 2017] tratou apenas da competência 1 da matriz de referência da redação do ENEM propondo um abordagem baseada em engenharia de *features* em que cada tipo de erro gramatical é considerado como uma caraterística do problema. Utilizando também o *corpus* de [Amorim et al. 2018], [Haendchen Filho et al. 2018] buscaram avaliar apenas a segunda competência por meio da extração de *features* de estruturas argumentativas associadas à contagem de palavras extraídas de um dicionário analógico da língua portuguesa [dos Santos Azevedo 2019]. Por fim, ainda utilizando o *corpus* de [Amorim et al. 2018], [da Silva Junior 2021] utilizou *features* relacionadas à quantidade de erros ortográficos e gramaticais, *features* relacionadas à sofisticação léxica e à entidades nomeadas para propor um modelo de AAR apenas para a terceira competência.

O diferencial desta pesquisa é que, como resultado da abordagem adotada, buscou-se definir um conjunto de *features* para cada competência da redação do ENEM, treinando cinco modelos diferentes. Tal estratégia tem como fundamento o fato de que, de acordo com o manuais de correção da redação do ENEM[4], a correção de cada uma das cinco competências de uma redação tem caráter totalmente independente no sentido de que, a nota atribuída pelo avaliador a uma competência não está diretamente relacionada ao desempenho do candidato nas outras competências.

4. Métodos desenvolvidos

Nesta seção, apresentam-se os detalhes sobre o *corpus* utilizado nesta pesquisa, bem como os métodos desenvolvidos para avaliação automática de redações.

4.1. Corpus

O *corpus* utilizado neste trabalho é uma parte da versão estendida do *corpus* ***Essay-Br*** [Marinho et al. 2021] que foi criado a partir da extração automatizada de redações

[4]https://www.gov.br/inep/pt-br/areas-de-atuacao/avaliacao-e-exames-educacionais/enem/outros-documentos

de dois sites públicos: Universo Online (UOL)[5] e Brasil Escola[6]. Ao todo, o *corpus Essay-Br* possui 4.570 redações com 86 temas produzidas entre os meses de dezembro de 2015 a abril de 2020. Além das redações, o *corpus* possui também os temas das redações e as notas das cinco competências. Na versão estendida do *corpus*, foram extraídas mais 2.009 redações e mais 65 temas, perfazendo um total de 6.579 redações distribuídas em 151 temas. Neste trabalho, utilizou-se apenas 5.730 redações, pois são as redações que estão de acordo com a pontuação do ENEM.

Nos experimentos realizados, definiram-se conjuntos estratificados de treinamento, desenvolvimento e teste para cada uma das cinco competência da matriz de referência, levando em consideração a distribuição das notas. Organizou-se os conjuntos em 80% para treinamento, 10% para teste e desenvolvimento.

4.2. Método baseado em Engenharia de *Features*

A partir da análise dos manuais do avaliador do ENEM, foram definidos conjuntos de *features* específicas para cada competência da matriz de referência da redação do ENEM a fim de buscar extrair os aspectos léxicos, sintáticos e semânticos observados pelos avaliadores humanos durante o processo de correção. Para cada competência, foram definidos os seguintes conjuntos de *features*:

Competência 1: quantidade de palavras erradas e erros gramaticais; média de sentenças por parágrafos; quantidade de parágrafos com uma sentença, média de orações por sentença; quantidade de sentenças simples e compostas, quantidade de sentenças sem verbo; média de palavras antes dos verbos principais das orações principais das sentenças; quantidade de orações com advérbio antes do verbo principal em relação à quantidade de orações do texto; proporção de locuções conjuntivas em relação à quantidade de sentenças; quantidade de sentenças iniciadas com locuções conjuntivas e proporção entre verbos e palavras.

Competência 2: quantidade de parágrafos; quantidade média de palavras por parágrafos; similaridade com os textos motivadores; quantidade de entidade nomeadas; quantidade de pronomes e verbos na primeira pessoa; diversidade lexical e estatística de honoré [Honoré 1979].

Competência 3: diversidade léxica; quantidade de entidades nomeadas; proporção de operadores argumentativos em relação à quantidade de palavras do texto; similaridade com os textos motivadores; média da similaridade entre pares de sentenças adjacentes; média da similaridade entre todos os pares de sentenças; similaridade entre pares de parágrafos adjacentes e similaridade entre uma sentença e todas as outras anteriores a ela.

Competência 4: proporção de operadores argumentativos em relação à quantidade de palavras do texto; similaridade com os textos motivadores; média das proporções de candidatos a referentes na sentença anterior em relação aos pronomes pessoais do caso reto nas sentenças; média de candidatos a referente na sentença anterior por pronome anafórico do caso reto e por pronome demonstrativo anafórico; quantidade média de referentes que se repetem nos pares de sentenças adjacentes do texto e entre todos os pares de

[5]https://educacao.uol.com.br/bancoderedacoes/

[6]https://vestibular.brasilescola.uol.com.br/banco-de-redacoes

sentença no texto; quantidade média de palavras de conteúdo e de radicais de palavras de conteúdo que se repetem nos pares de sentenças adjacentes do texto e em todas os pares de sentenças do texto.

Competência 5: proporção de marcadores de discurso conclusivos em relação a quantidade de palavras de conteúdo; proporção de verbos na primeira pessoa em relação a quantidade de verbos; proporção de verbos no modo imperativo em relação a quantidade de verbos no texto; quantidade de parágrafos e tamanho médio dos parágrafos.

Para a extração das *features*, utilizaram-se várias ferramentas, léxicos e bibliotecas disponíveis para a linguagem Python. Quanto as ferramentas utilizadas nesta abordagem, para a análise ortográfica das redações, utilizou-se um corretor ortográfico e analisador morfológico PyHunSpell[7] juntamente com os dicionários da língua Portuguesa Brasileira Hunspell[8] e Unitex-PB[9]. A análise gramatical foi obtida utilizando a biblioteca CoGrOO4py[10] que funciona como uma interface para acessar o analisador morfológico e o corretor gramatical CoGrOO[11]. A identificação dos períodos das redações, bem como a análise das classes gramaticais do texto foram obtidas utilizando a ferramenta Stanza [Qi et al. 2020].

A similaridade entre os textos foi calculada utilizando o método de Análise Semântica Latente (do inglês, *Latent Semantic Analysis* - LSA) [Landauer et al. 1998]. O LSA computa similaridade entre trechos de textos considerando conhecimento implícito além de palavras similares fazendo uso de métodos que extraem e representam o significado de uso contextual de palavras por cálculos estatísticos aplicados a um grande*corpus* de texto.

A biblioteca NLTK [Bird et al. 2009] foi utilizada para tokenizar os textos das redações e a ferramenta spaCy[12] para identificar as entidades nomeadas.

4.3. Método baseado em *Embeddings Doc2Vec*

Esta abordagem fez uso de modelos *Doc2Vec* PV-DM [Le and Mikolov 2014] e *Doc2Vec* PV-DBOW [Le and Mikolov 2014] treinados sobre o *corpus* de redações. Nesta abordagem, treinaram-se regressores independentes para cada competência.

Aplicou-se um pré-processamento no *corpus* de redações em que todas as letras foram convertidas em minúsculas e foram removidas: *stopwords*, *hapax legonemas* (palavras com ocorrência única no texto), números e caracteres de pontuação.

Após o pré-processamento, dois modelos de *embeddings*, um *Doc2Vec* PV-DM e um *Doc2Vec* PV-DBOW, com tamanho vetorial de 300 dimensões foram treinados sobre a totalidade do *corpus* de redação durante 100 épocas.

Os modelos de *embeddings* resultantes foram utilizados como entrada no treinamento de algoritmos tradicionais de regressão a fim de predizer a nota de cada uma das 5 competências da matriz de referência da redação do ENEM.

[7]https://github.com/blatinier/pyhunspell
[8]https://natura.di.uminho.pt/wiki/doku.php?id=dicionarios:main
[9]http://www.nilc.icmc.usp.br/nilc/projects/unitex-pb/web/index.html
[10]https://github.com/gpassero/cogroo4py
[11]http://comunidade.cogroo.org/
[12]https://spacy.io/

4.4. Método baseado em RNN do tipo LSTM

Para o método baseado em RNN do tipo LSTM, os experimentos conduzidos utilizaram *Embeddings* treinadas sobre o *corpus* de redação e *Embeddings* pré-treinadas.

Nesta abordagem, os textos das redações passaram por um pré-processamento no qual foram removidos caracteres não alfanuméricos e todas as letras foram convertidas para minúsculas. Após o pré-processamento, os textos foram transformados em vetores de tamanho 1.000.

Para os experimentos com *Embeddings* pré-treinadas, foi utilizado um modelo *Word2Vec* CBOW com 300 dimensões gerado a partir de 17 *corporas* diferentes do português do Brasil e português europeu, de fontes e gêneros variados, com um total de 1.395.926.282 *tokens* [Hartmann et al. 2017].

A estrutura da RNN foi formada por uma camada de *embeddings*, que recebe como entrada as redações pré-processadas e tokenizadas e é responsável por gerar uma matriz formada pelos vetores de *embeddings* de 300 dimensões. Nos experimentos realizados com as *embeddings* pré-treinadas, essa primeira camada foi configurada como não treinável. A segunda camada da RNN foi formada por 10 células LSTM seguida da camada de saída com uma função de ativação *softmax*.

As RNNs foram treinadas por 100 épocas com lotes de 10 redações por vez. Durante o treinamento, foi utilizado o otimizador Adam [Kingma and Ba 2015] com uma taxa de aprendizado de 0.001 e o erro quadrático médio como função de erro.

Todos os hiperparâmetros utilizados para os experimentos com a RNN foram definidos de forma empírica. A Tabela 1 apresenta um sumário dos hiperparâmetros utilizados.

Tabela 1. Sumário de hiperparâmtros utilizados.

Hiperparâmetro	**Valor**	**Hiperparâmetro**	**Valor**
Word Embedding	CBOW	Tamanho dos *batchs*	10
Dimensão das *embeddings*	300	Otimizador	Adam
Tamanho do vocabulário	30000	Taxa de aprendizado	0.001
Tamanho do vetor de *tokens*	1000	Valor de *dropout*	0.5
Número de épocas	100		

5. Resultados

Para a avaliação dos modelos implementados, utilizou-se a métrica Kappa Quadrático Ponderado (do inglês, *Quadratic Weighted Kappa* - QWK) [Cohen 1968], pois é a métrica utilizada na tarefa de avaliação automática de redações. Em virtude da QWK considerar apenas valores discretos, as notas preditas pelos modelos implementados foram discretizadas para corresponder ao padrão de notas definidas pelo ENEM de acordo com a Tabela 2.

Para o método baseado em engenharia de *features* foi utilizado o agoritmo GBTD [Géron 2019] com um total de 100 estimadores e uma função de erro absoluto com uma taxa de aprendizagem igual a 0.1. Já para o método baseado em *Doc2Vec*, foi utilizado o algoritmo de regressão de *Ridge* [Géron 2019] configurado com um termo de

Tabela 2. Discretização das notas preditas pelos regressores para o padrão de notas do ENEM.

nota ***n*** do regressor	nota	nota ***n*** do regressor	nota
n <20	0	100 <= ***n*** <140	120
20 <= ***n*** <60	40	140 <= ***n*** <180	160
60 <= ***n*** <100	80	***n*** >180	200

regularização α igual a 0.8 e o número máximo de iterações igual a 10000. Outros algoritmos de regressão foram testados, porém o GBTD e a regressão de *Ridge* obtiveram os melhores desempenhos.

A Tabela 3 apresenta os resultados obtidos em cada abordagem adotada para as cinco competências da matriz de referência. Os resultados são referentes aos conjuntos de teste do *corpus* de redações estratificados para cada competência. Na tabela, é possível observar em destaque os melhores valores de QWK para cada competência.

Tabela 3. Valores de QWK em cada abordagem.

Abordagem	Competência 1	Competência 2	Competência 3	Competência 4	Competência 5
Features + GBDT	**0.4661**	**0.4917**	0.3015	0.5272	0.2117
Doc2Vec PV-DM + *Ridge*	0.3799	0.4035	0.4297	0.5043	0.4563
Doc2Vec PV-DBOW + *Ridge*	0.3403	0.1845	0.2880	0.3322	0.2555
RNN LSTM + Embeddings	0.4096	0.4072	**0.4445**	**0.5591**	**0.5636**
RNN LSTM + Embeddings pré-treinadas	0.3638	0.4614	0.3816	0.5309	0.4127

Para a primeira competência, o modelo de engenharia de *features* apresentou o melhor desempenho, possivelmente pelo fato delas conseguirem capturar bem os erros ortográficos e gramaticas da redação. Para a segunda competência, a engenharia de *features* também obteve o melhor desempenho pois o aspecto mais relevante relacionado à avaliação dessa competência é a similaridade com os textos motivadores.

O modelo baseado em RNN com *embeddings* treinadas sobre o *corpus* de redação obteve os melhores valores na predição da nota das competências 3, 4 e 5, demostrado a capacidade das RNNs capturarem bem a subjetividade envolvida na avaliação destas competências.

De acordo com a interpretação do valor de QWK proposta por [Altman 1990] e apresentada na Tabela 4, tanto os modelos baseados em engenharia de *features* quanto os modelos baseados em RNNs obtiveram uma concordância moderada.

Tabela 4. Interpretação dos valores de QWK.

valor de QWK	Nível de concordância
<0.20	pobre
0.21 - 0.40	razoável
0.41 - 0.60	moderado
0.61 - 0.80	bom
0.81 - 1.00	muito bom

A Figura 1 apresenta as matrizes de confusão dos modelos que obtiveram os melhores resultados em cada competência, ou seja, dos modelos baseados em engenharia de

features para a primeira e segunda competência e dos modelos baseados em RNN para as competências 3, 4 e 5.

Ao analisar as matrizes de confusão da Figura 1 é possível observar que o alto grau de desbalanceamento das notas de cada competência impactaram negativamente no desempenho dos modelos na correta predição das notas menores que 80 e maiores que 160. Apenas para as competências 1 e 3 os modelos selecionados foram capazes de predizer corretamente notas 40 e o modelo selecionado para a primeira competência foi capaz de predizer apenas uma nota 200 corretamente.

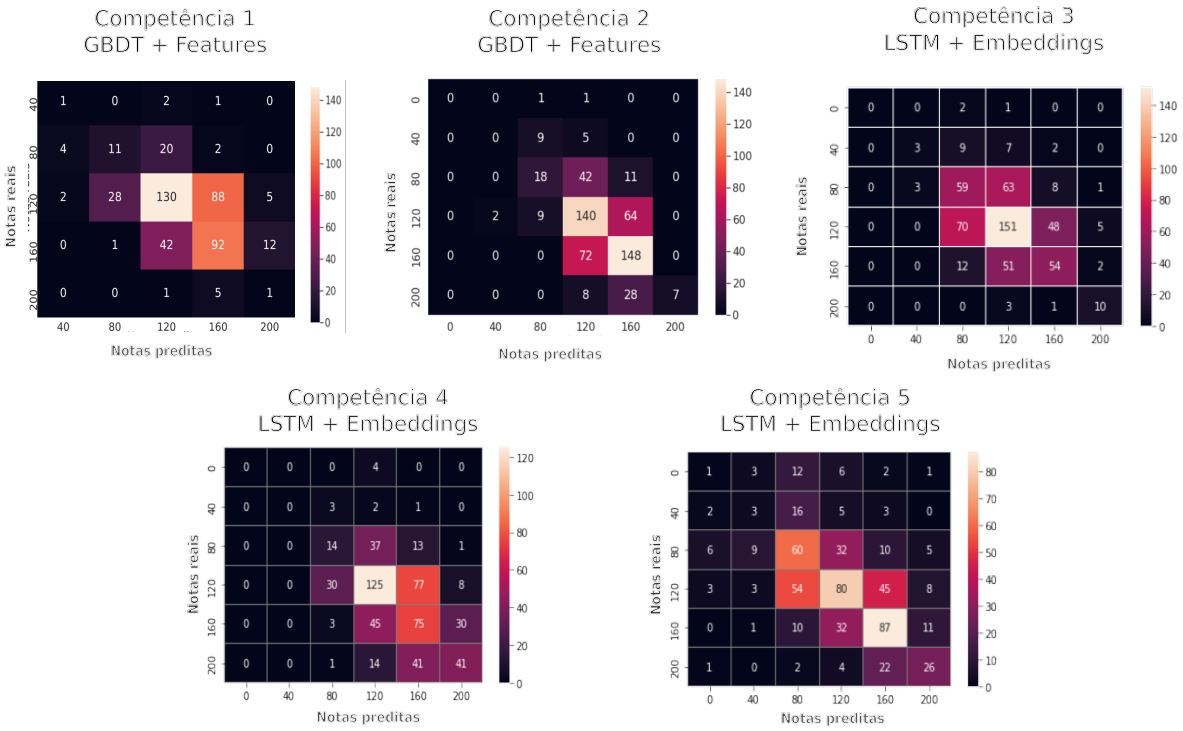

Figura 1. Matrizes de confusão dos modelos que obtiveram os melhores resultados em cada competência.

Os modelos propostos por [Amorim et al. 2018] e [Fonseca et al. 2018] foram replicados de acordo com as especificações apresentadas pelos autores e aplicados nos conjuntos de treino e teste do *corpus* utilizado neste trabalho. Os resultados obtidos com a replicação destes trabalhos podem ser observados na Tabela 5 que apresenta um comparativo com métodos implementados nesta pesquisa que obtiveram os melhores resultados em cada competência.

Tabela 5. Comparação de resultados.

	Competência 1	Competência 2	Competência 3	Competência 4	Competência 5
Nossa abordagem	**0.4661**	**0.4917**	**0.4445**	**0.5591**	**0.5636**
[Amorim et al. 2018]	0.2598	0.2315	0.1691	0.2295	0.2011
[Fonseca et al. 2018]	0.2083	0.3131	0.1930	0.2863	0.2718

6. Conclusão

Considerando que cada competência da matriz de referência da redação do ENEM pode ser avaliada de forma independente uma da outra, a nossa abordagem buscou definir *fe-*

atures e modelos de AAR independentes para cada competência. Durante a pesquisa, insvestigou-se o uso de métodos baseados em engenharia de *features*, *embeddings* e modelos baseados em RNNs.

Para as competências 1 e 2, o método de engenharia de *features* se mostrou mais eficiente. Já para as competência 3, 4 e 5, a RNN obteve os melhores resultados, mostrando que elas se apresentam como uma melhor opção em relação à engenharia de *features* para capturar a subjetividade de um texto.

Na interpretação de [Altman 1990] para os valores de QWK, os resultados dos experimentos conduzidos nesta pesquisa atingiram uma concordância moderada nos conjuntos de teste do *corpus* de redações. Entretanto, apesar destes valores serem significativamente melhores do que os trabalhos de [Amorim et al. 2018] e [Fonseca et al. 2018], novos estudos devem ser conduzidos a fim de melhorar o desempenho dos modelos de AAR para a língua portuguesa.

Por fim, como trabalhos futuros, planeja-se a realização de outros experimentos como a aplicação de técnicas de geração de amostragens sintéticas para o balanceamento do *corpus* de redação a fim de tentar reduzir o impacto do desbalanceamento no desempenho dos modelos. Planeja-se também experimentar outros algoritmos de *deep learning* como Redes BiLSTM e Redes convolucionais. Além disso, pretende-se fazer uso de métodos de PLN reconhecidamente mais robusto como os modelos de língua BERT [Devlin et al. 2019] ou ELMO [Peters et al. 2018].

Agradecimentos

Os autores agradecem ao IFMA por apoiar este trabalho.

Referências

Altman, D. (1990). *Practical Statistics for Medical Research*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis.

Amorim, E., Cançado, M., and Veloso, A. (2018). Automated essay scoring in the presence of biased ratings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 229–237, New Orleans, Louisiana. Association for Computational Linguistics.

Beigman Klebanov, B. and Madnani, N. (2020). Automated evaluation of writing – 50 years and counting. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7796–7810, Online. Association for Computational Linguistics.

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly.

Caruana, R. (1998). *Multitask Learning*, pages 95–133. Springer US, Boston, MA.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Cohen, J. (1968). Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4):213–220.

da Silva Junior, J. A. (2021). Um avaliador automático de redações. Master's thesis, Universidade Federal do Espírito Santo.

de Almeida Júnior, C. R. C. (2017). Proposta de um sistema automático de avaliação de redações do enem, foco na competência 1: Demonstrar domínio da modalidade escrita formal da língua portuguesa. Master's thesis, Universidade Federal do Espírito Santo.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dikli, S. (2006). An overview of automated scoring of essays. *The Journal of Technology, Learning and Assessment*, 5(1).

dos Santos Azevedo, F. (2019). *Dicionário analógico da língua portuguesa: ideias afins/ thesaurus*. Obras de referência. Lexikon.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2):179–211.

Fonseca, E., Medeiros, I., Kamikawachi, D., and Bokan, A. (2018). Automatically grading brazilian student essays. In *Proceedings of the 13th International Conference on Computational Processing of the Portuguese Language*, pages 170–179, Canela, Brazil. Springer International Publishing.

Gonçalves, C. R. and Carvalho, M. T. N. d. (2010). Prática textual: ensino, produção e revisão. *Scripta*, 14(26):235–249.

Géron, A. (2019). *Mãos à Obra: Aprendizado de Máquina com Scikit-Learn e TensorFlow*. Alta Books.

Haendchen Filho, A., Prado, H., Ferneda, E., and Nau, J. (2018). An approach to evaluate adherence to the theme and the argumentative structure of essays. *Procedia Computer Science*, 126:788–797.

Hartmann, N. S., Fonseca, E. R., Shulby, C. D., Treviso, M. V., Rodrigues, J. S., and Aluísio, S. M. (2017). Portuguese word embeddings: Evaluating on word analogies and natural language tasks. In *Anais do XI Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana*, pages 122–131, Porto Alegre, RS, Brasil. SBC.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Honoré, A. (1979). Some simple measures of richness of vocabulary. *Association for literary and linguistic computing bulletin*, 7(2):172–177.

INEP (2019). *Apostila de capacitação dos corretores de redação, Competência 1*.

INEP (2020). *A redação do ENEM, cartilha do participante*.

INEP (2021). Enem 2020, resultados edição impressa, digital e ppl.

Izbicki, R. and dos Santos, T. M. (2020). *Aprendizado de máquina: uma abordagem estatística*.

Júnior, J. A. S. B. (2020). Avaliação automática de redação em língua portuguesa empregando redes neurais profundas. Master's thesis, Universidade Federal de Goiás, Goiânia.

Ke, Z. and Ng, V. (2019). Automated essay scoring: A survey of the state of the art. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 6300–6308. International Joint Conferences on Artificial Intelligence Organization.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, San Diego, CA, USA.

Landauer, T., Foltz, P., and Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, 25:259–284.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196, Bejing, China. PMLR.

Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

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.

Marinho, J. C., Anchiêta, R. T., and Moura, R. S. (2022). Essay-br: a brazilian corpus to automatic essay scoring task. *Journal of Information and Data Management*, 13:65–76.

Page, E. B. (1966). The imminence of... grading essays by computer. *The Phi Delta Kappan*, 47(5):238–243.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., and Manning, C. D. (2020). Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Sarkar, D. (2019). *Text Analytics with Python: A Practitioner's Guide to Natural Language Processing*. APress, 2nd edition.

Shermis, M. and Hamner, B. (2013). Contrasting state-of-the-art automated scoring of essays. *Handbook of automated essay evaluation*, pages 313–346.