Universidade Federal de Catalão
Instituto de Biotecnologia
Curso de Bacharelado em Ciência da Computação

Análise e uso de técnicas para avaliação automática de redações no contexto do ENEM com foco na Competência 1

Gustavo Evangelista Araújo

Catalão – GO
2021

Gustavo Evangelista Araújo

Análise e uso de técnicas para avaliação automática de redações no contexto do ENEM com foco na Competência 1

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Federal de Catalão, como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação. *VERSÃO REVISADA*

Orientador: Prof. Dr. Sérgio Francisco da Silva

Catalão – GO
2021

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Araújo, Gustavo Evangelista
Análise e uso de técnicas para avaliação automática de redações no contexto do ENEM com foco na Competência 1 [manuscrito] / Gustavo Evangelista Araújo. – 2021.
81 p.: il.

Orientador: Prof. Dr. Sérgio Francisco da Silva
Monografia (Graduação) – Universidade Federal de Catalão, Instituto de Biotecnologia, Ciência da Computação, 2021.
Bibliografia.

1. Avaliação Automática de Redações. 2. Aprendizado de Máquina. 3. *Long short-term memory,*. I. Silva, Sérgio Francisco da, orient. II. Título.

CDU 004

Gustavo Evangelista Araújo

Análise e uso de técnicas para avaliação automática de redações no contexto do ENEM com foco na Competência 1

Monografia apresentada ao curso de Bacharelado em Ciência da Computação da Universidade Federal de Catalão.

Trabalho aprovado em xx de dezembro de 2021.

Sérgio Francisco da Silva
Orientador

Nádia Félix Felipe da Silva
Universidade Federal de Goiás

Márcio de Souza Dias
Universidade Federal Catalão

Catalão – GO
2021

“We can only see a short distance ahead, but we can see plenty there that needs to be done.”

Alan Turing

RESUMO

ARAÚJO, G. E.. **Análise e uso de técnicas para avaliação automática de redações no contexto do ENEM com foco na Competência 1**. 2021. 81 p. Monografia (Graduação) – Instituto de Biotecnologia, Universidade Federal de Catalão – , Catalão – GO.

Avaliação Automática de Redações (AAR) é uma alternativa válida para possíveis desgastes na rotina de preparo de candidatos do Exame Nacional do Ensino Médio (ENEM). Sistemas AAR podem fornecer uma correção efetiva, de baixo custo e que diminua a espera por correções humanas, contribuindo com o aprendizado e autonomia dos participantes. Diversos sistemas AAR foram propostos para ambas línguas inglesa e portuguesa, cujo a segunda ainda é pouco explorada. Nesta monografia, foi investigado o uso representações vetoriais de palavras estáticos (*word embeddings* Word2vec skip-gram, Word2vec GBOW e GloVe) e Redes Neurais Artificiais *Long Short-Term Memory* (LSTM) para avaliar a qualidade de redações no modelo ENEM para a Competência 1: Domínio da escrita formal da língua portuguesa. Os *word embeddings* foram treinados a partir de grande *corpora* da língua portuguesa, e os modelos desenvolvidos foram avaliados no *corpus* de redações Essay-BR, utilizando-se de medidas de erro *Mean Absolute Error* (MAE) e *Root-Mean-Square Error* (RMSE). Foi alcançado um RMSE de 0,148 quando comparado com a correção humana, a partir de um intervalo que varia entre 0 a 1, avanço significativo perante os resultados obtidos em trabalhos anteriores.

Palavras-chave: Avaliação Automática de Redações, Aprendizado de Máquina, *Long short-term memory,*.

ABSTRACT

ARAÚJO, G. E.. **Análise e uso de técnicas para avaliação automática de redações no contexto do ENEM com foco na Competência 1**. 2021. 81 p. Monografia (Graduação) – Instituto de Biotecnologia, Universidade Federal de Catalão – , Catalão – GO.

Automatic Essay scoring (AES) is a valid alternative for possible _ in the routine of preparing candidates for the National Hich School Exam (ENEM). AES systems can provide an effective, low-cost correction that reduces the wait for human corrections, contributing to the learning and autonomy of participants. Several AAR systems have been proposed for both English and Portuguese languages, the second of which is still scarcely explored. In this work, we investigated the use of static word embeddings (Word2vec skip-gram, Word2vec GBOW and GloVe) and Artificial Neural Networks *Long Short-Term Memory* (LSTM) to assess the quality of essays in the ENEM model for Competence 1: Adherence to the formal written norm of Portuguese.The word embeddings were trained using a large *corpora* of the Portuguese language, and the models developed were evaluated in the *corpus* of Essay-BR essays, using *error measures Mean Absolute Error* (MAE) and *Root-Mean-Square Error* (RMSE). An RMSE of 0.148 was achieved when compared to the human correction, from an interval that varies between 0 to 1, a significant advance compared to the results obtained in previous works.

Keywords: Automatic Essay Scoring, Machine Learning, Long short-term memory, LSTM.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

CAPÍTULO

1

INTRODUÇÃO

Órgãos nacionais e internacionais somam esforços em um compromisso para a formação humana integral e a construção de uma sociedade justa, democrática e inclusiva, orientada pelos princípios éticos e políticos. A UNESCO (2019) vê a melhoria das habilidades de alfabetização ao longo da vida como uma parte intrínseca do direito à educação. Em âmbito nacional, o Plano Nacional de Educação (PNE) estabelece metas para democratização do acesso à educação superior, com inclusão e qualidade (BRASIL, 2015a). Somado a isso, a Base Nacional Comum Curricular (BNCC) é o documento de caráter normativo que define as aprendizagens essenciais à todo aluno (BRASIL, 2018). Diante do desafio de garantir o acesso a educação de qualidade, previsto no artigo 26º da Declaração Universal dos Direitos Humanos de 1948 (ASSEMBLY *et al.*, 1948), uma das responsabilidades destes órgãos, para a execução efetiva dos planejamentos, é o monitoramento do avanço da qualidade do ensino.

O ENEM Exame Nacional do Ensino Médio - realizado anualmente desde 1998 pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP) - herda esta responsabilidade de avaliar a qualidade do ensino médio no país. Uma vez que as competências avaliadas também descrevem aptidões desejáveis para a atuação no mercado de trabalho e, sobretudo, vivência em sociedade, o ENEM se estabeleceu como prova para seleção de ingressantes ao ensino superior. Processo gradativo que começa em 2004, após sancionamento da lei do Programa Universidade para Todos (ProUni) (BRASIL, 2005), e se fortalece mais tarde em 2010 com o Sistema de Seleção Unificada (SISU) (BRASIL, 2010). Com o aumento do interesse por um exame único e nacional, o exame ganha destaque desde 2015 como o maior vestibular do país (TRAVITZKI, 2013) e o segundo maior do mundo (BRASIL, 2015b).

O ENEM é composto por 180 questões objetivas de múltipla escolha, divididas igualmente entre quatro áreas de conhecimento: Ciências da Natureza e suas tecnologias, Ciências humanas e suas tecnologias, Linguagens, Códigos e suas tecnologias, Matemática e suas tecnologias, além de uma redação, objeto de foco desta monografia. A influência da redação pode ser

observada em seu grande impacto na pontuação geral do ENEM. Pelo SISU, em faculdades que adotam peso igual entre os eixos, a redação carrega sozinha o valor de 20% na nota final. Ao observar os termos de adesão para os cursos com maior número de inscritos no país e relacionar aos pesos respectivos nas faculdades mais procuradas em 2020, percebe-se que o impacto pode ser maior (ENADE, 2020; UFPE, 2020; UFG, 2020; UFBA, 2020; UFRJ, 2020; UFF, 2020). Em muitos cenários, a redação apresenta superior à 20%, conforme pode ser visto na Tabela 1, através da soma total dos pesos de cada área, dividida pelo peso atribuído à redação.

Tabela 1 – Ponderação por área do conhecimento nas faculdades mais concorridas, nos cursos de maior busca

	UFPE						UFG						UFBA						UFRJ						UFF					
	R	N	H	L	M	Razão	R	N	H	L	M	Razão	R	N	H	L	M	Razão	R	N	H	L	M	Razão	R	N	H	L	M	Razão
Medicina	2	3	1	2	2	0,20	2	3	1,5	2	1,5	0,20	3	4	3	3	2	0,20	4	4	1	2	2	**0,31**	2	3	1	2	2	0,20
Direito	3	1	2	3	1	**0,30**	2,5	1	2	2,5	2	**0,25**	3	2	4	4	2	0,20	3	1	2	2	1	**0,33**	5	3	4	4	2	**0,28**
Administração	2,5	1	2	2	2,5	**0,25**	2,5	1	2	2,5	2	**0,25**	3	2	4	4	2	0,20	3	1	4	1	4	**0,23**	1	2	2	3	2	0,10
Enfermagem	3	2,5	1	2	1,5	**0,30**	2	3	1,5	2	1,5	0,20	3	4	3	3	2	0,20	3	2	2	2	1	**0,30**	4	4	2	2	2	**0,29**
Pedagogia	2,5	1,5	2	2	2,5	**0,24**	2,5	1	3	2,5	1	**0,25**	3	2	4	4	2	0,20	3	1	1	1	1	**0,43**	2	1	2	1	1	**0,29**
Psicologia	2,5	2	3	1,5	1	**0,25**	2,5	1	3	2,5	1	**0,25**	3	2	4	4	2	0,20	3	1	2	2	1	**0,33**	2	1	2	1	1	**0,29**
Educação Física	2	2,5	1,5	2	2	0,20	2,5	1	3	2,5	1	**0,25**	3	2	4	4	2	0,20	3	2	2	1	1	**0,33**	1	1	1	1	1	0,20
Medicina Veterinária	-	-	-	-	-	-	2	3	1	2	2	0,20	3	4	3	3	2	0,20	-	-	-	-	-	-	2	2	1	1	1	**0,29**
Engenharia Civil	1	3	1	2	3	0,10	1,5	2,5	1	2	3	0,15	3	4	2	2	4	0,20	3	4	1	2	5	0,20	2	4	1	3	5	0,13
Ciências Contábeis	2	1	2	2	3	0,20	2,5	1	2	2,5	2	**0,25**	3	2	4	4	2	0,20	3	1	4	1	4	**0,23**	3	2	2	2	4	**0,23**

R = Redação; N = Ciências da Natureza e suas tecnologias; H = Ciências Humanas e suas tecnologias; L = Linguagens, códigos e suas tecnologias; M = Matemática e suas tecnologias; Razão $= (R+N+H+L+M)/R)$

Fonte: Elaborada pelo autor.

Ainda neste ínterim, é importante destacar outro aspecto contribuinte para seu impacto. A Teoria de Resposta ao Item (TRI) é uma metodologia de avaliação usada pelo Ministério da Educação no ENEM, um conjunto de modelos matemáticos que busca representar a relação entre a probabilidade de o participante responder corretamente a uma questão, seu conhecimento na área em que está sendo avaliado e as características (parâmetros) dos itens. No modelo utilizado, não há um limite padrão inferior ou superior, o que significa que as proficiências dos participantes não variam entre zero e mil (BRASIL, 2021). Como a nota da redação não é calculada dessa forma, é a única área que permite que a nota dos candidatos varie entre zero e mil, fator que potencializa o peso da redação atribuído pelo curso. Mediante os argumentos apresentados, é possível inferir que a redação é uma área que oferece grande impacto para a nota final do participante, o que a torna o ponto focal de iniciativas que aprimorem o desempenho dos estudantes.

Um bom desempenho na redação normalmente exige que o estudante pratique constantemente a escrita e tenha respaldo de especialistas com uma correção e/ou comentários, o que permite aperfeiçoar as competências específicas e, então, pontuar mais. Diante da dinâmica que envolve o preparo dos estudantes para a prova de redação, fica evidente a interferência de diversos fatores socioeconômicos, como revela Melo *et al.* (2021), ao quantificar uma profunda e já conhecida desigualdade de acesso às experiências escolares. Também observou em seu estudo

que questões socioeconômicas como rendimento familiar e formação das mães têm forte relação com o desempenho dos estudantes tanto na prova objetiva quanto na redação, esta em menor proporção.

Essas desigualdades também se manifestam na sobrecarga dos docentes de forma geral, aqui colocadas na ótica da redação. Com base nos censos de 2009, 2013 e 2017, Carvalho (2018) reporta que a grande maioria dos docentes (acima de 70%) leciona para 6 turmas ou mais. Isso cria uma altíssima demanda de correções, planejamentos, atendimentos às dúvidas dos estudantes, entre outras atividades. Em consequência, esta demanda produz um retorno inconstante de avaliações e faz com que os alunos esperem por longo tempo por suporte. Neste cenário, muitas vezes, o atendimento fornecido pelo professor é insuficiente para suprir as necessidades do aluno, que sente um forte impacto negativo em seu aprendizado (BURSTEIN; CHODOROW; LEACOCK, 2003). Uma vez que se descobriu que a adequação do *feedback* é altamente específica ao indivíduo e/ou à situação específica (HYLAND, 1998), é essencial considerar um método eficaz para analisar um grande número de redações.

Na busca por superar os obstáculos apresentados, alternativas como o auxílio de professores ou corretores particulares tanto no modelo presencial quanto à distância (em plataformas *online*) colocam no estudante a responsabilidade de amenizar as desigualdades estruturais que lhe são impostas. Ainda que se faça necessário um olhar profundo para a superação dessas marcas no contexto escolar, há espaço para soluções que atuem a curto prazo na democratização de acesso e permanência nas instituições de ensino.

Essa circunstância converge a atenção para a Avaliação Automática de Redações (AAR), (ou também *Automatic Essay Scoring* - AES), uma tecnologia que abranda o esforço do docente assim como incentiva a autonomia do estudante. Seu principal objetivo é pontuar o desempenho na redação de forma precisa e instantânea. Esta área de pesquisa começou com Page (1967), e apesar de ser investigada por mais de 50 anos, ainda toma muita atenção da comunidade científica pelo seu valor comercial e educacional, cujo estado-da-arte aponta avanços promissores, mas não é suficientemente precisa para ser aplicada em larga escala na língua portuguesa e, sobretudo, no modelo do ENEM (KE; NG, 2019; COSTA; OLIVEIRA; JÚNIOR, 2020).

1.1 Problema de pesquisa

Motivado pelos resultados recentes obtidos na língua inglesa com o uso de uma arquitetura híbrida entre *word embeddings* (técnicas de representação semântica de palavras em um espaço vetorial) e características extraídas manualmente (baseadas em tamanho, sintáticas, baseada em palavras e índices de legibilidade), busca-se combiná-las em uma arquitetura de Rede Neural Artificial *Long-Short Term Memory* (RNA-LSTM) para produzir um resultado competitivo na língua portuguesa. Desta forma, tem-se como foco a primeira das cinco competências do ENEM (BRASIL, 2020):

Tabela 2 – Competências do Exame Nacional do Ensino Médio (ENEM)

Competência 1: Domínio da escrita formal da língua portuguesa;
Competência 2: Compreender o tema e não fugir do que é proposto;
Competência 3: Selecionar, relacionar, organizar e interpretar informações, fatos, opiniões e argumentos em defesa de um ponto de vista;
Competência 4: Conhecimento dos mecanismos linguísticos necessários para a construção da argumentação;
Competência 5: Respeito aos direitos humanos;

Fonte: (BRASIL, 2020)

No que tange a esta escolha, é necessário frisar que cada competência de avaliação das redações do ENEM se ocupa de questões específicas e que a nota final de uma redação é a soma das notas atribuídas em cada uma das competências. As notas das competências não estão atreladas entre si, ou seja, se uma redação for avaliada no nível máximo da Competência 1, ela não necessariamente deve receber o nível máximo nas outras competências também (INEP, 2020). Assim sendo, são bem-vindas soluções parciais à tarefa, inclusive a que se edifica esta monografia.

Não obstante, esta competência se apresenta computacionalmente tratável, uma vez que em sua maioria, oferece critérios objetivos (como a quantidade de desvios de convenções da escrita e gramaticais) para a análise da adequação da redação com a norma padrão da língua, os quais serão melhor explorados no Capítulo 2.

Também é relevante considerar que esta competência tem correlação aproximada com o interesse central dos modelos de redações para a língua inglesa: *Test of English as a Foreign Language* (TOEFL), *Cambridge Learner corpus First Certificate in English* (CLC-FCE) e *Automated Student Assessment Prize* (ASAP), objeto de estudo dos trabalhos correlatos da língua inglesa com maiores resultados na área (KE; NG, 2019; UTO, 2021), o que pode facilitar a aplicação de métodos bem sucedidos nela.

Para tanto, serão investigadas técnicas atuais de Aprendizado de Máquina (AM) e PLN aplicadas a tarefa de AAR. A investigação será norteada pela seguintes questões:

1. Qual a eficácia de uma Rede Neural Artificial LSTM para inferir e avaliar informações gramaticais e semânticas da Competência 1 de redações no modelo ENEM?

2. Qual a eficácia da aplicação de *word embeddings* aprendidas em grandes *corpus* da Língua Portuguesa, para a caracterização das palavras que constituem uma redação?

3. Qual a eficácia da combinação de características extraídas manualmente (ex.: características léxicas, baseadas em tamanho e frequência de elementos textuais)?

1.2 Objetivo Geral

O objetivo geral deste trabalho é desenvolver e implementar uma arquitetura de RNA-LSTM e *word embeddings* com a utilização de características extraídas manualmente para avaliar automaticamente a Competência 1 de redações no modelo do Exame Nacional do Ensino Médio.

1.2.1 Objetivos específicos

Este trabalho tem como objetivos específicos:

- Investigar os modelos de previsão baseados em RNA-LTSM para as línguas inglesa e portuguesa, com destaque para as que utilizam *word embeddings*.
- Pré-processar o *corpus* criado por Marinho, Anchieta e Moura (2021) com 4.570 redações nos moldes do ENEM, efetuando transformações necessárias.
- Experimentar e comparar as (*word embeddings*) Word2Vec e GloVe extraídas de grandes *corpus* da Língua Portuguesa para a AAR, e analisar se estas representações possibilitam analisar aspectos estabelecidos pela Competência 1 do ENEM.
- Definir um modelo de RNA-LSTM para predição de avaliação de redações e ajustar seus parâmetros, com base nos trabalhos correlatos e análise empírica.
- Investigar a utilização de características extraídas manualmente.
- Comparar os resultados da avaliação automática com as avaliações feitas por especialistas humanos, com base no estudo das métricas mais utilizadas para a tarefa de AAR.

1.3 Organização do Texto

Esta monografia é estruturada da seguinte forma: O presente Capítulo contextualizou e apresentou o problema de Avaliação Automática de Redações e os objetivos desta pesquisa. No Capítulo 2, são abordados os fundamentos teóricos de Processamento de Linguagem Natural dando ênfase as técnicas de pré-processamento e (*word embeddings*). Posteriormente, no Capítulo 3 são desenvolvidos os conceitos para a construção de uma RNA-LSTM. Em seguida, o Capítulo 4 descreve uma gama de revisões e trabalhos encontrados na literatura inglesa e portuguesa com foco na Avaliação Automática de Redações. No Capítulo 5 é especificada a metodologia utilizada para a Avaliação Automática de Redações, são apresentados e discutidos os resultados obtidos. E por fim, no Capítulo 6 são apresentadas as conclusões desta monografia e a sugestão de trabalhos futuros.

CAPÍTULO

2

FUNDAMENTOS DE PROCESSAMENTO DE LINGUAGEM NATURAL

O marco inicial para o Processamento de Linguagem Natural [1] emergiu em 1940, em contexto da Segunda Guerra Mundial, manifestando-se através da necessidade de decifrar mensagens interceptadas na mesma velocidade em que eram produzidas e re-codificadas, feito humanamente impossível até mesmo para especialistas. A *Bombe*, máquina que alcançou este objetivo, garantiu acesso aos Aliados à informações sobre a movimentação de tropas e recursos do Eixo, vantagem estratégica ao qual teve profundo impacto na vitória do primeiro sobre o segundo (JACK, 2012; JURAFSKY; MARTIN, 2009). Alan Turing, um dos especialistas envolvidos em sua criação, deu continuidade às contribuições para a área: publicou em 1950 o artigo *"Computing Machinery and Intelligence"*(TURING; HAUGELAND, 1950), no qual cunhou o termo PLN e o definiu como uma das capacidades necessárias para uma máquina ser definida como inteligente. De acordo com Russell e Norvig (2010), ambos contextos remontam a origem dos dois principais interesses em construir agentes capazes de processar linguagens naturais: **se comunicar com seres humanos** e **adquirir informação a partir da língua natural**, este último como foco desta monografia.

Apesar deste interesse compartilhado com as primeiras abordagens, a escolha das informações a serem extraídas de um texto definem o grau de dificuldade de seu processamento. Enquanto que o objetivo de Turing era a decodificação de mensagens interceptadas, hoje a PLN também se ocupa em extrair o significado, sentimentos, padrões, veracidade, dentre outras informações de textos e falas em língua natural. Este cenário exige reflexão crítica sobre a relação entre a linguagem, pensamento e entendimento, uma vez que a língua natural, depende muito das habilidades linguísticas, conhecimento do domínio de interesse e expectativas nesse

[1] Também pode ser conhecida como, Linguística Computacional, Processamento de Língua Natural e Engenharia das Línguas Naturais. Adota-se aqui o termo *Processamento de Linguagem Natural* por ser este mais difundido e tradicionalmente usado no Brasil.

domínio. Compreender a linguagem não é apenas a transmissão de palavras: também requer inferências sobre os objetivos, conhecimento e suposições do falante, bem como sobre o contexto da interação (LUGER, 2005).

Visando uma introdução e reflexão ampla do assunto de Avaliação Automática de Redações (AAR), são descritas as principais categorias de técnicas usadas na construção de sistemas e ferramentas para AAR, desde a representação de palavras (*word embeddings*), pré-processamento textual, extração de características (*feature engineering*) até as métricas com maior adesão.

2.1 Representação de palavras (*Word embeddings*)

As definições subsequentes desta seção são fundamentadas majoritariamente nos trabalhos de Pilehvar e Collados (2020).

Dado a palavra ”casa”, ao ser armazenada em um computador, essa palavra torna-se uma sequência de 4 caracteres: ”c”, ”a”, ”s” e ”a”. Entretanto, computadores apenas entendem sequências binárias (0 e 1), ou seja, cada caractere tem que ser armazenado como um padrão de *bits*. O número de *bits* depende da codificação. Por exemplo, a estendida codificação ASCII precisa de 8 *bits* para armazenar cada caractere. Assim, a palavra, ”casa” é *representada* como uma sequência binária de 32 algarismos [2]. Já a palavra ”lar” terá uma representação de 24 algarismos 0 ou 1[3]. De acordo com Pilehvar e Collados (2020), essa abordagem não é um caminho favorável para representar semanticamente as palavras, devido as seguintes limitações:

1. A representação não incorpora informações semânticas das palavras, *e.g.* mesmo sejam sinônimas, ”casa”, ”lar” e ”residência” terão representações totalmente diferentes.

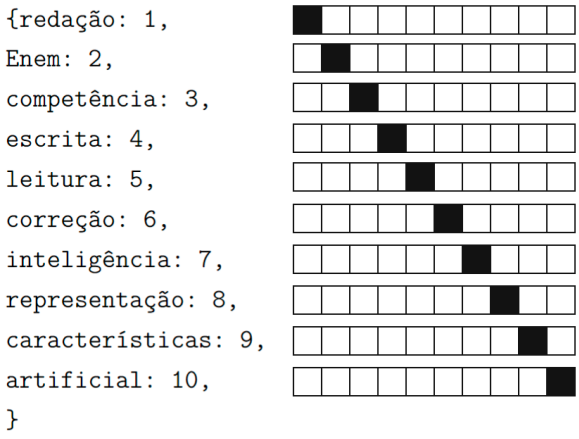
2. A representação é dada pelos caracteres. Ou seja, o tamanho da representação depende da quantidade de caracteres da palavra. A variabilidade do tamanho são propriedades não desejáveis que futuramente complicam a comparação da representação de diferentes palavras.

One-hot

A primeira iniciativa da PLN tradicional resolveu parte da segunda limitação supracitada através da técnica *one-hot*, que ao invés de representar caracteres, representa diretamente as palavras. Nesta, são utilizados vetores de tamanho fixo em um padrão binário (0 e 1). Ficou conhecida como uma representação simplória de palavras, que é ilustrada na Figura 1 e descrita como se segue. Assumindo-se 100 palavras em vocabulário, primeiro, é associado um índice

[2] codificação ASCII para ”casa”: 01100011 01100001 01110011 01100001
[3] codificação ASCII para ”lar”: 01101100 01100001 01110010

Figura 1 – Representação one-hot



Fonte: adaptada de Pilehvar e Collados (2020)

(entre 1 e 100) para cada uma. Em segundo, cada palavra é representada em um vetor de 100-dimensões, todas posições preenchidas com 0, exceto a do índice da palavra, preenchida com 1 (PILEHVAR; COLLADOS, 2020).

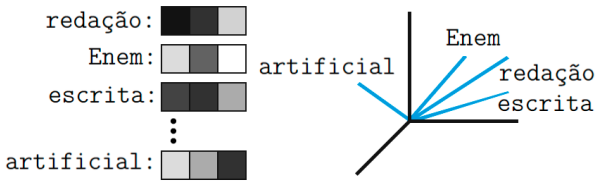
Neste exemplo, podemos perceber uma representação bastante esparsa, uma vez que temos, em cada vetor, apenas um *bit* 1 para o restante 0. Esta ainda sofre pela primeira limitação descrita, uma vez que não há nenhuma representação de similaridade entre as palavras, ou até pior, palavras como ”casa” e ”casas” são representadas de forma completamente diferentes. Somado a isso, os vetores de cada palavra crescem à medida de do vocabulário, que em um vocabulário de uma língua natural, podemos esperar centenas de milhares de palavras, algo computacionalmente difícil de processar. Apesar de sua simplicidade, construiu grande parte dos fundamentos para a técnica *Vector Space Models (VSM)* (PILEHVAR; COLLADOS, 2020).

Vector Space Models

Vector Space Model (VSM), proposto pela primeira vez por Salton, Wong e Yang (1975), fornece uma solução para as limitações da representação *one-hot*. Neste modelo, os objetos são representados como vetores em um espaço contínuo multidimensional imaginário. No PLN, o espaço é geralmente chamado de **espaço semântico** e a representação dos objetos é chamada de **representação distribuída**. Deste modo, esse modelo passa de uma natureza local e discreta para uma distribuída e contínua. Com este avanço, é introduzido a noção de similaridade de duas palavras (vetores), medida através da distância no espaço. Não obstante, o problema do crescimento proporcional ao vocabulário também é resolvido, uma vez que um grande vocabulário m pode caber em um espaço vetorial n-dimensional, onde $n << m$.

A Figura 2 demonstra um espaço semântico tridimensional que representa quatro palavras com seus vetores correspondentes. Considerando-se um cenário realista, é esperado que centenas de milhares de palavras sejam dispostas nesse espaço, e que as suas distâncias correspondam à

Figura 2 – Representação Vector Space Model



Fonte: Adaptada de Pilehvar e Collados (2020)

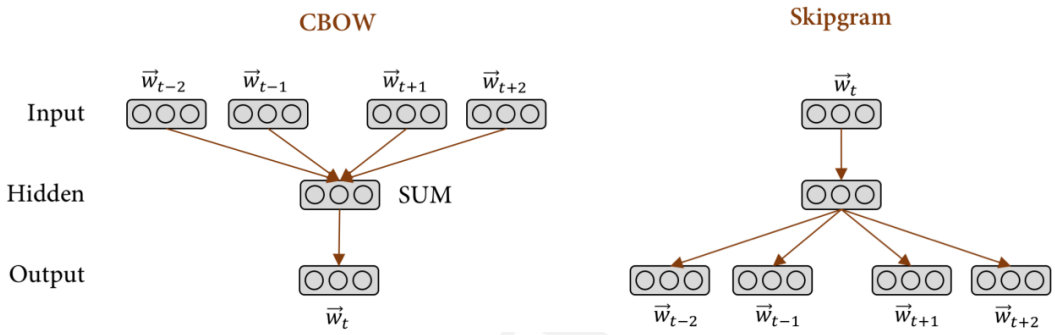
sua semelhança semântica. Para tanto, estes espaços são construídos automaticamente através da análise da coocorrência de palavras em um grande *corpora* de textos. Com base no que foi denominado por Firth (1957) como **hipótese distribucional**, *"uma palavra é caracterizada pelas palavras que a acompanham"*, ou seja, palavras que aparecem em contextos similares tendem a ter significados similares. Por exemplo, *Júpiter* e *Vênus* tendem a ser relacionadas semanticamente, dado que estas palavras aparecem em contextos similares, como em uma descrição do *sistema solar, estrelas, planetas* e *astronomia*. Desta forma, pode-se coletar estatísticas de coocorrência de palavras e com estas inferir relações semânticas.

Através do VSM, edificaram-se os modelos estatísticos tradicionais baseados em matrizes de frequência de palavras, usualmente conhecidos como métodos *count-based*. Nestes, foram atribuídas técnicas de normalização de coocorrência de palavras e redução de dimensionalidade para a identificação da relação semântica entre as palavras, processo computacionalmente custoso (PILEHVAR; COLLADOS, 2020). Posteriormente, sob a característica das redes neurais artificiais aprenderem representações densas sem a etapa de redução de dimensionalidade, foi possível apresentar uma alternativa rápida e viável em contraste com as técnicas *count-based*. Assim, os modelos baseados em redes neurais emergiram em tarefas de PLN, nomeados como modelos preditivos *(do inglês, preditive models)* por terem como proposta inicial a predição da próxima palavra de uma sentença ou a palavra ausente. *Word embeddings* são popularizados enfim em 2013 pela técnica Word2vec, com a apresentação de uma alternativa com menor complexidade computacional.

Word2vec

Desenvolvido por Mikolov *et al.* (2013a), Word2vec é uma arquitetura neural treinada para modelagem da língua, utilizada em sua maioria para a identificação de palavras ausentes em uma sentença. Dois modelos de aprendizado foram desenvolvidos: *Continuous Bag-Of-Words* (CBOW) e *Continuous Skip-gram*. O primeiro prevê a palavra central dado as palavras vizinhas, minimizando a função perda (Equação 2.1), onde w_t é a palavra-alvo e $W_t = w_{t-n}, \ldots, w_t, \ldots, w_{t+n}$ representa a sequência de palavras em contexto. Já o modelo *continuous skip-gram* prevê as

Figura 3 – Representação Word2vec



Fonte: Pilehvar e Collados (2020)

palavras vizinhas com base na palavra central, ambos apresentados na Figura 3.

$$E = -\log(p(\overrightarrow{w}_t|\overrightarrow{W}_t)) \tag{2.1}$$

Contudo, ambos modelos aprendem com seu contexto local de uso, no qual é definido por um janela de palavras vizinhas. Essa janela é um parâmetro configurável do modelo. Conforme Goldberg e Hirst (2017), o tamanho da **janela deslizante** tem um forte efeito sobre o tipo de características extraídas. Janelas grandes tentem a produzir características de relações entre tópicos, enquanto que janelas pequenas tentem a produzir características mais relacionadas a sintaxe. Através do Word2vec, foi demonstrado como *word embeddings* são capazes de inferir relações semânticas entre pares de palavras, como demonstrado na Tabela 3.

Tabela 3 – Exemplo de relação semântica entre pares de palavras com Word2vec

Relação	Exemplo 1	Exemplo 2	Exemplo 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

Fonte: Adaptado de (MIKOLOV *et al.*, 2013b)

Global Vectors (GloVe)

Uma vez que as inferências semânticas do método Word2vec são limitadas ao tamanho da janela deslizante, ele possui dificuldade na identificação de estatísticas globais do *corpus* - algo que os métodos *count-based* se destacavam. Em resposta, o método *Global Vectors for Word Representation* (GloVe), desenvolvido por Pennington, Socher e Manning (2014), mantém e aprimora a detecção de relação semântica do Word2vec. O pensamento principal que motivou a construção deste modelo é a observação de que **as razões das probabilidades de coocorrência entre as palavras têm o potencial de codificar alguma forma de significado**. É demonstrado

na Tabela 4, probabilidades reais de um corpus de 6 bilhões de palavras, com o objetivo de demonstrar as probabilidades de co-ocorrência para as palavras *ice* (gelo) e *steam* (vapor) com várias palavras de teste do vocabulário.

Tabela 4 – Exemplo de relação semântica entre pares de palavras com GloVe

Probabilidade e Razão	k = *solid*	k = *gas*	k = *water*	k = *fashion*
P(K | *ice*)	$1.9 * 10^{-4}$	$6.6 * 10^{-5}$	$3.0 * 10^{-3}$	$1.7 * 10^{-5}$
P(K | *steam*)	$2.2 * 10^{-5}$	$7.8 * 10^{-4}$	$2.2 * 10^{-3}$	$1.8 * 10^{-5}$
P(K | *ice*)/P(K | *steam*)	8.9	$8.5 * 10^{-2}$	1.36	0.96

Fonte: Pennington, Socher e Manning (2014)

Como demonstrado acima, a relação dessas palavras pode ser examinada estudando a razão de suas probabilidades de co-ocorrência com várias palavras de prova k. Para palavras k relacionadas a *ice*, mas não ao *steam*, como k = *solid* (sólido), esperamos que a razão P_{ik}/P_{jk} seja grande. Da mesma forma, para palavras k relacionadas ao *steam*, mas não a *ice*, como k = *gas* (gás), a razão deve ser pequena. Para palavras k como *water* (água) ou *fashion* (moda), que estão relacionadas tanto a *ice* quanto a *steam*, ou a nenhum, a proporção deve ser próxima de um.

Contextualized Embeddings

Desde a sua introdução, as *word embeddings* pré-treinadas dominaram o campo da representação semântica. Elas têm sido um componente-chave na maioria dos sistemas neurais de Processamento de Linguagem Natural. Normalmente, um sistema de PLN é alimentado com *word embeddings* pré-treinadas em grandes *corpora* para todas as palavras do vocabulário da língua-alvo. Na camada de entrada, o sistema procura a representação vetorial de uma determinada palavra e propaga às camadas subsequentes (em oposição a uma representação *one-hot*). Passar de representações *one-hot* para um espaço contínuo de *word embeddings* geralmente resulta em maior poder de generalização do sistema e, portanto, melhor desempenho.

No entanto, *word embeddings* pré-treinadas, como Word2vec e GloVe, calculam uma única representação estática para cada palavra. A representação é fixa; é independente do contexto em que a palavra aparece. Por exemplo, a mesma representação vetorial seria usada na camada de entrada para a palavra "banco", mesmo que fosse usada em contextos diferentes, os quais teriam desencadeado outros significados como “banco da praça”, “banco de dados” e “agência do banco”.

Ao contrário das *word embeddings* estáticas (independentes do contexto), as incorporações contextualizadas (dinâmicas) não são fixas: elas se adaptam à sua representação ao contexto. O modelo de representação contextualizado processa o contexto da palavra-alvo (célula na figura) e gera sua incorporação dinâmica. Um dos mais importantes benefícios desta recente abordagem é a integração perfeita na maioria dos modelos de processamento de linguagem

neural. Curiosamente, os Word embeddings contextualizados não apenas podem capturar vários papéis semânticos de uma palavra, mas também suas propriedades sintáticas.

Apesar do conhecido avanço destas técnicas para as tarefas de PLN, com avanços norteados pelos modelos *Embeddings from Language Models* (ElMo) (PETERS *et al.*, 2018) e, principalmente, *Bidirectional Encoder Representations from Transformers* (BERT) (DEVLIN *et al.*, 2018), esta monografia se ocupa com a utilização dos modelos estáticos de *word embeddings*: Word2vec e GloVe. A análise desta escolha está no Capítulo 4. Ainda assim, encaminha-se a implementação futura destes no Capítulo 6.

2.2 Avaliação Automática de Redações (AAR)

Sistemas AAR[4] usam técnicas estatísticas, de Aprendizado de Máquina (AM) e PLN com suas vantagens amplamente discutidas desde os trabalhos de Page (1967) e Page (1968). Dentre essas, pode-se citar a redução das atividades intensas de avaliação de redações por profissionais, a capacidade de avaliar grandes quantidade de redações com baixo custo, a aplicação de um critério consistente por um sistema que seja menos sujeito a subjetividade humana e não é afetado por cansaço ou aborrecimentos aos quais os humanos são submetidos, assim como aqui se enfatiza, o fortalecimento da autonomia e independência do participante na rotina de estudo para o exame.

A produção de redações é habitualmente utilizada como meio de seleção para vagas em cursos de formação e empregos. Dependendo de seu objetivo, diferentes critérios são utilizados. Os exames que norteiam as técnicas de AAR na língua inglesa (detalhados no Capítulo 4), não se distinguem do ENEM apenas pelas diferenças da língua, mas também por centralizar a análise da proficiência na língua inglesa e interpretação de texto. No que tange ao objetivo do exame brasileiro, destaca-se para o contexto dessa monografia, a sétima das dez competências gerais da Educação Básica (Educação Infantil, Ensino Fundamental e Ensino Médio):

> *"Argumentar com base em fatos, dados e informações confiáveis, para formular, negociar e defender ideias, pontos de vista e decisões comuns que respeitem e promovam os direitos humanos, a consciência socioambiental e o consumo responsável em âmbito local, regional e global, com posicionamento ético em relação ao cuidado de si mesmo, dos outros e do planeta" (MEC, 2018).*

[4] É importante salientar que existem outros nomes relacionados a esta tarefa: **Correção Automática de Redações**, **Avaliação Assistida por Computador**. Adota-se o nome de Avaliação Automática de Redações pela correspondência ao interesse deste trabalho em única e exclusivamente **pontuar** redações com base nos critérios da redação do ENEM. Direciona-se os interessados nas tarefas relacionadas a leitura dos trabalhos correlatos, descritos no Capítulo 4.

É crível que a responsabilidade da análise desta competência seja atribuída à redação, sobretudo, por ser o único eixo que avalia o aluno pelo gênero textual dissertativo-argumentativo, em contraste com as questões objetivas das demais áreas. Deste modo, ainda que se faça necessária uma análise profunda da relação entre as competências avaliadas nos exames de ambas línguas, é plausível que o ponto de partida seja um denominador comum ao qual as técnicas de uma possam ser reproduzidas com similar desempenho na outra: o **domínio da escrita formal da língua**.

Como apresentado no capítulo anterior, um bom desempenho na redação faz a diferença no resultado final do participante do ENEM. Para obter uma nota alta nesta área, o candidato deve seguir alguns critérios avaliados pelos organizadores da prova. O Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP), que realiza o exame, a descreve em 6 pontuações (0 a 200), com incremento de 40 pontos de uma para o outra, como é apresentado na Tabela 5. (BRASIL, 2020)

Tabela 5 – Matriz de Referência da Competência 1

Demonstra excelente domínio da modalidade escrita formal da Língua Portuguesa e de escolha de registro. Desvios gramaticais ou de convenções da escrita serão aceitos somente como excepcionalidade e quando não caracterizem reincidência.	200
Demonstra bom domínio da modalidade escrita formal da Língua Portuguesa e de escolha de registro, com poucos desvios gramaticais e de convenções da escrita. Pode haver poucos desvios gramaticais de menor gravidade, tais quais pontuação, ortografia e acentuação. Raramente, desde que não haja regularidade, pode haver alguns desvios relacionados à falta de concordância verbal ou nominal.	160
Demonstra domínio mediano da modalidade escrita formal da Língua Portuguesa e de escolha de registro, com alguns desvios gramaticais e de convenções da escrita.	120
Demonstra domínio insuficiente da modalidade escrita formal da Língua Portuguesa, com muitos desvios gramaticais, de escolha de registro e de convenções da escrita	80
Demonstra domínio precário da modalidade escrita formal da Língua Portuguesa, de forma sistemática, com diversificados e frequentes desvios gramaticais, de escolha de registro e de convenções da escrita.	40
Demonstra desconhecimento da modalidade escrita formal da Língua Portuguesa	0

Fonte: adaptada de INEP (2020)

Para a análise desta competência, o avaliador deve se atentar aos seguintes aspectos: a **estrutura sintática** e a presença de **desvios**. Essa avaliação é pautada pelo que dispõe a norma-padrão e deve levar em consideração que o domínio dessa norma está estratificado em níveis que contemplam tanto o léxico e a gramática quanto a fluidez da leitura, a qual pode ser prejudicada ou valorizada por uma construção sintática ruim ou boa (INEP, 2020). A critério de contraste, os Anexos 1 e 2 representam, respectivamente, redações de nível 0 e 5 nesta competência. Ainda sob a análise de INEP (2020), a primeira é caracterizada pela "estrutura sintática inexistente (independente da quantidade de desvios)". Neste nível estão as redações com letras ou palavras formadas, mas compostas majoritariamente por frases ininteligíveis. A segunda, por sua vez, é pontuada no nível 5, apresentando apenas uma única falha da estrutura sintática (duplicação da

palavra "pela") e dois desvios (grafia de "afim"e falta de acentuação em "critica"), quantidade de falhas que representa o limite exato da tolerância para cada tipo de erro neste nível.

De modo geral, a Competência 1 preconiza a **gramática normativa**, diferente da **gramática internalizada** (a qual descreve o reconhecimento de frases em Língua Portuguesa a qualquer falante desta língua e produzir frases que sejam reconhecíveis por outros falantes) e a **gramática descritiva** (Baseada no que os falantes de fato produzem no uso cotidiano da língua), (INEP, 2020). Desta forma, é esperado que se tenha definições objetivas que facilitem a distinção clara entre os níveis de pontuação, e portanto, forneça definições operacionalmente satisfatórias para a avaliação da nota.

2.2.1 Feature engineering

Uma grande quantidade de trabalhos em AAR envolve o projeto de métodos de extração de características. (UTO, 2021; KE; NG, 2019; DIKLI, 2006; COSTA; OLIVEIRA; JÚNIOR, 2020) Enquanto existem os modelos neurais desenvolvidos recentemente para AAR para evitar a necessidade de engenharia de características, acredita-se nesta pesquisa que o desenvolvimento de características continuará a desempenhar um papel crucial no AAR pesquisa.

Primeiro, para modelos neurais serem eficazes, precisam ser treinados em uma grande quantidade de dados anotados, que neste caso são . Mesmo acreditando que há dados suficientes para treinar modelos AAR precisos para inglês, o mesmo não é verdade para a grande maioria das línguas naturais. Para construir AAR para essas linguagens, a maneira mais prática é empregar uma abordagem baseada em características. Em segundo lugar, mesmo para o inglês, a quantidade de dados disponíveis para treinamento de AAR específicos de dimensão sistemas é bastante limitado. Até que se tenha corpora anotados maiores, acredita-se que a engenharia de características continuará sendo uma etapa importante do processo. Terceiro, enquanto muitos modelos de pontuação holística neural alcançaram o estado-da-arte em resultados, é possível que esses modelos possam ser melhorados incorporando características artesanais obtidas por meio de engenharia de características. No geral, acredita-se que com base em características as abordagens e as abordagens neurais devem ser vistas como abordagens complementares e não concorrentes.(UTO; XIE; UENO, 2020)

Essa abordagem exige grande esforço de especialistas na construção de características suficientemente descritivas para o processo de pontuação de uma redação. No entanto, para o contexto do ENEM, no intuito de prover uma correção idônea e equivalente entre a grande diversidade de corretores, criaram-se documentos norteadores que dão o ponto de partida para o processo de extração de características objetivas e possivelmente computáveis. Nas Tabelas 6 e 7 são descritos, respectivamente, a grade específica sob o critério dos avaliadores e a descrição do que se entende por erros de estrutura sintática e desvios.

Tabela 6 – Grade específica da Competência 1

Estrutura sintática excelente (no máximo, uma falha) E, no máximo, dois desvios	200
Estrutura sintática boa E poucos desvios	160
Estrutura sintática regular E alguns desvios	120
Estrutura sintática deficitária OU muitos desvios	80
Estrutura sintática deficitária com muitos desvios	40
Estrutura sintática inexistente (independentemente da quantidade de desvios)	0

Fonte: Adaptada de INEP (2020).

Tabela 7 – Erros comuns à Competência 1

Problemas comuns de estrutura sintática	Truncamento de períodos, Justaposição de orações e/ou períodos, Excesso, duplicação ou ausência de palavras (elementos sintáticos)
Desvios de convenções da escrita	Acentuação, Ortografia, Hífen, Maiúsculas/-minúsculas, Separação silábica (translineação)
Desvios gramaticais	Regência, Concordância, Pontuação, Paralelismo sintático, Emprego de pronomes, Crase;
Desvios de escolha de registro	Informalidade/marca de oralidade
Desvios de escolha vocabular	Escolhas lexicais imprecisas

Fonte: Adaptada de INEP (2020).

Tipos de avaliação

Os tipos de AAR fornecem especificações interessantes à busca por correlacionar técnicas entre as línguas inglesa e portuguesa. Foram classificadas por Uto (2021) seguindo os seguintes tipos:

Prompt-specifc holistic scoring Abordagem mais comum na língua inglesa, no qual um modelo é treinado para prever a pontuação total (abordagem holística) de uma redação inédita no mesmo tema em que foi treinado (específica ao tema).

Prompt-specifc trait scoring Busca prever pontuações de características específicas como: legibilidade, argumentação, estrutura sintática, dentre outros, para cada redação no mesmo tema em que são avaliadas. A análise de características específicas é utilizada no intuito de fornecer um retorno mais detalhado para o aluno, cumprindo fins educacionais.

Cross-prompt holistic scoring Nesta tarefa, um modelo AAR é avalia, de forma holística, redações para temas diferentes do qual foi treinado, onde o modelo treinado é transferido para um tema de destino. Esta tarefa de avaliação em cruzada de temas, recentemente atraiu a atenção porque é difícil obter um número suficiente de redações avaliados escritos para um tema de destino na prática. A tarefa AAR *cross-prompt* está relacionada a tarefas de

adaptação e transferência de aprendizado de domínio, que são amplamente estudadas em campos de aprendizado de máquina.

Cross-prompt trait scoring Esta tarefa envolve a predição de pontuações em características específicas, treinadas em redações cujos temas são diferentes dos temas do conjunto de teste. Esta representa a abordagem utilizada nesta monografia ao qual se indica a leitura do Capítulo 5 para melhor entendimento.

CAPÍTULO

3

FUNDAMENTOS DE REDES NEURAIS ARTIFICIAIS

A inteligência humana é vista por nós como um importante fator evolutivo. O epíteto específico *sapiens* (do latim, **sábio**) marca a valorização das faculdades mentais e o reconhecimento da vantagem adaptativa que elas nos proporcionam. Ainda que se tenha muito a descobrir sobre como nos percebemos, como pensamos, compreendemos, prevemos e manipulamos um mundo muito maior e complicado que nós mesmos, o campo da Inteligência Artificial (IA) se apresenta como um passo além: a busca pela *criação* de entidades inteligentes. (RUSSELL; NORVIG, 2010)

Muitos fundamentos da Inteligência Artificial têm sido de grande importância na filosofia desde os tempos antigos, sendo abordados por Aristóteles, São Tomás de Aquino, Guilherme de Ockham, René Descartes, Thomas Hobbes e Gottfried W. Leibniz. Questões como: "Quais são as operações cognitivas básicas?","Como a mente funciona?" e "O raciocínio pode ser automatizado?" estruturaram o arcabouço filosófico para a formulação da pergunta fundamental: **"É possível construir um sistema inteligente?"**. Com os experimentos possibilitados pelos primeiros computadores no século XX, esta questão foi modelada sob um novo ponto de vista: "Quando podemos dizer que um sistema construído por um ser humano é inteligente?" (FLASIŃSKI, 2016)

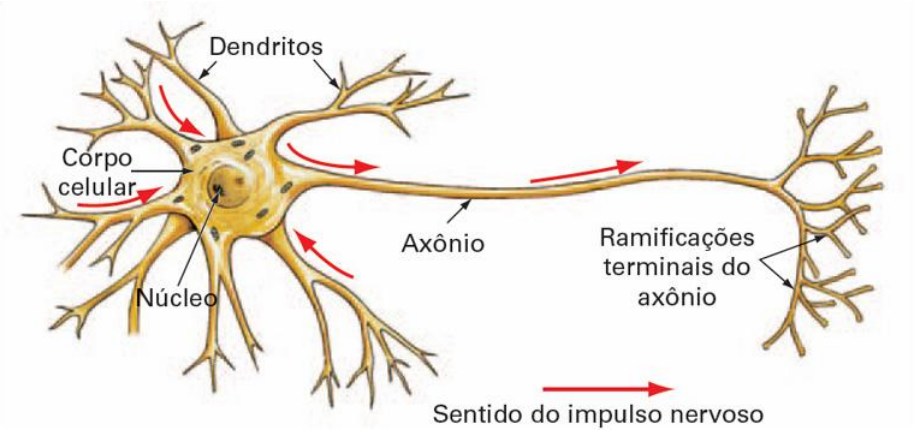
A busca histórica por uma definição satisfatória da Inteligência Artificial e pela construção de entidades inteligentes ainda não chegou ao fim. Por outro lado, ela forneceu abordagens suficientemente descritivas para o objetivo deste trabalho. Antes de descrever exclusivamente a arquitetura de rede neural utilizada nessa monografia, este capítulo busca desenvolver os conceitos utilizados em sua construção.

3.1 Rede Neural Artificial (RNA)

O cérebro humano serviu de inspiração para a elaboração das redes neurais artificiais pela sua organização complexa de neurônios. O cérebro humano é um processador complexo, não linear e paralelo, com a capacidade de realizar tarefas como reconhecimento variados tipos de padrões (como visuais, sonoros, temporais, entre outros), controle motor, planejamento, de forma mais rápida e precisa que os computadores atuais (HAYKIN, 2009).

As redes neurais artificiais - também chamadas apenas de redes neurais - são técnicas de aprendizado de máquina que buscam simular o aprendizado provido por sistemas neurais biológicos. O sistema nervoso humano contém células chamadas neurônios (Figura 4) que trocam informações entre si a partir dos dendritos e axônios, e as interconexões entre eles são chamadas de sinapses. A força (peso) das conexões mudam de acordo com os estímulos recebidos, e isso caracteriza o aprendizado em organismos vivos CITAR. Esse mecanismo é simulado nas redes neurais artificiais utilizando unidades computacionais que imitam os neurônios, que são conectadas às outras a partir de pesos que representam as forças das sinapses.

Figura 4 – Representação de um neurônio.



Fonte: (GOODFELLOW; BENGIO; COURVILLE, 2016)

Uma rede neural calcula uma função das entradas, propagando os valores calculados dos neurônios de entrada para os neurônios de saída, utilizando os pesos como intermediários para inibir ou reforçar os sinais. No modo supervisionado, o aprendizado ocorre ajustando esses pesos de acordo com padrões de treinamento que consistem de estímulos recebidos (sinais de entrada) e os dados corretos de saída. Dessa forma é possível utilizar uma entrada, comparar o resultado calculado com o resultado esperado, e utilizar os erros para ajustar os pesos a fim de convergir para a resposta correta. Portanto, os pesos devem ser ajustados de uma maneira cautelosa e matematicamente fundamentada com o intuito de reduzir os erros sobre os padrões de treinamento.

Após ajustar os pesos com base nos exemplos de treinamento, a rede neural se torna refinada naquela tarefa e portanto realiza predições mais precisas. Essa característica que permite

as redes neurais fazerem predições de valores que não foram previamente usados no treinamento é chamada de generalização do modelo (AGGARWAL, 2018).

O Neurônio artificial

O neurônio artificial, ilustrado pela Figura 5, é a principal unidade de processamento de uma rede neural. Um neurônio k recebe m sinais de entrada, e para cada sinal j associa um peso w_{kj}. Matematicamente, tal modelagem para um neurônio de índice k é dada por:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{3.1}$$

onde $x_1, x_2, \ldots, x_m$ são os sinais de entrada e $w_{k1}, w_{k2}, \ldots, w_{km}$ são os respectivos pesos.

Para ajustar o cálculo ao contexto de aplicação, existe uma entrada que é invariante, independente dos sinais de entrada, referida como tendência ou ***bias***. Relacionando o *bias* (b_k) ao neurônio, tem-se o valor de entrada, chamado de **campo local induzido** (v_k) e descrito pela equação:

$$v_k = u_k + b_k \tag{3.2}$$

Para adicionar o *bias como um termo aprendizado pelo neurônio, é acrescentado uma entrada de valor fixo, geralmente 1*, a qual participará de todo o processo de cálculo possuindo também um peso associado (AGGARWAL, 2018). Desta forma, a modelagem do neurônio se torna:

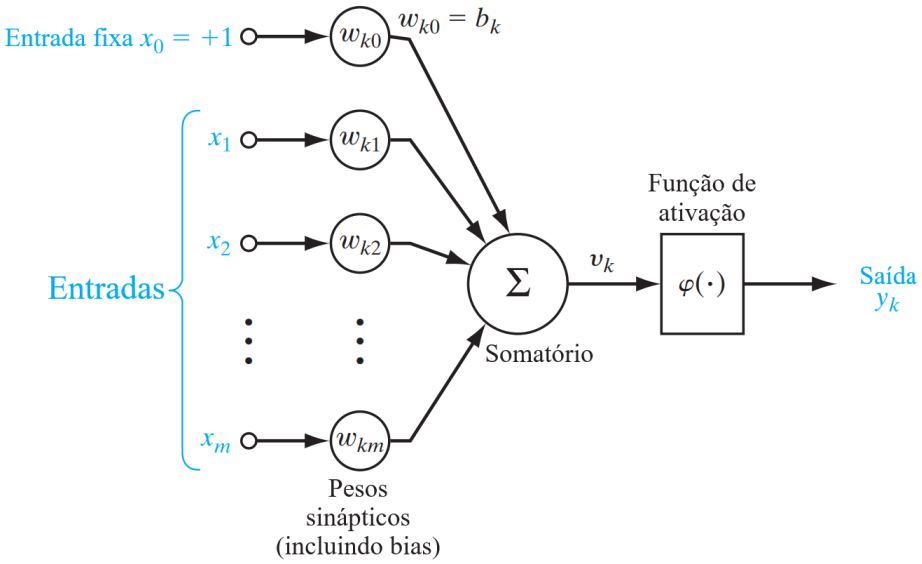
$$v_k = \sum_{j=0}^{m} w_{kj} x_j \tag{3.3}$$

onde $x_0 = 1$ e w_{k0} é o bias, que agora é aprendido pela rede.

Em seguida, o campo local induzido v_k passa por uma **função de ativação** (φ) para limitar a saída em um dado intervalo (tipicamente valores entre 0 e 1 ou -1 e 1), resultando na saída y_k (HAYKIN, 2009).

Existem vários tipos de função de ativação. Entre as funções mais tradicionais estão a função *threshold* e as funções sigmoide. A função *threshold*, dada pela Equação 3.4, também representada no gráfico da Figura 6, recebe um valor e retorna 1 se o valor for maior ou igual à 0, ou 0 caso contrário. Uma função *sigmoide*, tipo mais utilizado, possui um gráfico em "S"(ver Figura 7) e é definida como uma função crescente com equilibro entre comportamento linear e não linear. Uma formulação bastante usada para função de sigmoide é a função logística dada na Equação 3.5. Uma função *sigmoide* possui seu retorno limitado à valores entre 0 e 1, porém

Figura 5 – Neurônio de uma Rede Neural Artificial.

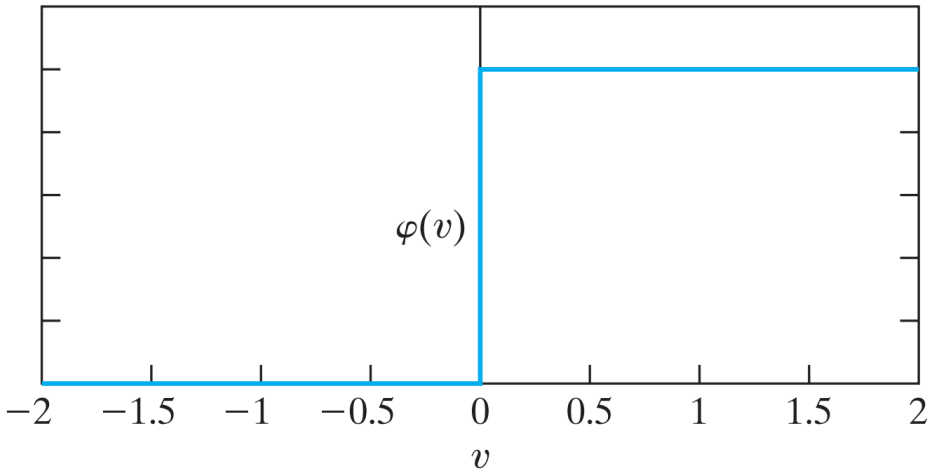


Fonte: Adaptada de Haykin (2009, página 12).

há casos que é desejável uma função de ativação que retorne valores entre -1 e 1 e para isso é utilizado uma função tangente hiperbólica chamada de *tanh*, definida em Equação 3.6.

$$\varphi(v) = \begin{cases} 1 & \text{se } v \geq 0 \\ 0 & \text{se } v < 0 \end{cases} \tag{3.4}$$

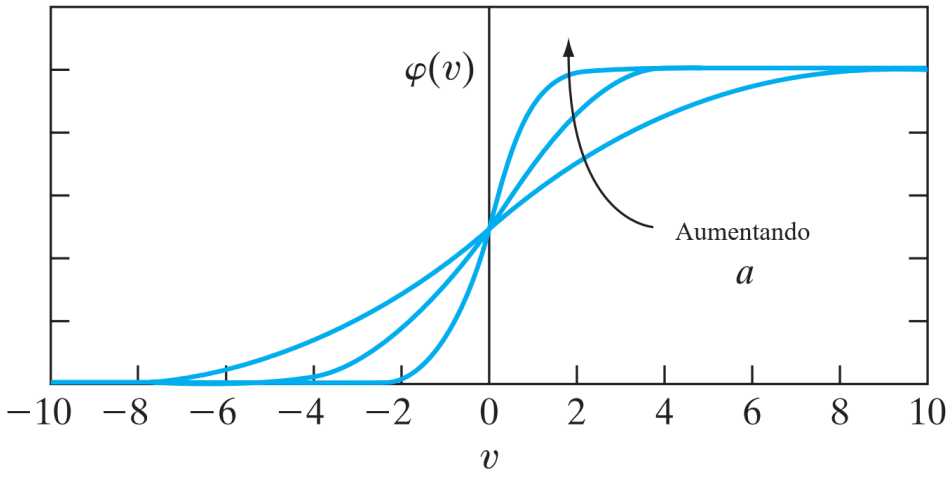
Figura 6 – Função *threshold*.



Fonte: Haykin (2009, página 13).

$$\varphi(v) = \frac{1}{1 + exp(-av)} \tag{3.5}$$

Figura 7 – Função sigmoide.



Fonte: Adaptada de Haykin (2009, página 13).

$$tanh(v) = \frac{e^{v} - e^{-v}}{e^{v} + e^{-v}} \tag{3.6}$$

Redes Neurais Feedforward

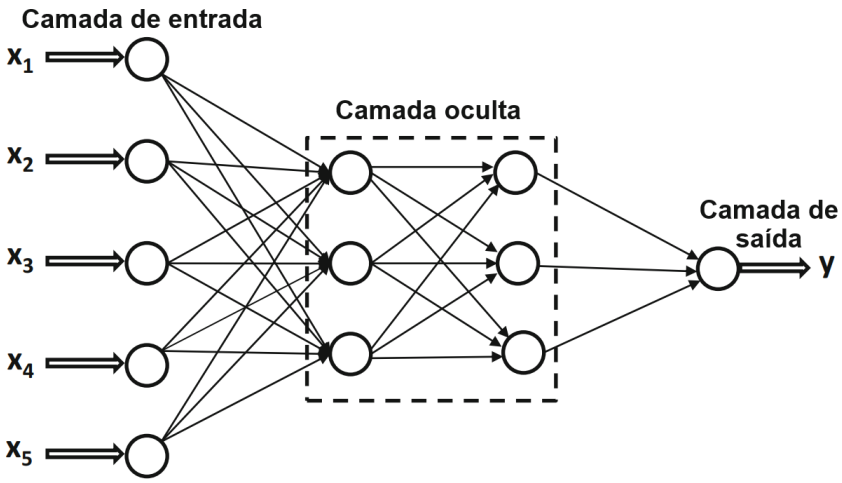
Geralmente uma rede neural é composta de três tipos de camadas (vide Figura 8), a de entrada (única), camada(s) ocultas (pode se ter múltiplas camadas) e a camada de saída (única). O fluxo de informações em uma rede neural é, tradicionalmente, unidirecional com sentido da camada de entrada para a de saída, ou seja, cada camada recebe unicamente informações da camada anterior e os resultados são entregues para a próxima camada, caracterizando assim uma Rede Neural *Feedforward* (AGGARWAL, 2018; HAYKIN, 2009). É importante frisar que em redes recorrentes, parte da informação é contida entre as camadas da rede, como melhor descrito futuramente.

Gradiente Descendente

Utilizado no cálculo vetorial, o gradiente é um vetor que indica direção e sentido, à partir de um ponto específico, para decrementar o valor da função analisada o mais rapidamente possível.

Como apontado por Ruder (2017) gradiente descendente é um método matemático de otimização de uma função objetivo $f(\theta)$ cujos parâmetros seguem o modelo $\theta \in \mathbb{R}^d$. Para tal os parâmetros são atualizados na direção oposta do gradiente, porém em uma proporção α a ser definida para evitar passos curtos ou longos demais em direção ao mínimo local. Portanto, pode-se definir superficialmente o gradiente descendente como uma técnica que procura minimizar o resultado de uma determinada função à partir de um ponto específico.

Figura 8 – Camadas de uma rede *feedforward*.



Fonte: Adaptada de Aggarwal (2018, página 18).

Nas redes neurais a função de custo é utilizada para definir o quão divergente está a resposta da rede em relação à resposta esperada. Quanto maior o resultado da função de custo mais diferente está a resposta do esperado. Nesse contexto é utilizado o gradiente descendente para minimizar a diferença entre a resposta e o esperado.

Retropropagação de erro (Error Back Propagation)

Nesta seção é definido o algoritmo de retropropagação de erro, seguinte estritamente a notação e formulação de (HAYKIN, 2009).

Retropropagação de erro é um algoritmo que usa o gradiente descendente para atualizar os pesos de uma rede neural de várias camadas. O objetivo é calcular eficientemente as derivadas parciais, da função de ativação, obtidas pela rede com respeito à todos os pesos ajustáveis da matriz w para um valor no vetor de entrada x.

O algorítimo consiste em duas partes, a computação para frente e a computação para trás. A primeira consiste em apresentar os dados de entrada para a rede e propagar os sinais da rede até a saída, e a segunda consiste em passar por cada camada, na direção saída para entrada, calculando o gradiente descendente para atualizar os pesos e em seguida, fazer a atualização dos pesos de forma a minimizar a função de custo.

Computação para frente

Uma época é a apresentação de todos os dados de entrada pela rede. Um exemplo de treinamento em uma época é denotado por um vetor de entrada $x(n)$ e um vetor de saída desejada $d(n)$. Para cada camada l e neurônio j é calculado seu resultado $\upsilon_j^{(l)}(n)$ definido por

$$\upsilon_j^{(l)}(n) = \sum_i w_{ji}^{(l)}(n) y_i^{(l-1)}(n) \tag{3.7}$$

onde $y_i^{(l-1)}(n)$ é a saída do neurônio i na camada anterior $l-1$ na iteração n, e $w_{ji}^{(l)}(n)$ é o peso do neurônio j da camada l que é alimentado pelo neurônio i na camada $l-1$, ou seja, o peso que será aplicado à saída do neurônio i da camada anterior $l-1$. Para $i=0$ temos $y_0^{l-1}(n)=+1$, e $w_{j0}^{(l)}(n)=b_j^{(l)}(n)$ caracterizando o bias aplicado ao neurônio j na camada l. Se o neurônio estiver na primeira camada oculta ($l=1$) então $y_j^{(0)}(n)=x_j(n)$ onde $x_j(n)$ é o j-ésimo elemento do vetor de entrada.

Assumindo a utilização de uma função de ativação (φ), como por exemplo a função sigmoide, o sinal de saída de um neurônio j na camada l é

$$y_j^{(l)} = \varphi(\upsilon_j(n)) \tag{3.8}$$

se o neurônio j estiver na camada de saída L então podemos definir

$$y_j^{(L)} = o_j(n) \tag{3.9}$$

onde $o_j(n)$ é uma notação que indica a saída (*output*) do neurônio j da camada de saída para o n-ésimo exemplo de treinamento. Então sinal de erro é dado por

$$e_j(n) = d_j(n) - o_j(n) \tag{3.10}$$

onde $d_j(n)$ é o j-ésimo elemento do vetor de saída desejada.

Computação para trás

Calcula os gradientes locais δ, para os neurônios da rede, definido por

$$\delta_j^{(l)}(n) = \begin{cases} e_j^{(L)}(n)\varphi_j'(\upsilon_j^{(L)}(n)) & \text{para um neurônio } j \text{ na camada de saída } L \\ \varphi_j'(\upsilon_j^{(l)}(n))\sum_k \delta_k^{(l+1)}(n) w_{kj}^{(l+1)}(n) & \text{para um neurônio } j \text{ na camada oculta } l \end{cases} \tag{3.11}$$

onde o apóstrofo em $\varphi_j'(.)$ denota a diferenciação com respeito ao argumento. Para a próxima iteração do algoritmo de retropropagação de erro, os pesos são ajustados por:

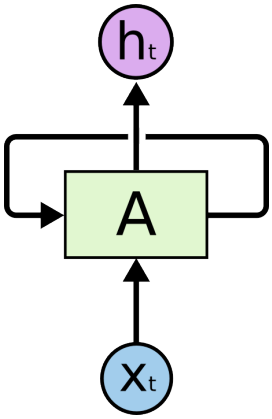
$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \alpha[\Delta w_{ji}^{(l)}(n-1)] + \eta\,\delta_j^{(l)}(n) y_i^{(l-1)}(n) \tag{3.12}$$

onde η é o parâmetro da taxa de aprendizagem e α a constante de momentum.

3.1.1 Redes Neurais Recorrentes

Devido a necessidade de prever dados com dependências temporais, como frases de texto e séries temporais, as redes neurais foram adaptadas com a opção de possuírem ciclos internos de sinal, que produzem a retenção de dados, gerando algo como uma memória. Dessa forma, uma rede neural com no mínimo um ciclo de sinal (*feedback loop*) é chamada de Rede Neural Recorrente (HAYKIN, 2009). Um exemplo básico de tal rede neural é ilustrado na Figura 9.

Figura 9 – Neurônio de uma Rede Neural Recorrente.



Fonte: Olah (2015).

Essa característica, apesar de parecer simples, permite que a RNN possua a capacidade de calcular os resultados a partir de todo o histórico dos dados passados por ela, pois os ciclos de sinal permitem a persistência de dados, algo como uma "memória" (GRAVES, 2012).

RNN são ditas Turing-completo, ou seja, dado suficiente poder computacional e memória física, é possível simular qualquer algorítimo (GRAVES; WAYNE; DANIHELKA, 2014). Porém, na prática as RNNs possuem muitos problemas na generalização de dependências de longo tempo, pois a quantidade de recursos computacionais necessários para treiná-las aumenta de forma inviável. Além disso, há um problema em encontrar o valor de ajuste dos pesos, pois o gradiente pode desaparecer (tender a zero) ou explodir (tender ao infinito) (AGGARWAL, 2018).

Para calcular eficientemente a computação para trás de redes neurais recorrentes são bem conhecidos dois algoritmos, *back propagation through time* (BPTT) e *real-time recurrent learning* (RTRL) (GRAVES, 2012). Segundo Haykin (2009) BPTT requer menos computação que RTRL, mas o o espaço de memória aumenta rapidamente à medida que o comprimento de uma sequência de pares consecutivos de entrada e saída aumenta. Temos então que BPTT é melhor para o treinamento off-line e o RTRL é mais adequado para o treinamento contínuo *on-line*.

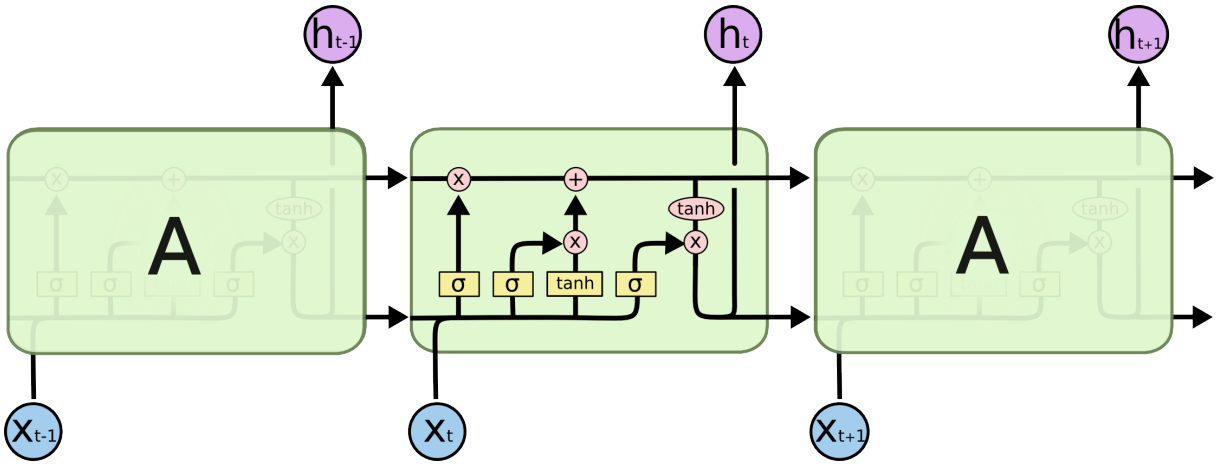
3.1.2 Rede Neural Artificial Long Short-Term Memory (RNA-LSTM)

Com o intuito de resolver os problemas de dependências temporais de longo prazo, Hochreiter e Schmidhuber (1997) desenvolvolveram o conceito *Long Short-Term Memory* (LSTM), que trata diretamente o problema de desaparecimento e explosão do gradiente através de blocos de memória. Cada bloco possui três portões, o portão de entrada, o de esquecimento (adicionado mais tarde por GER (2001)) e o de saída. As funções destes portões são definir: 1) o quanto de informação do estado atual entrará em memória (portão de entrada); 2) o quanto de informação da memória será esquecida (portão de esquecimento); e 3) combinar a memória atual

com as novas informações de memória (portão de saída). Além dos portões, cada bloco possui uma ou mais células de memória, cujo acesso de entrada e saída é controlado pelos portões. Adicionalmente, cada célula possui um ciclo interno de dados para representar a memória, também chamada de estado da célula.

A Figura 10 ilustra o fluxo de dados de uma célula de memória LSTM. O fluxo interno de dados em uma célula de memória começa com o portão de esquecimento. Este, observa a entrada X_t e a saída do cálculo anterior h_{t-1} e aplica uma função sigmoide dando origem a um valor entre 0 e 1, onde 0 significa esquecer completamente e 1 lembrar de tudo, para cada informação da memória. O portão de entrada analisa também a entrada X_t e a saída do cálculo anterior h_{t-1}, aplica uma função sigmoide, e uma função tangente hiperbólica (*tanh*) para normalizar os valores de entrada. O resultado da função sigmoide é multiplicado pelo resultado da função *tanh*. Esse cálculo resulta nas informações que devem ser atualizadas na memória. O resultado então é somado ao estado da célula para adicionar as informações na memória. Por fim, o portão de saída analisa também a entrada x_t e a saída do cálculo anterior h_{t-1} e aplica a função sigmoide multiplicando o resultado pelo estado da célula aplicado uma função *tanh*, resultando então no processamento final da célula de memória.

Figura 10 – Célula de memória LSTM.



Fonte: Olah (2015).

Equações da Rede LSTM

Nesta seção apresenta-se as equações de computação para frente e para trás de uma camada oculta de LSTM em uma Rede Neural Recorrente. A descrição segue estritamente a formulação de Graves (2012).

As equações serão especificadas para um única célula de memória LSTM, visto que para múltiplas células de memória basta repetir a equações para as mesmas. Seja: w_{ij}, o peso da conexão da unidade i para a unidade j; a_j^t, a entrada da rede para a unidade j no tempo t e b_j^t, a ativação da unidade j no tempo t. Os subscritos ι, ϕ e ω se referem ao portão de entrada,

de esquecimento e de saída, respectivamente. O subscrito c refere-se a uma das células C de memória. Os pesos da célula c para as portas de entrada, esquecimento e saída são denotados $w_{c\iota}, w_{c\phi}$ e $w_{c\omega}$, respectivamente; s_c^t é o estado da célula c no tempo t; f é a função de ativação das portas; e g e h são respectivamente as funções de ativação de entrada e saída da célula.

Seja I o número de entradas, K o número de saídas e H o número de células na camada oculta. Note que somente as saídas das células b_c^t estão conectadas aos outros blocos na camada. As outras ativações LSTM, como os estados, as entradas de células ou as ativações de portão, são visíveis apenas dentro do bloco. O índice h para referir às saídas de células de outros blocos na camada oculta. Ao contrário das redes neurais recorrentes padrão, uma camada LSTM contém mais entradas do que saídas, pois os portões e as células recebem entrada do resto da rede, mas apenas as células produzem saída visível para o resto da rede. Por isso, define-se G como o número total de entradas para a camada oculta, incluindo células e portões, e usa-se o índice g para se referir a essas entradas supondo que não se deseja distinguir entre os tipos de entrada. Para uma camada LSTM padrão com uma célula por bloco, G é igual a $4H$.

Tal como acontece com as RNNs padrão, a computação para frente é calculada para uma sequência de entrada X de comprimento T começando em $t = 1$ e aplicando recursivamente as equações de atualização enquanto incrementa t, e a computação para trás (BPTT) é calculada começando em $t = T$, e calculando recursivamente as derivadas unitárias enquanto diminui t para 1. As derivadas de peso finais são encontradas somando as derivadas em cada etapa de tempo.

- **Computação para frente:**

  Portão de saída:

$$\begin{aligned} a_\iota^t &= \sum_{i=1}^{I} w_{i\iota} x_i^t + \sum_{h=1}^{H} w_{h\iota} b_h^{t-1} + \sum_{c=1}^{C} w_{c\iota} s_c^{t-1} \\ b_\iota^t &= f\left(a_\iota^t\right) \end{aligned} \tag{3.13}$$

  Portão de esquecimento:

$$\begin{aligned} a_\phi^t &= \sum_{i=1}^{I} w_{i\phi} x_i^t + \sum_{h=1}^{H} w_{h\phi} b_h^{t-1} + \sum_{c=1}^{C} w_{c\phi} s_c^{t-1} \\ b_\phi^t &= f\left(a_\phi^t\right) \end{aligned} \tag{3.14}$$

  Estado da célula:

$$\begin{aligned} a_c^t &= \sum_{i=1}^{I} w_{ic} x_i^t + \sum_{h=1}^{H} w_{hc} b_h^{t-1} \\ s_c^t &= b_\phi^t s_c^{t-1} + b_\iota^t g\left(a_c^t\right) \end{aligned} \tag{3.15}$$

Portão de saída:

$$\begin{aligned} a_{\omega}^{t} &= \sum_{i=1}^{I} w_{i\omega} x_{i}^{t} + \sum_{h=1}^{H} w_{h\omega} b_{h}^{t-1} + \sum_{c=1}^{C} w_{c\omega} s_{c}^{t} \\ b_{\omega}^{t} &= f\left(a_{\omega}^{t}\right) \end{aligned} \tag{3.16}$$

Saída da célula:

$$b_{c}^{t} = b_{\omega}^{t} h\left(s_{c}^{t}\right) \tag{3.17}$$

- **Computação para trás:**

Saída da célula:

$$\varepsilon_{c}^{t} = \sum_{k=1}^{K} w_{ck} \delta_{k}^{t} + \sum_{g=1}^{G} w_{cg} \delta_{g}^{t+1} \tag{3.18}$$

Portão de saída:

$$\delta_{\omega}^{t} = f'\left(a_{\omega}^{t}\right) \sum_{c=1}^{C} h\left(s_{c}^{t}\right) \varepsilon_{c}^{t} \tag{3.19}$$

Estados:

$$\varepsilon_{s}^{t} = b_{\omega}^{t} h'\left(s_{c}^{t}\right) \varepsilon_{c}^{t} + b_{\phi}^{t+1} \varepsilon_{s}^{t+1} + w_{c\iota} \delta_{\iota}^{t+1} + w_{c\phi} \delta_{\phi}^{t+1} + w_{c\omega} \delta_{\omega}^{t} \tag{3.20}$$

Células:

$$\delta_{c}^{t} = b_{\iota}^{t} g'\left(a_{c}^{t}\right) \varepsilon_{s}^{t} \tag{3.21}$$

Portão do esquecimento:

$$\delta_{\phi}^{t} = f'\left(a_{\phi}^{t}\right) \sum_{c=1}^{C} s_{c}^{t-1} \varepsilon_{s}^{t} \tag{3.22}$$

Portão de entrada:

$$\delta_{\iota}^{t} = f'\left(a_{\iota}^{t}\right) \sum_{c=1}^{C} g\left(a_{c}^{t}\right) \varepsilon_{s}^{t} \tag{3.23}$$

Treinamento em lote, mini-lotes e online

Segundo Haykin (2009) o método de treinamento em lotes consiste no ajuste dos pesos sinápticos após a apresentação de todos os dados de entrada, essa apresentação de todos os resultados de entrada é nomeada época ou *epoch*. Utilizando o algorítimo de gradiente descendente para aplicar o treinamento em lote percebe-se algumas vantagens como estimativa precisa do vetor gradiente e a possibilidade de aplicar paralelismo para realizar os cálculos mais rapidamente. Já no método de aprendizado online, a atualização dos pesos sinápticos é realizada após cada dado de entrada individualmente. Dessa forma o código se torna fácil de implementar e provê maior eficiência para soluções de larga escala e problemas de classificação de padrões

difíceis. Já no treinamento em mini-lotes, ou *mini-batch*, o ajuste dos pesos sinápticos é feito após a apresentação de mini-lotes, de tamanho n, dos dados de entrada (DOKUZ; TUFEKCI, 2021). Esta estratégia reduz a variância nas atualizações dos pesos, e é mais eficiente na computação dos gradientes que o treinamento em lotes. Já no treinamento *online* é ajuste dos pesos é feito após a submissão de cada exemplo do conjunto de treinamento. Este trabalho faz uso do conceito de treinamento em mini-lotes.

> Zhang *et al.* (2019)

CAPÍTULO

4

TRABALHOS CORRELATOS

Neste capítulo é feita uma revisão de pesquisas recentes e proeminentes para a tarefa de Avaliação Automática de Redações (AAR), partindo da menção aos primeiros sistemas. Esta revisão tem como um dos focos a exposição de técnicas de Rede Neural como base de comparação para a arquitetura desenvolvida na presente monografia.

4.1 Técnicas de AAR para a língua inglesa

Para melhor entendimento da motivação inicial e dos desafios na construção de sistemas de AAR, escolheu-se a revisão elaborada por Dikli (2006). No que tange a análise de técnicas, métodos e ferramentas, foi escolhido as recentes revisões do tema (KE; NG, 2019)

Dikli (2006) fornece um panorama das primeiras e mais notáveis iniciativas de sistemas AES para a língua inglesa. Dentre os trabalhos citados estão *Project Essay Grader* (PEG) (PAGE, 1966; PAGE, 1967), *Intelligent Essay Assessor* (IEA) (FOLTZ; LAHAM; LANDAUER, 1999), *E-rater* (SHERMIS; BURSTEIN, 2003), *Criterion* (BURSTEIN; CHODOROW; LEACOCK, 2003), IntelliMetric (ELLIOT, 2003), *MY Access!* (LEARNING, n.d.) e *Bayesian Essay Test Scoring System* (BETSY) (BETSY, n.d.). Mediante a análise desses sistemas, a autora discorre sobre a viabilidade e importância de sistemas AES como um todo. Além disso, são levantados diversos desafios ainda relevantes para os sistemas AES de hoje, como: a vulnerabilidade de sistemas AES que avaliam através de características baseadas em tamanho, no qual o candidato poderia escrever palavras ou redações maiores para receber melhores pontuações; A dificuldade de analisar competências semânticas e o conteúdo da redação em sistemas com características superficiais; e a falta de *corpora* mais robustos para garantir que os sistemas estejam bem treinados.

Ke e Ng (2019) apresentam uma visão geral dos principais marcos alcançados nas pesquisas de AES, desde o seu início, e foca em técnicas atuais. Com base neste estudo, também

apontam o estado-da-arte para a tarefa: para avaliação holística, ambos valores obtidos para as métricas QWK e PCC são bastante altos nos *corpora Cambridge Learner Corpus First Certificate in English* (CLC-FCE), *Automated Student Assessment Prize* (ASAP) e *Test of English as a Foreign Language* (TOEFL11). Os autores argumentam que as técnicas *in-domain* e *cross-domain* alcançaram valores acima de 0.6 (que correspondem aos tipos de avaliação *prompt-specific* e *cross-prompt*, respectivamente, descritos no Capítulo 2). Agora, para avaliação *prompt-specifc trait scoring*, referente aos *corpora International Corpus of Learner English* (ICLE) e *Argument Annotated Essays* (AAE) em termos da métrica PCC, são piores que as avaliações *prompt-specific holistic scoring*, que procuram obter diretamente a nota total da redação. Por outro lado, isso não sugere que esta se sobrepõe àquela. Isso se deve ao fato de que foram resultados obtidos em diferentes *corpora*, assim como o número de redações usadas para treinar avaliação holística tendem a ser maiores do que as utilizadas para avaliação específica a competência.

Para organizá-las de modo comparativo entre si e oferecerem um ponto de partida para a análise dos trabalhos correlatos em língua portuguesa, as Tabelas 8 e 9 foram adaptadas de Ke e Ng (2019) e destacam, respectivamente, os corpora de maior popularidade para a tarefa e a descrição das técnicas, características extraídas e resultados. Algumas observações podem ser feitas sobre as características apontadas na Tabela 8. Existe um grande desafio em comparar técnicas usadas na língua inglesa com os trabalhos de língua portuguesa com foco no ENEM. Os corpora ingleses se utilizam de diferentes tipos textuais, com redações de diferentes tamanhos, diferentes intervalos de pontuação e com estudantes de perfis diferentes. Não obstante, o objetivo dos exames ingleses tem como foco a análise da proficiência do aluno na língua, em contraste com a redação do ENEM, que também analisa a qualidade da proposta de intervenção e repertório de fatos com argumentos de autoridade. Mesmo que os objetivos se alinhem em alguns pontos, questões intrínsecas das línguas podem distanciar os métodos de reconhecimento de desvios e avaliação de domínio da escrita formal. No que tange a análise da Tabela 9, observa-se que as características pontuadas são dificilmente categorizadas entre as competências do ENEM, sem que arrisque as suas independências. Destarte, cabe uma análise aprofundada das relações que existem entre as características apresentadas e as competências do ENEM, para que seja permitido uma comparação de métodos e ferramentas de forma mais assertiva.

Uto (2021) desenvolve uma revisão de sistemas AAR com foco em abordagens com Redes Neurais Profundas (RNP, ou também do inglês, *Deep Neural Network* - DNN). Neste, organiza trabalhos entre quatro tipos e descreve as abordagens mais proeminentes para a tarefa à nível de arquitetura e implementação. Iniciativa útil para a reprodução e superação de trabalhos futuros. Muitos sistemas AAR com abordagem RNP alcançaram o estado-da-arte (UTO, 2021). Dentre estas, destacam-se as que utilizam arquiteturas híbridas entre RNA-LSTM e *feature engineering*.

Tabela 8 – Comparação entre os *corpora* populares para a tarefa de AES

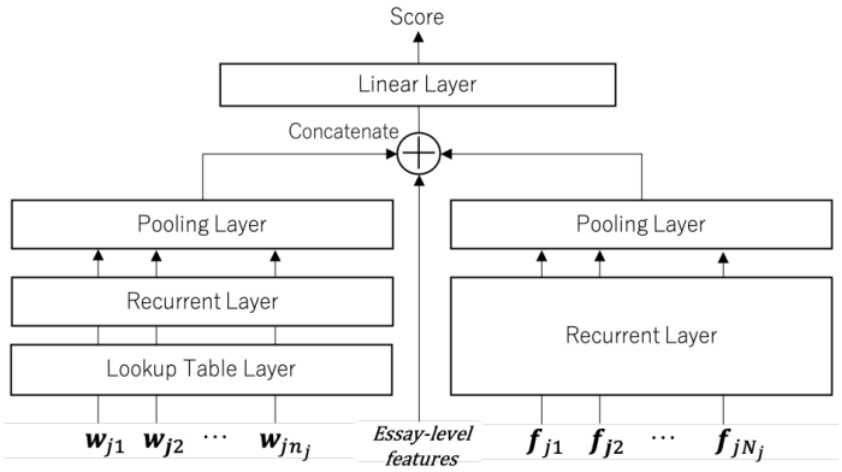
Os tipos de redações presentes no corpus (A = argumentativo, R = resposta, N = narrativo, C = comentário, S = sugestões e L = carta.

Corpus	Tipo da redação	Perfil do escritor	Nº de redações	Nº de temas	Tarefa de avaliação	Intervalo de pontuação
CLC-FCE	A,N,C,S,L	Não-nativos; Participantes do ESOL	1244	10	Holística	1-40
ASAP	A,R,N	Estudantes dos EUA; 7 a 10 anos	17450	8	Holística	pequeno [0-3] grande [0-60]
TOEFL11	A	Não-nativos; Participantes do TOEFL	1100	8	Holística	Baixo, Médio, Alto
ICLE	A	Não-nativos; Estudantes de graduação	1003	12	Organização	1-4 (com incrementos de 0.5)
			830	13	Clareza da tese	
			830	13	Aderência ao tema	
			1000	10	Persuasão	
AAE	A	Comunidade *online*	102	101	Persuasão	1-6

Adaptada de Ke e Ng (2019)

Por conseguinte, Uto, Xie e Ueno (2020) propõe um modelo de RNP híbrida, com a inclusão de características à nível de redação extraídas manualmente. Os autores argumentam, assim como exposto no Capítulo 2, que são abordagens complementares, e que a concatenação dessas características pode ser facilmente aplicada aos modelos RNP tradicionais sem o aumento extensivo de complexidade, mas incrementando significativamente os resultados. Em sua abordagem, apresentada pela Figura 11

Figura 11 – Representação convencional do modelo com características à nível de redação



Fonte: Adaptada de Uto, Xie e Ueno (2020).

Tabela 9 – Performance do estado-da-arte para sistemas de AES na língua inglesa

As características avaliadas nos trabalhos obedecem 10 categorias: L = Baseadas em comprimento; X = Léxicas; E = Word Embeddings; C = Baseadas em categoria; P = Relevância ao tema; R = Legibilidade; S = Sintática; A = Argumentação; M = Semântica; D = Discurso

Corpus	Sistema	Tarefa de avaliação	Abordagem	Características										Métricas		
				L	X	E	C	P	R	S	A	M	D	QWK	PCC	MAE
CLC-FCE	2012_Yannakoudakis	Holística	Ranqueamento	☑			☑			☑		☑	☑	-	0.749	-
ASAP	2018_Cozma (In-domain)	Holística	Regressão			☑								0.785	-	-
	2018_Cozma (Cross-domain)	Holística	Regressão			☑								1->2: 0.661 3->4: 0.779 5->6: 0.788 7->8: 0.649	-	-
TOEFL11	2018_Vajjala	Holística	Regressão	☑	☑			☑		☑			☑	-	0.800	0.400
ICLE	2016_Wachsmuth	Organização	Regressão		☑		☑	☑		☑	☑	☑		-	-	0.315
	2013_Persing	Clareza da tese	Regressão		☑		☑			☑		☑		-	-	0.483
	2014_Persing	Aderência ao Tema	Regressão		☑		☑			☑		☑		-	0.360	0.348
	2016_Wachsmuth	Persuasão	Regressão		☑		☑	☑		☑	☑	☑		-	-	0.378
AAE	2018_Ke	Persuasão	Regressão	☑		☑	☑		☑					-	0.236	1.0335

Adaptada de Ke e Ng (2019)

4.2 Técnicas de AAR para a língua portuguesa

Costa, Oliveira e Júnior (2020) apresentaram um Mapeamento Sistemático de Literatura com o objetivo de obter um panorama dos corretores automáticos para Língua Portuguesa. Tal estudo conclui que os tipos mais recorrentes de abordagens são: Processamento de Linguagem Natural, Aprendizagem de Máquina e outros tipos como Algoritmos Genéticos. No que tange os critérios, a grande maioria se baseia no ENEM, com foco na Competência 1, e os demais em abordagens holísticas. Também é destacado que as bases de dados UOL[1] e Brasil Escola[2] são as mais utilizadas, apesar de alguns trabalhos utilizarem bases privadas. As métricas mais populares são *Pearson's Coefficient Correlation (PCC), Quadratic Weighted Kappa (QWK), Mean Absolute Error (MAE), bem como Precision, Recall e Accuracy* em tarefas de identificação de erros gramaticais eou ortográficos.

Um ponto a ser levantado também diz respeito a construção das técnicas para essa tarefa, uma vez que cada trabalho exploram não uma, mas diversas abordagens e estratégias, nas Tabelas 10 e 11 é destacado que as que obtiveram melhores resultados dentre as abordagens experimentadas com foco em pontuação holística (T) e específica a Competência 1 (C1), mas também é citado trabalhos que envolvam demais competências. Destaca-se no caso de Júnior *et al.* (2016) a utilização de métricas para ambos módulos apresentados (L = Léxico e G = Gramatical), uma vez que fazem parte de uma mesma estratégia.

Bazelato e Amorim (2013) descreveram um classificador bayesiano que avalia redações em português. Dentre as contribuições desse trabalho está a construção de uma base de dados inédita para a língua portuguesa, com notas variando de 0 a 10. Para tanto, extraiu um total

[1] <https://educacao.uol.com.br/bancoderedacoes>
[2] <https://vestibular.brasilescola.uol.com.br/banco-de-redacoes>

de 429 redações da base de dados da UOL com uma média de 13 redações por tema. No que tange a sua técnica, foi possível alcançar 0.396 para PCC e acurácia de 52% para a predição de desvios ortográficos e sintáticos com tolerância de 1.5 pontos das notas atribuídas por corretores humanos.

Júnior *et al.* (2016) apresentaram um Algoritmo Genético para avaliação Léxico-Sintática de redações do ENEM, a qual se relaciona com a competência 1 do ENEM. O método se divide em dois Módulos: *Ortográfico* e *Gramatical*, os quais têm o objetivo de identificar erros sintáticos, léxicos e gramaticais. Com base na quantidade de erros identificados na redação, esta é classificada entre cinco categorias de notas. A técnica usada utiliza *Latent Semantic Analisys* (LSA) para extração de características semânticas. Avaliado sob um conjunto pequeno de somente 20 redações corrigidas por especialistas humanos, o método obteve *recall* de 100% e *precision* de 89,6% para a análise léxica; e *recall* de 81,2% e *precision* também de 81,2% para a análise gramatical. Esses valores valores são mensurados através da quantidade de desvios identificados.

Oliveira (2017) desenvolveu um sistema AES para a Competência 1 de redações do ENEM: "Demonstrar domínio da modalidade escrita formal da língua portuguesa". Neste, o autor extraiu várias características gramaticas das redações e usou o algoritmo evolucionário *Particle Swarm Optimization* (PSO) para selecionar as características que mais influenciam na nota conforme esta competência. O critério de avaliação otimizado pelo algoritmo PSO na seleção das características textuais consistia de uma medida de separabilidade das classes do conjunto de treinamento. Sendo assim, a tarefa é tratada como um problema de classificação. Para a classificação foi usado *Support Vector Machine* (SVM), devido a sua boa capacidade de generalização, robustez em grandes dimensões e por possuir uma teoria bem definida. O autor reporta que os resultados foram avaliados em uma base de dados do UOL de 4902 redações. Foram feitos vários experimentos: com e sem ponderação de características, diferentes revisores gramaticais, assim como variações na tolerância da distância com a nota atribuída pelo corretor humano. O erro médio absoluto, *precision* e *recall* de 0,0224, 0,9773, e 0,9777, foram obtidos, respectivamente, considerando uma tolerância de 0.5, popularmente usada em outros trabalhos.

Amorim e Veloso (2017a) propuseram um sistema de correção automática de redações multi-tema para o português brasileiro, conforme as competências descritas pelo ENEM. Os autores fizeram uma análise de características para cada competência, aplicando parte das categorias de características definidas em trabalhos da língua inglesa (*Domain features e General features*). Com base nas características, os autores usam um cálculo de regressão simples para prever as notas individuais para cada competência. Os autores reportam que certas características tem bom desempenho para algumas competências, enquanto que outras tem desempenho precário. O melhor valor de QWK para a competência 1 é 0.335, enquanto que para a nota geral, alcançou 0.425.

Diferente de seu primeiro trabalho, Júnior *et al.* (2017) utilizaram de uma base de dados

conjunta entre UOL e outra base privada, somando um total de 5407 redações para treino e 60 redações para teste, com um total de 20 temas. Neste, utilizaram um sistema baseado em Rede Neural Profunda, e através da técnica de Raciocínio Baseado em Casos, objetivaram obter resultados para todas as competências do ENEM. Nos resultados, dentre as métricas selecionadas, a PCC é a única possível de ser observada a nível de comparação com os demais trabalhos, a qual obteve o valor 0.111, interpretada como uma correlação muito fraca.

Fonseca *et al.* (2018) apresentaram duas estratégias para Correção Automática de Redações. A primeira utiliza uma Rede Neural Artificial Profunda LSTM bidirecional (BiLSTM). A segunda se utiliza de características pré-projetadas, divididas em quatro categorias: Métricas de contagem; Expressões específicas; *Tokens n-Grams*; *POS n-Grams*; *POS counts*). Os autores reportam que a segunda estratégia teve um desempenho melhor nas quatro primeiras competências, enquanto que a primeira teve desempenho superior na competência 5. Na pontuação geral, o melhor modelo atingiu um valor surpreendente de QWK de 0.752 e RMSE de 100.0, resultados que superam com larga escala os outras abordagens holísticas. Na competência 1, sua pontuação foi de 0.676 para QWK e 25.81 para RMSE. Outro grande destaque é sua base de dados privada de 56644 com pontuações variando de 0 a 1000.

Filho *et al.* (2018) desenvolveram uma abordagem baseada em aprendizado de máquina para classificação automática da aderência da redação ao tema e a estrutura argumentativa de redações. Foram utilizadas 4245 redações dos corpora UOL e Brasil Escola. Utilizaram Máquinas de Vetores de Suporte (SVM) com dois modos de inferência: regressão (R-SVM) e classificação (C-SVM), cujo segundo foi apresentado na Tabela 11 por obter melhor resultado na métrica PCC (0.7470). Outras métricas também foram utilizadas, como *Precision* e *Recall*, as quais foram calculadas entre toda a distribuição de notas, representada como intervalos percentuais nesta mesma Tabela. Apesar do foco atribuído a competência 2 do ENEM, o grande destaque de sua abordagem é a técnica de *Class Balancing*, que os autores argumentaram ser uma etapa essencial para melhorar os resultados, sobretudo, para bancos de redação menores, iniciativa inédita em relação aos trabalhos levantados. Não obstante, também fundamentam a importância da extração de características relevantes ao contexto da competência.

Oliveira *et al.* (2019) destacaram a necessidade do Reconhecimento de Entidades Nomeadas (do inglês, *Named Entities Recognition - NER*) como características adicionais para melhorar a qualidade em duas das cinco competências do ENEM, competências 1 e 3, sendo a última o foco dos autores. Eles utilizaram técnicas de *Conditional Random Fields* e *Local Grammars* (CRF+LG), cuja primeira é um método de aprendizado de máquina para previsão estruturada e a segunda é um meio de representar as regras contextuais da abordagem linguística. Os autores argumentam que os resultados apresentados são superiores aos de Fonseca *et al.* (2018) para a competência 3, com 0.581 contra 0.508 para QWK, e para competência 1, com 0.690 contra 0.678. Além de apresentar um resultado superior para a métrica específica às competências, outro fator de destaque é o uso de um corpus com apenas 2211 redações, extraídas da base de

dados da UOL. Apesar de argumentarem que há grande variação da métrica QWK para notas maiores ou menores que 1.0, merece devida atenção dos sistemas de AES.

Júnior *et al.* (2020) compararam dez arquiteturas diferentes de redes neurais derivadas de rede neural LSTM, com mecanismos de agregação, rede neural hierárquica e rede neural de aprendizado multi-tema para tarefa de AES em língua portuguesa através de uma abordagem holística. Este trabalho dá destaque para a arquitetura de melhor resultado: uma versão modificada da Arquitetura da Rede Neural LSTM com Mecanismo de Agregação, na qual cada tema tem uma camada linear diferente com pesos não compartilhados (AMT-ATT). Esta técnica também utiliza do ELMO para representação das palavras. Os resultados foram analisados através das métricas *Quadratic Weighted Kappa* (QWK), *Root Mean Squared Error* (RMSE), obtendo os valores 0.514 e 101.133, respectivamente. Além de seus resultados competitivos, destaca-se pela base privada de dados com 27.184 redações, onde se tem, no mínimo, 710 redações por tema.

Ramisch (2020) desenvolve uma densa e detalhada descrição de análises linguísticas qualitativas dos desvios sintáticos, sugerindo um subsídio de grande impacto para Processamento de Línguas Naturais de forma geral, e consecutivamente, para Avaliação Automática de Redações. Não obstante, desenvolve um sistema de AES baseado na técnica de *Logistic Regression*, obtendo 75,62% de acurácia para a identificação dos desvios relacionados a competência 1. Diferente dos trabalhos anteriores, ela construiu uma base de dados com 1045 redações escritas por alunos do ensino médio e a segmentou em 10652 sentenças, as quais foram aplicadas as técnicas. Faz-se relevante o estudo das características extraídas por ela, assim como a análise de sua relevância com o treinamento em corpus maiores, ou até incrementá-las nas abordagens de maior destaque.

Diferente dos trabalhos supracitados, o trabalho de Neto *et al.* (2020) se baseia em redações de concursos públicos, com uma base privada de 1000 redações. Além disso, utiliza da técnica de *Random Forest*, algoritmo no qual se constroem diversas árvores de decisão considerando diferentes atributos, com o objetivo de retornar a relevância de cada atributo na etapa de classificação. Os autores argumentam que superaram o trabalho de Amorim e Veloso (2017a), com 0.68 contra 0.425 para a métrica QWK.

Por fim, apesar dos fatores levantados, é possível perceber que abordagens que envolvam Redes Neurais Profundas (RNP, ou também do inglês, *Deep Neural Network - DNN* obtiveram resultados expressivos, próximos ao estado da arte inglês, ainda que não sejam diretamente comparáveis. Se incluem a essa técnica, a utilização de derivações da arquitetura de redes neurais LSTM (Fonseca *et al.* (2018) Júnior *et al.* (2020)) com a aplicação de descrição vetorial de palavras (*Word Embeddings*), relativo às questões 1 e 2 desta monografia. Apesar das redes neurais retirarem a necessidade da inclusão de características, elas ainda são relevantes para corpus pequenos e ainda podem ser observadas com grande utilidade para diminuir o treinamento das redes neurais Ke e Ng (2019), relativo a questão 3 e 4. Por outro lado, abordagens que se apoiam unicamente em características simples, abrem brechas para *trapaças*, crítica levantada por Dikli (2006).

Tabela 10 – Comparação entre vários corpora utilizados.

#	Sistema	Corpus	Modelo	Nº de redações	Nº de temas	Técnica
1	2013_Bazelato	UOL	ENEM	429	~33	Classificação Bayesiana
2	2016_Santos Júnior	UOL	ENEM	20	-	Latent Semantic Analisys
3	2017_Almeida Júnior	UOL	ENEM	4902	-	Particle Swarm Optimization (PSO) + Support Vector Machine (SVM)
4	2017_Amorim	UOL	ENEM	1840	-	Regressão Linear
5	2017_Santos Júnior	UOL + Base privada	ENEM	5407	20	Rede Neural Profunda (DNN) / Raciocínio Baseado em Casos
6	2018_Fonseca	Base privada	ENEM	56644	-	Rede Neural Profunda (BiLSTM)
7	2018_Haendchen Filho	UOL + Brasil Escola	ENEM	4245	-	Support Vector Machine (SVM)
8	2019_Oliveira	UOL	ENEM	2211	-	Conditional Random Fields (CRF) + Local Grammars (LG) + Named Entities Recognition (NER)
9	2020_Bittencourt_Júnior	Base privada	ENEM	27.184	18	Rede Neural Profunda (AMT-ATT)
10	2020_Ramish	Base privada	ENEM	1.045	-	Logistic Regression
11	2020_Sirotheau	Base privada	Concurso	1.000	-	Random Forest

Fonte: Elaborada pelo autor.

Tabela 11 – Ponderação por eixo temático.

Sistema	Nº de redações	Nº de temas	Relacionado à Competência					Resultados						
			C1	C2	C3	C4	C5	QWK	PCC	MAE	RMSE	*Accuracy*	*Precision*	*Recall*
2013_Bazelato	429	~33	☑	☑	☑	☑	☑	-	0.396	-	-	52%	-	-
2016_Santos Júnior	20	-	☑	-	-	-	-	-	-	-	-	-	L = 89,4% G = 81,2%	L = 100% G = 81,2%
2017_Almeida Júnior	4902	-	☑	-	-	-	-	-	-	0,0224	-	-	0,9777	0,9773
2017_Amorim	1840	-	☑	☑	☑	☑	☑	C1 0.335 T 0.425	-	-	-	-	-	-
2017_Santos Júnior	5407	20	☑	☑	☑	☑	☑	-	C1 0.111	-	-	-	-	-
2018_Fonseca	56644	-	☑	☑	☑	☑	☑	C1 0.678 **T 0.752**	-	-	**C1 25.81 T 100.00**	-	-	-
2018_Haendchen Filho	4245	-	-	☑	-	-	-	-	0.7470	-	-	-	39.35% a 81.94%	31.3% a 90.34%
2019_Oliveira	2211	-	☑	-	☑	-	-	**C1 0.690** C3 0.678	-	-	-	-	-	-
2020_Bittencourt_Júnior	27.184	18	☑	☑	☑	☑	☑	T 0.514	-	-	101.133	-	-	-
2020_Ramish	1.045	-	☑	-	-	-	-	-	-	-	-	75,62%	-	-
2020_Sirotheau	1.000	-	☑	☑	☑	☑	☑	0.68	-	-	-	-	-	-

Fonte: Elaborada pelo autor.

CAPÍTULO

5

MATERIAIS, MÉTODOS E RESULTADOS

Neste capítulo são apresentados o corpus de redações, o repositório de *word embeddings* na língua portuguesa e as arquiteturas de redes neurais utilizadas para a predicação de pontuações nas redações. Não obstante, são apresentados os métodos de pré-processamento e as características extraídas a nível de redação. Por fim, são apresentados os resultados obtidos em comparação com os trabalhos correlatos.

5.1 Corpus

Esta pesquisa faz uso do *corpus* de redações disponibilizado por Marinho, Anchiêta e Moura (2021), que segue a especificação do ENEM, denominado Essay-BR: *a Brazilian Corpus of Essays*. O Corpus Essay-BR contém 4.570 redações relacionadas a 86 temas. Conforme os autores, essas redações foram coletados de Dezembro de 2015 a Abril de 2020 do 'Vestibular UOL'[1] e 'UOL Redações'[2]. Os temas de redações incluem direitos humanos, questões políticas, de saúde, culturais, notícias falsas (*fake-news*), movimentos populares, dentre outros. Cada redação é anotada com pontuações para as cinco competências, conforme o exame do ENEM. Na Tabela 12 é apresentado um sumário do *corpus* Essay-BR, caracterizando o tipo do texto, número de redações, de temas, o intervalo de proficiência, dentre outros. Na Tabela 13, onde se observa a distribuição de redações por pontuação, percebe-se um total de 86,83% de redações nos níveis 120 e 160, o que impacta na capacidade de generalização das abordagens subsequentes ao prever redações em diferentes níveis, tópico comparado com outras considerações no Capítulo 6. Na Tabela 14 são apresentadas as especificações estatísticas do *corpus* Essay-BR, onde observa-se que, em média, uma redação tem 4 parágrafos, e cada parágrafo tem 2 frases. Além disso, as frases são um pouco longas, com uma média de 30 *tokens*.

[1] <https://vestibular.brasilescola.uol.com.br/banco-de-redacoes>
[2] <https://educacao.uol.com.br/bancoredacoes>

Tabela 12 – Sumário do Corpus Essay-Br

Informação	**Valor em Essay-BR**
Tipo de texto	Dissertativo Argumentativo
Nível de escrita	Estudantes do segundo grau
Pontuação	Holística
Número de redações	4570
Número de temas	86
Número de competências	5
Intervalo de proficiência	$[0-200]$
Pontuações de proficiência	0, 40, 80, 120, 160, 200

Fonte: Adaptada de Marinho, Anchieta e Moura (2021).

Tabela 13 – Distribuição de redações por pontuação na Competência 1

Níveis	0	40	80	120	160	200
Redações	97	24	359	**2.630**	1.338	122
Porcentagem	2,12%	0,53%	7,86%	**57,55%**	29,28%	4,38%

Fonte: Adaptada de Marinho, Anchieta e Moura (2021).

Tabela 14 – Estatísticas das redações do *corpus* Essay-BR

Estatística	**Média**	**Desvio padrão**
Parágrafos por redação	4,08	1,15
Sentenças por redação	10,57	4,42
Sentenças por parágrafo	2,58	1,44
Token por redação	324,40	94,14
Token por redação	79,33	35,22
Token por sentença	30,66	17,68

Fonte: Adaptada de Marinho, Anchieta e Moura (2021).

Pré-processamento

Para o pré-processamento do corpus de redações elaborado por Marinho, Anchieta e Moura (2021), são utilizadas as mesmas técnicas que o Núcleo Interinstitucional de Linguística Computacional (NILC) da Universidade de São Paulo (USP) - São Carlos, utilizou para o pré-processamento nos *corpora* para o geração dos *word embeddings*. Estas técnicas de pré-processamento são descritas em detalhes em (HARTMANN *et al.*, 2017) e sumarizadas nos tópicos a seguir:

Tokenização: é utilizado o pacote de *tokenization* da biblioteca *Natural Language Toolkit* (NLTK), do Python 3.9, para obter os lexemas que constituem o texto. O processo de *tokenization* executado pelo pacote se baseia no uso de expressões regulares que descrevem lexemas significativos para uma dada linguagem. Neste caso é preciso configurar o tokenizador para a língua portuguesa para se fazer o uso das expressões regulares corretas,

uma vez que a língua portuguesa contém caracteres especiais como ’ç’, e acentuações com til, crase, circunflexo, agudo, dentre outros símbolos especiais. Optou-se pelo uso da biblioteca NLTK para manter o compatibilidade exata com o pré-processamento feito pelo NILC, antes da geração dos word embeddings.

Normalização: O *corpus* Essay-BR foi normalizado para reduzir o tamanho do vocabulário, sob a premissa de que a redução de vocabulário fornece vetores mais representativos. Palavras com menos de cinco ocorrências em todo o *corpus* foram substituídos por um símbolo especial UNKNOWN (do inglês, desconhecido). Os numerais foram normalizados para zeros; URLs foram mapeados para o *token* URL e *e-mails* foram mapeados para um *token* EMAIL.

Remoção de *stop-words*: A remoção de *stop-words* é o processo de remover lexemas ou palavras consideradas irrelevantes para o objetivo de extração de informações semânticas, principalmente por aparecer com alta frequência em todos os textos. Essas palavras pode ser pronomes (eu, meu, minha, seu, sua, etc.) preposições (em, de, para, sobre, atrás, etc.), conjunções (até, quando, dado que, depois, etc.), artigos (o, a, um, uma, etc.), e verbos auxiliares (ser, fazer, ter, ir, poder, etc.).Mesmo que os *word embeddings* fornecidos pelo NILC não faz tal remoção, esta etapa foi inclusa a critério de experimentação e comparação de resultados na "Etapa 1"dos resultados. Para a aplicação da remoção de *stop-words*, também se utiliza o pacote da biblioteca *Natural Language Toolkit* (NLTK) para a língua portuguesa, do Python 3.9.

Word Embeddings

Foram utilizadas *word embeddings* extraídas NILC, disponibilizadas no Repositório de Word Embeddings do NILC [3] (NILC-Embeddings), que tem por objetivo fomentar e tornar acessível recursos vetoriais prontos para serem utilizados nas tarefas de Processamento da Linguagem Natural e Aprendizado de Máquina. Neste, são disponibilizadas *word embeddings* como Word2vec (MIKOLOV *et al.*, 2013c), GloVe (PENNINGTON; SOCHER; MANNING, 2014), entre outros, geradas a partir de dezessete corpus diferentes do português do Brasil, de fontes e gêneros variados, totalizando 1.395.926.282 *tokens*.

Em consonância com os objetivos desta monografia, foram analisados os desempenhos de Word2vec e GloVe. Assim como descrito no Capítulo 2, ainda que seja conhecido o desempenho superior de *word embeddings* dependentes do contexto (e.g. ElMo e BERT), a escolha do GloVe se dá pela criação de um comparativo entre arquitetura híbrida com os demais trabalhos correlatos e analisar a consequência do uso de características extraídas a nível da redação. Já a escolha do Word2vec se dá por este ser o método percursor do GloVe.

[3] <http://nilc.icmc.usp.br/nilc/index.php/repositorio-de-word-embeddings-do-nilc>

5.2 Arquiteturas experimentadas

Nesta monografia foram experimentadas duas arquiteturas de redes neurais que são descritas a seguir e ilsutradas na Figura 12, sendo uma LSTM e uma *multilayer perceptron* (MLP).

Arquitetura LSTM

Camada *Lookup table*: na primeira camada da rede cada palavra é transformada em vetor d-dimensional, conforme a representação vetorial de palavra utilizada. Dada uma sequência de palavras $\mathbf{W}$ representada na notação *one-hot* $(\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_m)$, a saída do procedimento de *look-up table* é calculado conforme a Equação 5.1.

$$LT(\mathbf{W}) = (\mathbf{E}.\mathbf{w}_1, \mathbf{E}.\mathbf{w}_2, \ldots, \mathbf{E}.\mathbf{w}_m) \tag{5.1}$$

onde $\mathbf{E}$ é uma função que extrai a representação vetorial da palavra ativa no momento.

Camadas LSTM: Depois de carregadas as descrições vetoriais de palavras (pela camada de *look-up table*, as camadas recorrentes processam a entrada para gerar uma representação mais voltada para aspectos sintáticos/semânticos das redações. Esta representação deve idealmente codificar toda a informação requerida para se atribuir uma nota para a redação. Contudo, dado que as redações são usualmente longas, consistindo de centenas de palavras, são utilizadas duas camadas LSTM, sendo a primeira com 100 células LSTM e a segunda com 64. Essas camadas LSTM são seguidas de *dropout* de 40%, ou seja, 40% das conexões de saídas de uma camada LSTM são desconsideradas na alimentação da próxima.

Normalização: Segundo BATISTA (2003), uma normalização é a transformação dos valores originais de um conjunto de dados para valores em um determinado intervalo, normalmente entre -1 e 1 ou entre 0 e 1. Essa técnica permite que a diferença de escala entre as múltiplas variáveis que os dados fornecidos a um modelo de predição, não gere algum viés. Nos resultados desta pesquisa foi usada a normalização de cada variável no intervalo entre -1 e 1, usando as seguintes equações sequencialmente:

$$std = \frac{\mathbf{x} - min(\mathbf{x})}{max(\mathbf{x}) - min(\mathbf{x})} \tag{5.2}$$

$$scaled = std * (lim_{sup} - lim_{inf}) + lim_{inf} \tag{5.3}$$

onde $\mathbf{x}$ é um vetor de valores; $min(\mathbf{x})$ e $max(\mathbf{x})$ são respectivamente o menor e o maior valor de $\mathbf{x}$; e lim_{inf} e lim_{sup} os valores limites para a normalização, sendo lim_{inf} o menor valor e lim_{sup} o maior valor. Em resumo, a Equação 5.2 faz uma normalização dos valores de variável para o intervalo $[0,1]$ e a Equação 5.3 escala os valores para o intervalo $[-1,1]$.

Camada linear com função de ativação *Rectified Linear Unit* (ReLU) : A camada linear mapeia o vetor de saídas gerado pela última camada LSTM para um valor escalar conforme uma função linear retificada(AGARAP, 2019). A função de ativação *Rectified Linear Unit* (ReLu) e dada pela Equação 5.4, cuja derivada é dada pela pela Equação 5.5.

$$ReLU(x) = max\{0, x\} \tag{5.4}$$

$$ReLU'(x) = \begin{cases} 1, & \text{se } x \geq 0 \\ 0, & \text{c.c.} \end{cases} \tag{5.5}$$

No experimentos todas as notas de redações do conjunto de treinamento e validação são normalizados para o intervalo $[0, 1]$ e usadas para treinar a rede.

Arquitetura MLP

Camada de extração de características das redações: Em consonância com a abordagem de Uto, Xie e Ueno (2020), nesta monografia foram experimentadas a grande maioria de características descritas em seu trabalho, exceto *Automated readability index* (SENTER; SMITH, 1967) e *Flesh-Kincaid grade* (KINCAID *et al.*, 1975) assim como mostra a Tabela 15.

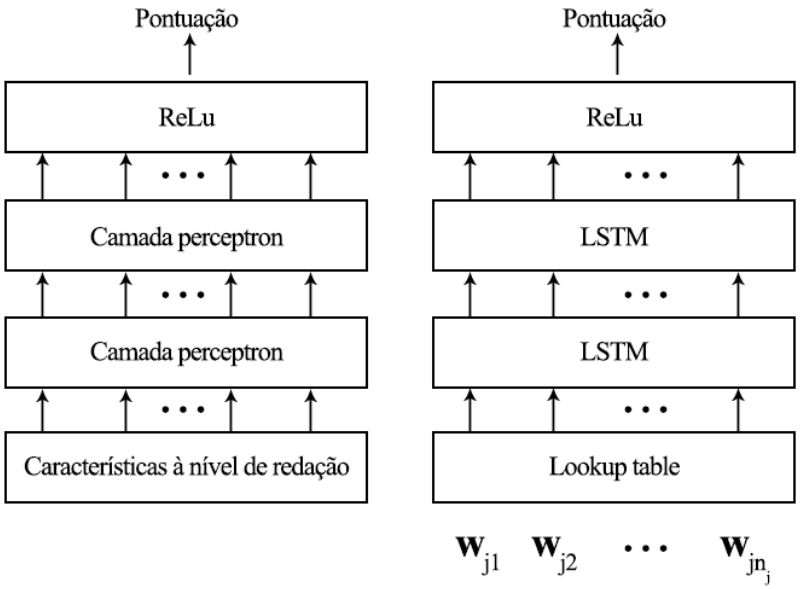
Tabela 15 – Características a nível de redação utilizadas neste estudo

Tipo	**Característica**
Baseadas em tamanho *(Lenght-based)*	Número de palavras, sentenças, lemas, símbolos de pontuação (vírgulas, pontos de exclamação e interrogação), tamanho médio de palavras e sentenças
Sintáticas	Número de substantivos, verbos, advérbios, adjetivos e conjunções
Baseados em palavras *(Word-based)*	Número de palavras escritas incorretamente e *stop-words*
Legibilidade	*Dale-Chall Readability Score* e *Flesch reading ease* (KINCAID *et al.*, 1975), *SMOG index*, Fitzsimmons *et al.* (2010), *Gunning Fog* (WHISNER, 2004), contagem de sílabas e contagem de palavras difíceis.

Fonte: Elaborada pelo autor.

Camadas perceptron: Depois de carregadas as características extraída das redações, as camadas perceptron processam a entrada para gerar uma representação mais voltada para

Figura 12 – Arquiteturas *Multilayer Perceptron (MLP) e LSTM*



Fonte: Elaborada pelo autor.

aspectos sintáticos/semânticos das redações. São utilizadas duas camadas perceptron, sendo a primeira com 100 neurônios e a segunda com 64 neurônios.

Normalização: Se utiliza a mesma normalização usada na rede LSTM, onde os valores de características são normalizados para o intervalo real [-1,1].

Camada linear com função de ativação *Rectified Linear Unit* (ReLU) : Aqui novamente se utilizada a mesma especificação usada na arquitetura LSTM.

Código-base

Para a construção da arquitetura proposta, foi utilizado como base o código fornecido pelo perfil Ivy Zhou na plataforma *GitHub* [4], o qual referencia aos trabalhos de Taghipour e Ng (2016) e Alikaniotis, Yannakoudakis e Rei (2016). Marinho, Anchiêta e Moura (2021) disponibilizaram na mesma plataforma o *corpus* Essay-BR juntamente com funções para a leitura deste; estas também foram utilizadas nesta monografia. Para a utilização de *word embeddings* e o pré-processamento do corpus Essay-BR, foi utilizado o código-base disponibilizado por Nathan Hartmann[5]. Para a utilização dos otimizadores e heurísticas apresentados, foi utilizada a biblioteca Keras do Tensorflow[6]. Como ferramentas de PLN, foi utilizado a biblioteca NLTK[7],

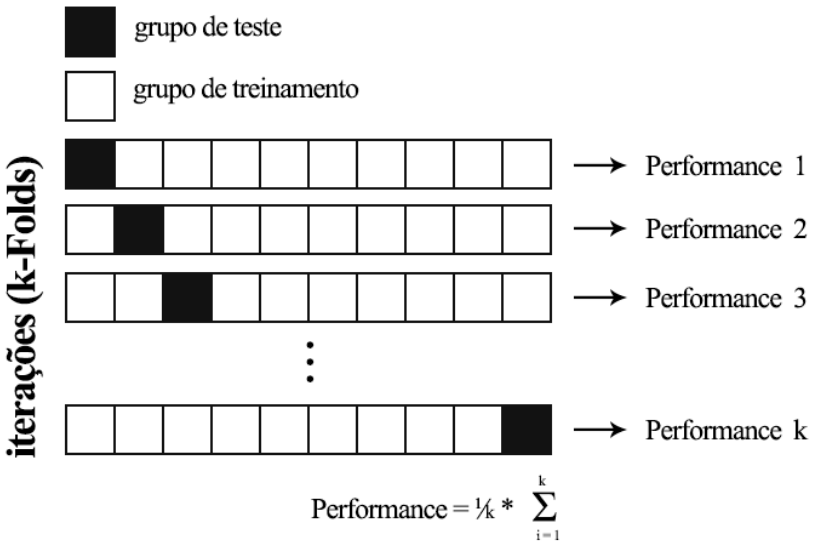
[4] <https://github.com/ivy-zhou/Automated-Essay-Scoring>
[5] <https://github.com/nathanshartmann/portuguese_word_embeddings>
[6] <https://www.tensorflow.org/addons/api_docs/python/tfa/optimizers>
[7] <https://www.nltk.org/>

assim como a biblioteca *pyphen*[8] e *spellchecker*[9] para a separação silábica e contagem de erros de escrita, respectivamente; estas duas últimas são utilizadas nas fórmulas de extração de características de legibilidade.

5.3 Treinamento

Para uma melhor verificação de consistência dos resultados foi utilizada a **validação cruzada *k-fold*** (HAN; PEI; KAMBER, 2011), uma técnica de validação que divide o conjunto total de dados em k partes de tamanhos iguais, ao qual em k iterações, $k-1$ partes são utilizadas para treino e apenas 1 para teste. A cada iteração são alternadas as partes de treino e teste. Este procedimento é ilustrado na Figura 13. Nos experimentos desta monografia foi utilizado o valor $k = 10$. Para cada conjunto de treinamento gerado, foi utilizado 30% dos dados para validação. Adicionalmente foi feito uso da estratégia de treinamento em mini-lotes para todos os otimizadores dos experimentos, usando o valor 64 como tamanho de mini-lote.

Figura 13 – Ilustração de validação cruzada *k-fold*



Fonte: Adaptada de Han, Pei e Kamber (2011)

Tipo de avaliação

Ainda que se faça necessária uma análise mais aprofundada, não foi observado técnicas de segmentação por tema no *corpus* Essay-BR, e portanto, infere-se que as AAR foram treinadas sem distinção de tema. Paralelamente, apesar deste ser denominado pelos autores Marinho, Anchieta e Moura (2021) como um *corpus* com pontuação holística, também é explicitado que ele fornece pontuações específicas às competências, e como este é o foco desta monografia, é crível interpretá-la como *Cross-prompt trait scoring*.

[8] <https://pyphen.org/>
[9] <https://pyspellchecker.readthedocs.io/en/latest/>

Métricas

Dado que a saída de um sistema de AAR é usualmente um número real, a tarefa é frequentemente tratada em abordagens da língua inglesa como uma tarefa de aprendizado de máquina do tipo regressão, embora o problema também seja abordado como um problema de preferência de ranqueamento (CHEN *et al.*, 2010; CHEN; HE, 2013). O modelo do ENEM com pontuação incremental de 40 pontos entre os níveis, abre a possibilidade para que sejam utilizadas de técnicas de classificação. A saída de sistema AAR pode ser comparado com as notas atribuídas por humanos usando várias medidas de correlação e concordância. Estas medidas incluem *Pearson Correlation Coefficient* (PCC) e *Quadratic Weighted Kappa* (QWK) (YANNAKOUDAKIS; CUMMINS, 2015) (ao qual se indica futura utilização no Capítulo 6, com destaque para *Root Mean Squared Error* e *Mean Absolute Error* (MAE), amplamente utilizados para tarefas de regressão, aplicadas na arquitetura proposta.

***Root Mean Squared Error* (RMSE)** Esta é uma medida tradicional de erro usada em problemas de regressão, onde se tem como saída do preditor um valor real. A RMSE, conforme diz o próprio nome consiste do cálculo de raiz quadrada da média das diferenças ao quadrado, entre as predições e o resultado esperado. Matematicamente, esta medida é descrita pela Equação 5.6.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i^*)^2} \tag{5.6}$$

O valor retornado pelo cálculo de RMSE é sempre não negativo e um valor 0 (quase nunca alcançado na prática) indicaria um uma predição perfeita conforme as saídas esperadas. Em geral, quanto mais baixo o valor de RMSE, melhor é o resultado. No entanto, as comparações entre diferentes tipos de dados seriam inválidas porque a medida depende da escala dos números usados. Também, no RMSE, erros maiores têm um efeito desproporcionalmente grande, devido ao fato da diferença ser elevada ao quadrado. Consequentemente, o RMSE é sensível a *outliers*; por exemplo, se uma predição for completamente errada, ela tem um grande peso na média final. Discussões sobre estão questões relacionadas ao RMSE são encontradas em Pontius, Thontteh e Chen (2008).

***Mean Absolute Error* (MAE)** Esta também é uma medida altamente tradicional da literatura voltada a análise de problemas de regressão. Ela basicamente consiste do erro médio em diferença absoluta entre a predição e o resultado esperado, sendo dada matematicamente pela Equação 5.7.

$$\text{MAE} = \frac{\sum_{i=1}^{n}|y_i - y_i^*|}{n} \tag{5.7}$$

Também, conforme discutido em (PONTIUS; THONTTEH; CHEN, 2008), o MAE possui vantagens na interpretabilidade sobre o RMSE, pois a diferença média absoluta pode ser

mais facilmente compreendida do que a raiz quadrada da média dos erros quadrados. Além disso, cada erro influencia o MAE em proporção direta ao valor absoluto do erro, o que não é o caso do RMSE.

Otimizadores

Baseados na técnica de gradiente descendente, oito algoritmos de otimização dos pesos da rede neural foram experimentados, sendo estes: *Stochastic Gradiente Descent* (SGD) (SUTSKEVER *et al.*, 2013), *Root Mean Squared Propagation* (RMSProp) (HINTON; SRIVASTAVA; SWERSKY, 2012), *Adaptive Gradient* (Adagrad) (DUCHI; HAZAN; SINGER, 2011), *Adaptive Delta* (Adadelta) (ZEILER, 2012), *Adaptive Moment* (Adam), *Adaptive Moment based on the infinity norm* (Adamax) (DOZAT, 2016) e *Nesterov-accelerated Adaptive Moment* (Nadam) (DOZAT, 2016) e *Follow The Regularized Leader* (Fltr) (MCMAHAN *et al.*, 2013). Para todos otimizadores, a taxa de aprendizado foi iniciada em 0.01. Todos esses otimizadores são fornecidos pelo pacote *Keras* do *TensorFlow*.

Heurísticas

São experimentadas quatro heurísticas distintas de treinamento da rede (envolvendo critério de parada do treinamento, e taxa de aprendizado adaptativa), são elas: 1) parada do treinamento somente quanto atingir 100 épocas de treinamento (100Epoch); 2) até 100 épocas de treinamento desde que não se tenha a taxa de perda estagnada por 15 épocas (*Early-stop*); 3) 100 épocas de treinamento com redução de taxa de aprendizado pelo fator 0.2 a cada 5 épocas de estagnação (*Reduce-LR*); 4) *Reduce-LR*, porém se houver estagnação por 15 épocas o treinamento é encerrado (*Reduce-LR+Early-stop*).

5.4 Resultados

Etapa 1

Nesta etapa, experimentou-se uma arquitetura RNA-LSTM com o *word embeddings* GloVe de 100 dimensões aplicado a dois conjuntos, um com a remoção de *stop-words* e outro sem. Foram experimentados os algoritmos de otimização: SDG, RMSProp, Adadelta, Adagrad, Adam, Adamax, Nadam, e Frtl, sendo cada um destes combinado com as heurísticas 100Epoch, Early-stop, Reduce-LR, Reduce-LR+Early-stop, com o objetivo de minimizar a função de perda dada pelo RMSE. Os resultados para as métricas RMSE e MAE (Equações 5.6 e 5.7, respectivamente) são descritos na Tabela 16.

A Tabela 16 apresenta todos os resultados da primeira etapa. Quanto menor o valor para as medidas RMSE e MAE, melhor é o resultado. Primeiramente, é observado que a remoção de *stopwords* não oferece melhora significativa. Assim, foi escolhido a retirada deste

pré-processamento para a próxima etapa. Posteriormente, é observado que a heurística 100Epoch se destaca das demais obtendo o melhor resultado em 3 otimizadores diferentes: RMSProp, Adam e Nadam com a remoção de *stopwords*. Para os resultados obtidos sem a remoção de *stopwords*, tem-se um destaque para a métrica RMSE com a utilização das heurísticas 100Epoch e Early-stop para os mesmos otimizadores RMSProp, Adam e Nadam. A critério de desempate entre os otimizadores, foram feitas médias simples com os valores de RMSE para cada heurística nos três otimizadores, considerando-se os resultados sem a remoção de *stopwords*. Desta forma, foi escolhido o otimizador RMSProp. É importante ressaltar que a diferença entre os otimizadores apresentados não é decisiva e, portanto, encoraja-se, no Capítulo 6, a execução de novos experimentos que também os envolvam, para um entendimento geral mais detalhado.

Tabela 16 – Resultados para a etapa 1

		Com remoção de stopwords		Sem remoção de stopwords	
Otimizador	**Heuristica**	**MAE**	**RMSE**	**MAE**	**RMSE**
SGD	100Epoch	0,111	0,161	0,108	0,161
SGD	Early-stop	0,111	0,161	0,107	0,164
SGD	Reduce-LR	0,111	0,161	**0,105**	0,164
SGD	Reduce-LR+Early-stop	0,111	0,161	0,106	0,164
RMSprop	100Epoch	**0,102**	**0,148**	0,110	**0,155**
RMSprop	Early-stop	0,103	0,152	0,109	**0,155**
RMSprop	Reduce-LR	0,104	0,152	0,112	0,158
RMSprop	Reduce-LR+Early-stop	0,105	0,152	0,116	0,161
Adadelta	100Epoch	0,394	0,416	0,394	0,417
Adadelta	Early-stop	0,429	0,456	0,399	0,421
Adadelta	Reduce-LR	0,479	0,511	0,434	0,462
Adadelta	Reduce-LR+Early-stop	0,455	0,484	0,458	0,489
Adagrad	100Epoch	0,108	0,155	0,216	0,245
Adagrad	Early-stop	0,109	0,155	0,216	0,245
Adagrad	Reduce-LR	0,111	0,161	0,257	0,286
Adagrad	Reduce-LR+Early-stop	0,111	0,161	0,261	0,290
Adam	100Epoch	**0,102**	**0,148**	0,109	**0,155**
Adam	Early-stop	0,105	0,152	0,110	**0,155**
Adam	Reduce-LR	0,104	0,152	0,112	0,158
Adam	Reduce-LR+Early-stop	0,105	0,152	0,120	0,164
Adamax	100Epoch	0,105	0,152	0,109	**0,155**
Adamax	Early-stop	0,105	0,152	0,114	0,158
Adamax	Reduce-LR	0,105	0,152	0,132	0,173
Adamax	Reduce-LR+Early-stop	0,105	0,152	0,147	0,187
Nadam	100Epoch	**0,102**	**0,148**	0,109	**0,155**
Nadam	Early-stop	0,104	0,152	0,109	0,155
Nadam	Reduce-LR	0,104	0,152	0,114	0,158
Nadam	Reduce-LR+Early-stop	0,105	0,152	0,122	0,167
Ftrl	100Epoch	0,516	0,536	0,516	0,535
Ftrl	Early-stop	0,516	0,536	0,516	0,536
Ftrl	Reduce-LR	0,516	0,536	0,516	0,536
Ftrl	Reduce-LR+Early-stop	0,516	0,536	0,516	0,536

Fonte: Elaborada pelo autor

Etapa 2

Diante do resultado da Etapa 1 para as métricas 100Epoch e Early-stop, optou-se em combiná-las através do aumento para 200 épocas com um encerramento caso não haja diminuição do erro por 15 épocas. Deste modo, é possível evitar o *overfitting* que, segundo Haykin (2009), é um problema que ocorre quando a rede decora os exemplos de treinamento, de forma a apresentar uma baixa taxa de erro neste conjunto, porém resulta em uma alta taxa de erro quando submetida ao conjunto de teste. Com base nesta decisão, foram experimentados os *word embeddings* Word2vec nos modelos CBOW, Word2vec Skip-gram e GloVe com 50, 100, 300 e 600 dimensões, com os resultados descritos na Tabela 17.

A primeira informação a ser observada é que a abordagem com GloVe de 100 dimensões, otimizador RMSprop e heurísticas 200Epoch + Early-stop obteve melhor resultado que a abordagem com mesmo *word embeddings*, dimensões e otimizador na Tabela 16 para a métrica MAE, mesmo que se manteve igual na métrica RMSE.

De modo geral, também se observa que a arquitetura com a utilização do Word2vec CBOW de 600 dimensões e heurísticas 200Epoch + Early-stop obteve o melhor resultado dentre as abordagens da Tabela 17. Dentre as abordagens com GloVe, o maior resultado se observa em 600 dimensões da mesma, o que em consonância com os demais resultados, indicam que maior dimensionalidade pode resultar em maiores valores para RMSE. Faz-se relevante, portanto, incluir experimentos com dimensões ainda maiores.

Tabela 17 – Resultados para a etapa 2

			Sem remoção de *stopwords*	
Word embeddings	**Otimizador**	**Heuristica**	**MAE**	**RMSE**
GloVe 50 dimensões	RMSprop	200Epoch + Early-stop	0,109	0,158
Word2vec Skip-gram 50 dimensões	RMSprop	200Epoch + Early-stop	0,105	0,158
Word2vec CBOW 50 dimensões	RMSprop	200Epoch + Early-stop	0,105	0,155
GloVe 100 dimensões	RMSprop	200Epoch + Early-stop	0,107	0,155
Word2vec Skip-gram 100 dimensões	RMSprop	200Epoch + Early-stop	0,106	0,155
Word2vec CBOW 100 dimensões	RMSprop	200Epoch + Early-stop	0,106	0,155
GloVe 300 dimensões	RMSprop	200Epoch + Early-stop	0,104	0,152
Word2vec Skip-gram 300 dimensões	RMSprop	200Epoch + Early-stop	0,103	0,152
Word2vec CBOW 300 dimensões	RMSprop	200Epoch + Early-stop	0,103	0,148
GloVe 600 dimensões	**RMSprop**	200Epoch + Early-stop	**0,102**	**0,148**
Word2vec Skip-gram 600 dimensões	RMSprop	200Epoch + Early-stop	0,104	0,148
Word2vec CBOW 600 dimensões	**RMSprop**	200Epoch + Early-stop	**0,101**	**0,145**

Fonte: Elaborada pelo autor

Etapa 3

Destarte, os resultados obtidos nessa monografia podem ser comparados com as abordagens construídas por Marinho, Anchieta e Moura (2021), uma vez que tenha reproduzido as

técnicas de Amorim e Veloso (2017a) e Fonseca *et al.* (2018) para o *corpus* Essay-BR, ambas abordagens proeminentes na tarefa de AAR na língua portuguesa. Com esta iniciativa é possível prover uma base comparativa para este e próximos trabalhos na tarefa. Para tanto, as pontuações de ambas abordagens para a métrica RMSE foram normalizadas para um intervalo de 0 a 1, diferente da que foram propostas (0 a 200).

Observa-se, enfim, que tanto as arquiteturas LSTMs destacadas na etapa anterior quanto as demais, obtiveram resultado superior que as abordagens trazidas em trabalhos anteriores, considerando-se a métrica RMSE, como se observa na Tabela 18.

Tabela 18 – *Root Mean Squared Error* do conjunto de teste para o *corpus* Essay-BR

Modelo	**C1**
(AMORIM; VELOSO, 2017b)	0,174
(FONSECA *et al.*, 2018)	0,170
Arquitetura LSTM	0,145
Arquitetura MLP	0,387

Fonte: Elaborada pelo autor.

O desempenho superior da arquitetura LSTM desenvolvida nesta monografia pode ser explicado por alguns aspectos:

1. Marinho, Anchieta e Moura (2021) apontam, assim como pode ser observado, que o resultado obtido por Fonseca *et al.* (2018) neste *corpus* não é equivalente ao obtido em sua base privada (12,90 em valores normalizados). Isso se deve pelo fato de que esta última tem um *corpus* muito maior, contando com mais de 50.000 redações, que não foi disponibilizado publicamente. Outro fator relevante, é que os autores também descreve que detalhes da implementação (como recursos léxicos) foram aplicados, mas não estão publicamente disponíveis. A publicação de (AMORIM; VELOSO, 2017a) não oferece resultados para a métrica de RMSE.

2. Nesta monografia foi utilizado um treinamento diferente do que foi proposto em ambas abordagens. Assim como supracitado, utilizamos k-fold com $k = 10$, enquanto que os autores dividiram o *corpus* nas proporções de 70% 15% e 15%, cujo corresponde à 3.198, 686 e 686 redações para treinamento, desenvolvimento e teste, respectivamente. Na busca por uma distribuição justa entre as segmentações, esta divisão seguiu as distribuições de notas entre as pontuações de 0 a 1000.

3. Inclui a esta análise, a utilização de *word embeddings* GloVe utilizado por (FONSECA *et al.*, 2018) foi treinado em um *corpus* de 560 milhões de *tokens*, em contraste com o que foi utilizado neste, com aproximadamente 1,395 bilhão de *tokens*. Não foi identificado se a reprodução de Marinho, Anchiêta e Moura (2021) também o utiliza.

4. Ainda que os fatores citados impeçam uma conclusão indubitável, é plausível interpretar que a arquitetura LSTM com a utilização de *word embeddings* proposta tenha apresentado grande papel no resultado final, o que indica um avanço promissor para a tarefa, o qual se sugere caminhos para trabalhos futuros.

Nesta etapa, também se aplicou a arquitetura *Multilayer Perceptron* (MLP). Conforme esperado, esta apresentou um resultado muito alto para a tarefa, duas vezes o valor de erro obtido pela arquitetura LSTM. Isso se deve ao fato de que foi construída com apenas características à nível de redação, bem como se sugere que estas também sejam extraídas por uma arquitetura LSTM (UTO; XIE; UENO, 2020).

CAPÍTULO

6

CONCLUSÃO

Com o objetivo de responder as perguntas levantadas no Capítulo 1 e com base nos trabalhos correlatos a essa monografia para as línguas inglesa e portuguesa, é importante salientar observações a cerca da eficácia das técnicas e o caráter competitivo das mesmas:

1. É notável a escassez de bases públicas com uma expressiva quantidade de dados para experimentação e testes, sendo esta uma limitação que deve ser observada para futuras pesquisas na área. Os melhores resultados para a língua portuguesa utilizam-se de bases privadas, o que dificulta a reprodução e comparação dessas pesquisas e utilização de novas estratégias;

2. A utilização do mesmo *corpus* não é unânime; Somado a isso, as abordagens que utilizam, o selecionaram em diferentes quantidades de redações; A quantidade de redações por tema também pode ser um fator de influência para as métricas avaliadas;

3. Uma vez que nenhum corpus é diretamente do banco de redações do ENEM, os que são acessíveis podem não representar a realidade da correção do exame. Deste modo, espera se que um nível satisfatório de generalidade seja obtido para o corpus mais popular.

4. Os corpora acessíveis não incluem os textos de apoio do tema em conjunto com a redação, o que pode interferir diretamente na capacidade de percepção de contexto de sistemas AAR.

5. Das abordagens selecionadas, não é conclusivo a escolha de características extraídas para uma competência específica; É necessário que se aprofunde na escolha de características relevantes às competências, para obter uma percepção suficientemente informativa para os agentes racionais em questão. Indica-se portanto o estudo dos documentos e manuais construídos para o esclarecimento do processo de correção do ENEM (INEP, 2020; BRASIL, 2020; BRASIL, 2021).

6. Muitos trabalhos utilizam-se de modelos ideais de construção linguística externos às competências do ENEM, por mais que tenham a língua portuguesa como denominador comum. Isso significa que há a possibilidade das competências perderem seu caráter de avaliação individual ao qual foram desenvolvidas, ou seja, incluir características comuns à mais de uma competência, dificultando abordagens específicas à competência.

7. Ainda há espaço para a discussão de métricas viáveis para a análise dos resultados de predição. O desafio *The Hewlett Foundation: Automated Student Assessment Prize (ASAP)* inclinou a decisão pela métrica QWK em ambas línguas, apesar de que Fonseca *et al.* (2018) advocam a favor da RMSE. Em métricas onde é possível a flexibilização da tolerância entre a distância da nota esperada com a nota obtida, é necessário que se estabeleça um valor fixo que se relacione com a tolerância permitida entre avaliadores no processo de correção de redações no ENEM.

Deste modo, é difícil determinar indubitavelmente o estado-da-arte de técnicas para Avaliação Automática de Redações (AAR) no modelo ENEM para a língua portuguesa de modo geral, com evidência para a competência 1: "Domínio da escrita formal da língua portuguesa", foco desta monografia.

Em contrapartida, mediante a comparação de técnicas desenvolvidas para o mesmo *corpus* Essay-BR, também se observou que todas as abordagens de arquiteturas LSTM com *word embeddings* GloVe, Word2vec Skip-gram e Word2vec CBOW (com dimensões de 50, 100, 300 e 600) propostas superaram as abordagens anteriores para a tarefa de Avaliação Automática de Redações na língua portuguesa, o que indica um avanço promissor para a área, ainda sobre ressalvas. Como esperado, a arquitetura MLP sozinha não apresentou bons resultados diante das demais, uma vez que representa apenas características à nível de redação. Também se observa que o otimizador RMSProp apresentou resultados de destaque em comparação com os demais experimentados, ainda que Adam e Nadam também sejam sugeridos para futuras implementações. No que tange heurísticas escolhidas, obteve-se melhor resultado com 100Epoch, Early-stop e 200Epoch + Early-stop.

Em trabalhos futuros, busca-se a utilização da métrica *Quadratic Weighted Kappa* para que a medição de desempenho seja amplamente comparada com demais trabalhos proeminentes na área. É também esperado que se utilize dimensões maiores na experimentação de novas arquiteturas assim como reitera-se a experimentação de *word embeddings* dinâmicas, amplamente conhecidas por resultados superiores às estáticas em tarefas de PLN. No que tange a inclusão de características extraídas manualmente, os resultados e a análise de trabalhos predecessores sugere a concatenação das arquiteturas MLP com LSTM para um melhor resultado. Inclui-se, por fim, a experimentação de outras arquiteturas para a extração de características manuais, bem como uma análise aprofundada da correlação entre as características relativas à Competência 1, para que se adéque uma a outra.

REFERÊNCIAS

AGARAP, A. F. **Deep Learning using Rectified Linear Units (ReLU)**. 2019. Citado na página 59.

AGGARWAL, C. C. **An Introduction to Neural Networks**. Cham: Springer International Publishing, 2018. 1–52 p. ISBN 978-3-319-94463-0. Disponível em: <https://doi.org/10.1007/978-3-319-94463-0_1>. Citado nas páginas 37, 39, 40 e 42.

ALIKANIOTIS, D.; YANNAKOUDAKIS, H.; REI, M. Automatic text scoring using neural networks. **arXiv preprint arXiv:1606.04289**, 2016. Citado na página 60.

AMORIM, E.; VELOSO, A. A multi-aspect analysis of automatic essay scoring for brazilian portuguese. In: **Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics**. [S.l.: s.n.], 2017. p. 94–102. Citado nas páginas 51, 53 e 66.

AMORIM, E. C. F. de; VELOSO, A. A multi-aspect analysis of automatic essay scoring for brazilian portuguese. In: **Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics**. Valencia, Spain: Association for Computacional Linguistics, 2017. p. 3–7. Citado na página 66.

ASSEMBLY, U. G. *et al.* Universal declaration of human rights. **UN General Assembly**, New York, NY, USA:, v. 302, n. 2, p. 14–25, 1948. Citado na página 17.

BATISTA, G. E. d. A. P. A. **Pré-processamento de dados em aprendizado de máquina supervisionado**. Tese (Doutorado) — Universidade de São Paulo, São Carlos : Instituto de Ciências Matemáticas e de Computação, 2003. Citado na página 58.

BAZELATO, B. S.; AMORIM, E. A bayesian classifier to automatic correction of portuguese essays. In: **Conferência Internacional sobre Informática na Educação (TISE)**. [S.l.: s.n.], 2013. v. 18, p. 779–782. Citado na página 50.

BETSY. n.d. Disponível em: <https://edres.org/betsy/>. Citado na página 47.

BRASIL. **LEI Nº 11.096, DE 13 DE JANEIRO DE 2005.** 2005. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2004-2006/2005/lei/l11096.htm>. Citado na página 17.

BRASIL. **PORTARIA NORMATIVA Nº 2, DE 26 DE JANEIRO DE 2010**. 2010. Disponível em: <https://www.gov.br/mec/pt-br/media/acesso_informacacao/pdf/SISUPortariaNormativa2.pdf>. Citado na página 17.

BRASIL. Instituto nacional de estudos e pesquisas educacionais anísio teixeira (inep). **Plano Nacional de Educação PNE 2014-2024 : Linha de Base**, p. 404, 2015. Citado na página 17.

BRASIL. **A segunda maior prova de acesso ao ensino superior do mundo**. 2015. Disponível em: <http://portal.mec.gov.br/ultimas-noticias/418-enem-946573306/31151-a-segunda-maior-prova-de-acesso-ao-ensino-superior-do-mundo>. Citado na página 17.

BRASIL. Ministério da educação. **Base Nacional Comum Curricular**, 2018. Citado na página 17.

BRASIL. Instituto nacional de estudos e pesquisas educacionais anísio teixeira (inep). **A redação no Enem 2020: cartilha do participante**, 2020. Citado nas páginas 19, 20, 30 e 69.

BRASIL. Instituto nacional de estudos e pesquisas educacionais anísio teixeira (inep). **Entenda a sua nota no Enem: guia do participante**, 2021. Citado nas páginas 18 e 69.

BURSTEIN, J.; CHODOROW, M.; LEACOCK, C. Criterionsm online essay evaluation: An application for automated evaluation of student essays. In: **IAAI**. [S.l.: s.n.], 2003. p. 3–10. Citado nas páginas 19 e 47.

CARVALHO, M. R. V. de. Perfil do professor da educação básica. **Serie Documental: Relatos de Pesquisa. Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira**, n. 41, p. 67p., 2018. Citado na página 19.

CHEN, H.; HE, B. Automated essay scoring by maximizing human-machine agreement. In: **Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing**. Seattle, Washington, USA: [s.n.], 2013. p. 1741–1752. Citado na página 62.

CHEN, Y.-Y.; LIU, C.-L.; CHANG, T.-H.; LEE, C.-H. An unsupervised automated essay scoring system. **IEEE Intelligent Systems**, v. 25, n. 5, p. 61–67, 2010. Citado na página 62.

COSTA, L.; OLIVEIRA, E.; JúNIOR, A. C. **Corretor Automático de Redações em Língua Portuguesa: um mapeamento sistemático de literatura**. Porto Alegre, RS, Brasil: SBC, 2020. 1403–1412 p. Disponível em: <https://sol.sbc.org.br/index.php/sbie/article/view/12896>. Citado na página 50.

COSTA, L.; OLIVEIRA, E. H. T. de; JÚNIOR, A. C. Corretor automático de redações em língua portuguesa: um mapeamento sistemático de literatura. In: SBC. **Anais do XXXI Simpósio Brasileiro de Informática na Educação**. [S.l.], 2020. p. 1403–1412. Citado nas páginas 19 e 31.

DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2018. Citado na página 29.

DIKLI, S. An overview of automated scoring of essays. **The Journal of Technology, Learning and Assessment**, v. 5, n. 1, 2006. Citado nas páginas 31, 47 e 53.

DOKUZ, Y.; TUFEKCI, Z. Mini-batch sample selection strategies for deep learning based speech recognition. **Applied Acoustics**, v. 171, p. 107573, 2021. ISSN 0003-682X. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0003682X20306770>. Citado na página 46.

DOZAT, T. Incorporating nesterov momentum into adam. 2016. Citado na página 63.

DUCHI, J.; HAZAN, E.; SINGER, Y. Adaptive subgradient methods for online learning and stochastic optimization. **Journal of machine learning research**, v. 12, n. 7, 2011. Citado na página 63.

ELLIOT, S. Intellimetric: From here to validity. **Automated essay scoring: A cross-disciplinary perspective**, Lawrence Erlbaum Associates, p. 71–86, 2003. Citado na página 47.

ENADE. **CURSOS MAIS CONCORRIDOS SISU 2020: Veja quais são!** 2020. Disponível em: <https://enade.inf.br/cursos-mais-concorridos-sisu-2020/>. Citado na página 18.

FILHO, A. H.; PRADO, H.; FERNEDA, E.; NAU, J. An approach to evaluate adherence to the theme and the argumentative structure of essays. **Procedia Computer Science**, v. 126, p. 788–797, 01 2018. Citado na página 52.

FIRTH, J. R. A synopsis of linguistic theory. **Oxford: Philological Society**, 1957. Reprinted in F. Palmer (ed.)(1968). Studies in Linguistic Analysis 1930-1955. Selected Papers of J.R. Firth., Harlow: Longman. Citado na página 26.

FITZSIMMONS, P. R.; MICHAEL, B.; HULLEY, J. L.; SCOTT, G. O. A readability assessment of online parkinson's disease information. **The journal of the Royal College of Physicians of Edinburgh**, v. 40, n. 4, p. 292–296, 2010. Citado na página 59.

FLASIŃSKI, M. **Introduction to artificial intelligence**. [S.l.]: Springer, 2016. Citado na página 35.

FOLTZ, P. W.; LAHAM, D.; LANDAUER, T. K. The intelligent essay assessor: Applications to educational technology. **Interactive Multimedia Electronic Journal of Computer-Enhanced Learning**, v. 1, n. 2, p. 939–944, 1999. Citado na página 47.

FONSECA, E.; MEDEIROS, I.; KAMIKAWACHI, D.; BOKAN, A. Automatically grading brazilian student essays. In: SPRINGER. **International Conference on Computational Processing of the Portuguese Language**. [S.l.], 2018. p. 170–179. Citado nas páginas 52, 53, 66 e 70.

GER, F. **Long Short-Term Memory in Recurrent Neural Networks**. Tese (Doutorado) — Ecole Polytechnique Federale de Lausanne, 2001. Citado na página 42.

GOLDBERG, Y.; HIRST, G. **Neural Network Methods in Natural Language Processing**. [S.l.]: Morgan Claypool Publishers, 2017. Citado na página 27.

GOODFELLOW, I. J.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Cambridge, MA, USA: MIT Press, 2016. <http://www.deeplearningbook.org>. Citado na página 36.

GRAVES, A. **Supervised Sequence Labelling with Recurrent Neural Networks**. Berlin: Springer, 2012. (Studies in Computational Intelligence). Disponível em: <https://cds.cern.ch/record/1503877>. Citado nas páginas 42 e 43.

GRAVES, A.; WAYNE, G.; DANIHELKA, I. **Neural Turing Machines**. 2014. Citado na página 42.

HAN, J.; PEI, J.; KAMBER, M. **Data mining: concepts and techniques**. [S.l.]: Elsevier, 2011. Citado na página 61.

HARTMANN, N.; FONSECA, E.; SHULBY, C.; TREVISO, M.; RODRIGUES, J.; ALUISIO, S. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. **arXiv preprint arXiv:1708.06025**, 2017. Citado na página 56.

HAYKIN, S. S. **Neural networks and learning machines**. Third. Upper Saddle River, NJ: Pearson Education, 2009. Citado nas páginas 36, 37, 38, 39, 40, 41, 42, 45 e 65.

HINTON, G.; SRIVASTAVA, N.; SWERSKY, K. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. **Cited on**, v. 14, n. 8, p. 2, 2012. Citado na página 63.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural Computation**, v. 9, n. 8, p. 1735–1780, 1997. Citado na página 42.

HYLAND, K. Persuasion and context: The pragmatics of academic metadiscourse. **Journal of pragmatics**, Elsevier, v. 30, n. 4, p. 437–455, 1998. Citado na página 19.

INEP. **Material de leitura - Módulo 03**. 2020. Disponível em: <https://download.inep.gov.br/educacao_basica/enem/downloads/2020/Competencia_1.pdf>. Citado nas páginas 20, 30, 31, 32, 69, 80 e 81.

JACK, C. **Alan Turing: The codebreaker who saved 'millions of lives'**. 2012. Disponível em: <http://portal.mec.gov.br/ultimas-noticias/418-enem-946573306/31151-a-segunda-maior-prova-de-acesso-ao-ensino-superior-do-mundo>. Citado na página 23.

JÚNIOR, J. A. S. B. *et al.* Avaliação automática de redação em língua portuguesa empregando redes neurais profundas. Universidade Federal de Goiás, 2020. Citado na página 53.

JÚNIOR, J. J. d. S. *et al.* Avaliação léxico-sintática de atividades escritas em algoritmo genético e processamento de linguagem natural: Um experimento no enem. **Revista Brasileira de Informática na Educação**, v. 24, n. 02, p. 92, 2016. ISSN 2317-6121. Disponível em: <https://www.br-ie.org/pub/index.php/rbie/article/view/6450>. Citado nas páginas 50 e 51.

JÚNIOR, J. J. d. S. *et al.* Modelos e técnicas para melhorar a qualidade da avaliação automática para atividades escritas em língua portuguesa brasileira. Universidade Federal de Alagoas, 2017. Citado na página 51.

JURAFSKY, D.; MARTIN, J. H. **Speech and Language Processing (2Nd Edition)**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2009. ISBN 0131873210. Citado na página 23.

KE, Z.; NG, V. Automated essay scoring: A survey of the state of the art. In: **IJCAI**. [S.l.: s.n.], 2019. v. 19, p. 6300–6308. Citado nas páginas 19, 20, 31, 47, 48, 49, 50 e 53.

KINCAID, J. P.; JR, R. P. F.; ROGERS, R. L.; CHISSOM, B. S. **Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel**. [S.l.], 1975. Citado na página 59.

LEARNING, V. n.d. Disponível em: <https://www.vantagelearning.com/>. Citado na página 47.

LUGER, G. F. **Artificial intelligence: structures and strategies for complex problem solving**. [S.l.]: Pearson education, 2005. Citado na página 24.

MARINHO, J.; ANCHIêTA, R.; MOURA, R. Essay-br: a brazilian corpus of essays. In: **Anais do III Dataset Showcase Workshop**. Porto Alegre, RS, Brasil: SBC, 2021. p. 53–64. ISSN 0000-0000. Disponível em: <https://sol.sbc.org.br/index.php/dsw/article/view/17414>. Citado nas páginas 55, 60 e 66.

MARINHO, J. C.; ANCHIETA, R. T.; MOURA, R. S. **Essay-BR: a Brazilian Corpus of Essays**. 2021. Citado nas páginas 21, 56, 61, 65 e 66.

MCMAHAN, H. B.; HOLT, G.; SCULLEY, D.; YOUNG, M.; EBNER, D.; GRADY, J.; NIE, L.; PHILLIPS, T.; DAVYDOV, E.; GOLOVIN, D. n. *et al.* Ad click prediction: a view from the trenches. In: **Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining**. [S.l.: s.n.], 2013. p. 1222–1230. Citado na página 63.

MEC. **Base Nacional Comum Curricular — Educação é a Base: Ensino Médio**. 2018. Ministério da Educação, MEC. Citado na página 29.

MELO, R.; FREITAS, A.; FRANCISCO, E.; MOTOKANE, M. Impacto das variáveis socioeconômicas no desempenho do enem: uma análise espacial e sociológica. **Revista de Administracao Publica**, v. 55, 11 2021. Citado na página 18.

MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. Efficient estimation of word representations in vector space. In: BENGIO, Y.; LECUN, Y. (Ed.). **1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings**. [S.l.: s.n.], 2013. Citado na página 26.

MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. **Efficient Estimation of Word Representations in Vector Space**. 2013. Citado na página 27.

MIKOLOV, T.; SUTSKEVER, I.; CHEN, K.; CORRADO, G.; DEAN, J. **Distributed Representations of Words and Phrases and their Compositionality**. 2013. Citado na página 57.

NETO, S. S. C.; FAVERO, E. L.; SANTOS, J. C. A. dos; FREITAS, S. N. de; JÚNIOR, M. A. N. Avaliação automática de redações na língua portuguesa baseada na coleta de atributos e aprendizagem de máquina. In: SBC. **Anais do XXXI Simpósio Brasileiro de Informática na Educação**. [S.l.], 2020. p. 1162–1171. Citado na página 53.

OLAH, C. **Understanding LSTM Networks**. 2015. Disponível em: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>. Acesso em: 1 dez. 2020. Citado nas páginas 42 e 43.

OLIVEIRA, C. A. J. e Marcos Spalenza e Elias de. Proposta de um sistema de avaliação automática de redações do enem utilizando técnicas de aprendizagem de máquina e processamento de linguagem natural. **Anais do Computer on the Beach**, v. 0, n. 0, p. 474–483, 2017. ISSN 2358-0852. Disponível em: <https://siaiap32.univali.br/seer/index.php/acotb/article/view/10592>. Citado na página 51.

OLIVEIRA, E.; ALVES, J.; BRITO, J.; PIROVANI, J. The influence of ner on the essay grading. In: SPRINGER. **International Conference on Intelligent Systems Design and Applications**. [S.l.], 2019. p. 162–171. Citado na página 52.

PAGE, E. B. The imminence of... grading essays by computer. **The Phi Delta Kappan**, JSTOR, v. 47, n. 5, p. 238–243, 1966. Citado na página 47.

PAGE, E. B. Grading essays by computer: Progress report. In: **Proceedings of the Invitational Conference on Testing Problems**. [S.l.: s.n.], 1967. p. 87–100. Citado nas páginas 19, 29 e 47.

PAGE, E. B. The use of the computer in analyzing student essays. **International Review of Education**, v. 14, n. 2, p. 210–225, Jun 1968. ISSN 1573-0638. Disponível em: <https://doi.org/10.1007/BF01419938>. Citado na página 29.

PENNINGTON, J.; SOCHER, R.; MANNING, C. GloVe: Global vectors for word representation. In: **Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)**. Doha, Qatar: [s.n.], 2014. p. 1532–1543. Citado nas páginas 27, 28 e 57.

PETERS, M. E.; NEUMANN, M.; IYYER, M.; GARDNER, M.; CLARK, C.; LEE, K.; ZETTLEMOYER, L. **Deep contextualized word representations**. 2018. Citado na página 29.

PILEHVAR, M. T.; COLLADOS, J. C. Embeddings in natural language processing: Theory and advances in vector representations of meaning. **Synthesis Lectures on Human Language Technologies**, v. 13, n. 4, p. 1–175, 2020. Disponível em: <https://doi.org/10.2200/S01057ED1V01Y202009HLT047>. Citado nas páginas 24, 25, 26 e 27.

PONTIUS, R. G.; THONTTEH, O.; CHEN, H. Components of information for multiple resolution comparison between maps that share a real variable. **Environmental and Ecological Statistics**, v. 15, n. 2, p. 111–142, Jun 2008. Disponível em: <https://doi.org/10.1007/s10651-007-0043-y>. Citado na página 62.

RAMISCH, R. Caracterização de desvios sintáticos em redações de estudantes do ensino médio: subsídios para o processamento automático das línguas naturais. Universidade Federal de São Carlos, 2020. Citado na página 53.

RUDER, S. **An overview of gradient descent optimization algorithms**. 2017. Citado na página 39.

RUSSELL, S.; NORVIG, P. **Artificial intelligence: a modern approach**. 3. ed. [S.l.]: Upper Saddle River, NJ : Prentice Hall, 2010. 20-83, 879-989 p. Citado nas páginas 23 e 35.

SALTON, G.; WONG, A.; YANG, C.-S. A vector space model for automatic indexing. **Commun. ACM**, v. 18, p. 613–620, 1975. Citado na página 25.

SENTER, R.; SMITH, E. A. **Automated readability index**. [S.l.], 1967. Citado na página 59.

SHERMIS, M. D.; BURSTEIN, J. C. **Automated essay scoring: A cross-disciplinary perspective**. [S.l.]: Routledge, 2003. 113-–121 p. Citado na página 47.

SUTSKEVER, I.; MARTENS, J.; DAHL, G.; HINTON, G. On the importance of initialization and momentum in deep learning. In: PMLR. **International conference on machine learning**. [S.l.], 2013. p. 1139–1147. Citado na página 63.

TAGHIPOUR, K.; NG, H. T. A neural approach to automated essay scoring. In: **Proceedings of the 2016 conference on empirical methods in natural language processing**. [S.l.: s.n.], 2016. p. 1882–1891. Citado na página 60.

TRAVITZKI, R. **ENEM: limites e possibilidades do Exame Nacional do ensino Médio enquanto indicador da qualidade escolar**. Tese (Doutorado) — Universidade de São Paulo, Faculdade de Educação, São Paulo, 2013. Citado na página 17.

TURING, A. M.; HAUGELAND, J. **Computing machinery and intelligence**. [S.l.]: MIT Press Cambridge, MA, 1950. Citado na página 23.

UFBA. 2020. Disponível em: <https://ingresso.ufba.br/sites/ingresso.ufba.br/files/termo_adesao_20201_ufba.pdf>. Citado na página 18.

UFF. 2020. Disponível em: <http://www.coseac.uff.br/20211/arquivos/UFF-TermodeAdesao-SISU2021-1.pdf>. Citado na página 18.

UFG. 2020. Disponível em: <https://t.co/AN5b0KQcGi>. Citado na página 18.

UFPE. 2020. Disponível em: <https://sipac.ufpe.br/public/baixarBoletim.do?publico=true&idBoletim=57>. Citado na página 18.

UFRJ. 2020. Disponível em: <https://acessograduacao.ufrj.br/processos/2021-1/2021-1-sisu-mec/termo-de-adesao-ufrj-sisu-mec-2021-1/2021_1-Termo_de_Adesao_SiSU.pdf>. Citado na página 18.

UNESCO. **UNESCO STRATEGY FOR YOUTH AND ADULT LITERACY (2020-2025)**. 2019. Disponível em: <https://epale.ec.europa.eu/sites/default/files/unesco_strategy_for_youth_and_adult_literacy.pdf>. Citado na página 17.

UTO, M. A review of deep-neural automated essay scoring models. **Behaviormetrika**, Springer, v. 48, n. 2, p. 459–484, 2021. Citado nas páginas 20, 31, 32 e 48.

UTO, M.; XIE, Y.; UENO, M. Neural automated essay scoring incorporating handcrafted features. In: **Proceedings of the 28th International Conference on Computational Linguistics**. [S.l.: s.n.], 2020. p. 6077–6088. Citado nas páginas 31, 49, 59 e 67.

WHISNER, M. When judges scold lawyers. **Law Libr. J.**, HeinOnline, v. 96, p. 557, 2004. Citado na página 59.

YANNAKOUDAKIS, H.; CUMMINS, R. Evaluating the performance of automated text scoring systems. In: **Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications**. Denver, Colorado: Association for Computational Linguistics, 2015. p. 213–223. Disponível em: <https://aclanthology.org/W15-0625>. Citado na página 62.

ZEILER, M. D. **ADADELTA: An Adaptive Learning Rate Method**. 2012. Citado na página 63.

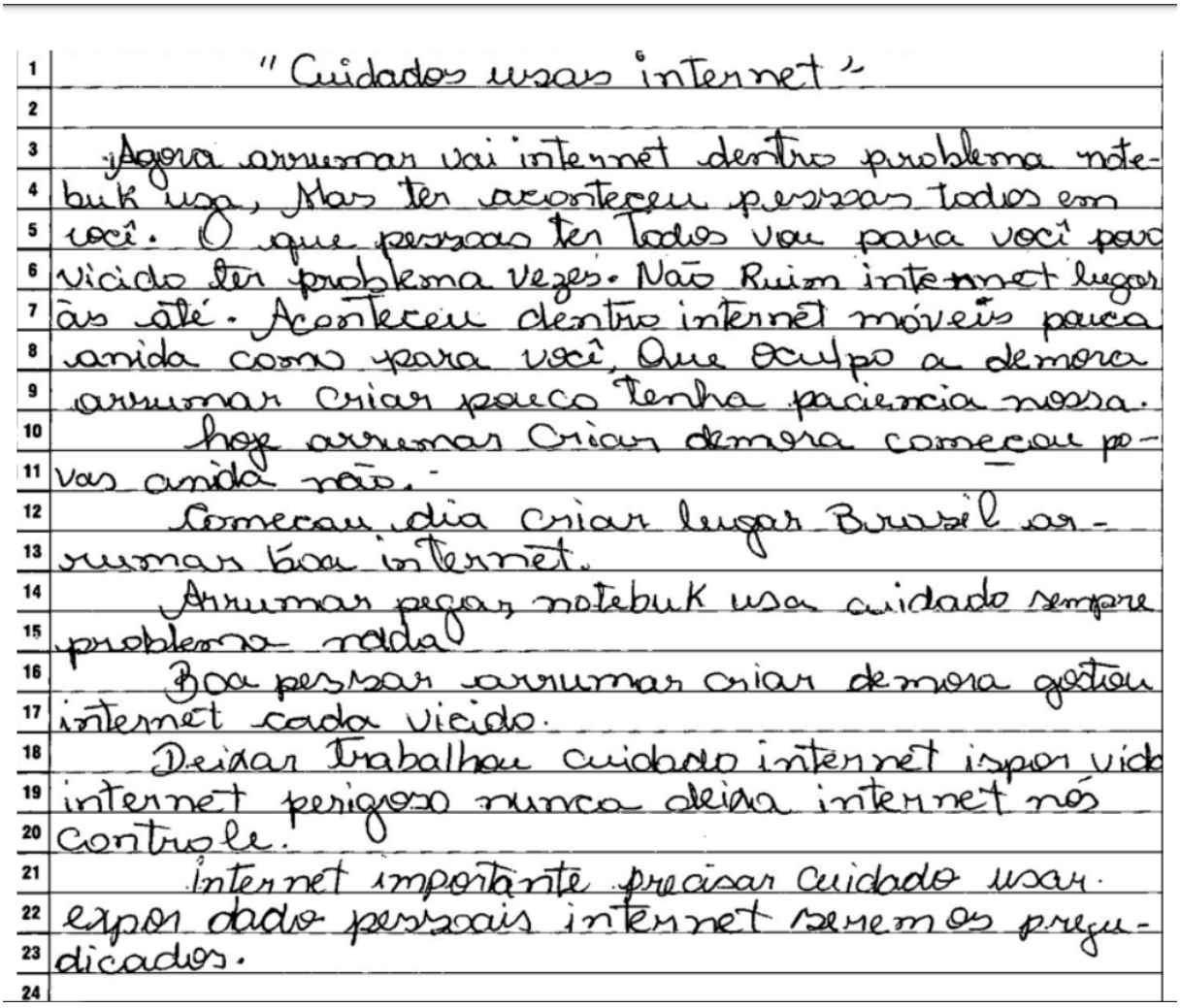
ZHANG, A.; LIPTON, Z. C.; LI, M.; SMOLA, A. J. **Dive into Deep Learning**. [S.l.: s.n.], 2019. <http://www.d2l.ai>. Citado na página 46.

ANEXO

A

REDAÇÕES

Figura 1 – Redação do ENEM de nível 0 para Competência 1

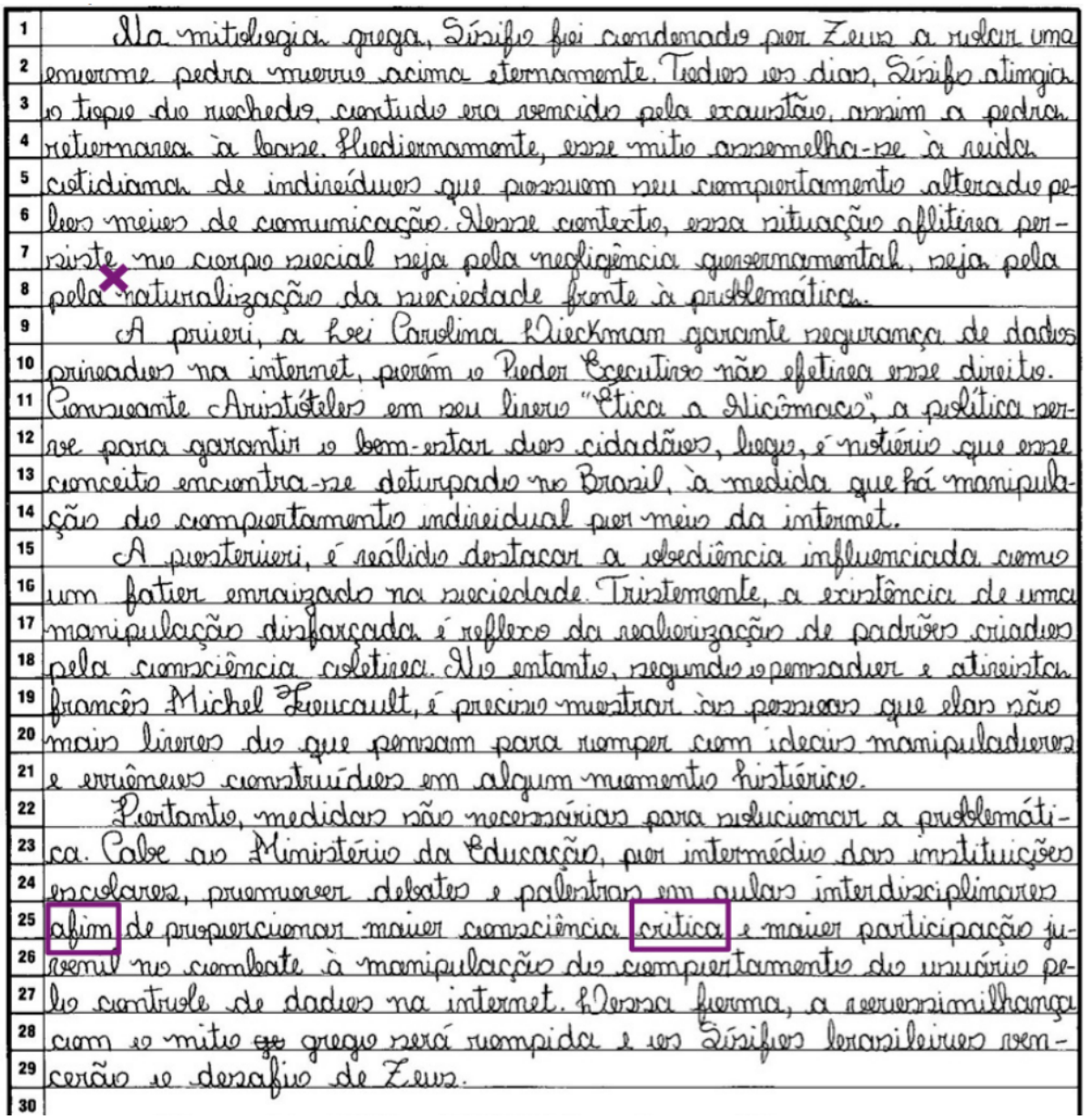


"Cuidados usas internet"

Agora arrumar vai internet dentro problema note-
buk usa, Mas ter acontecer pessoas todos em
você. O que pessoas ter todos vou para você pas
vicido ter problema vezes. Não Ruim internet lugar
às até. Acontecer dentro internet móveis pouca
anida como para você, Que Culpo a demora
arrumar Criar pouco tenha paciencia nossa.
hoje arrumar Criar demora começar po-
vas anida não.
Começar dia Criar lugar Brasil ar-
rumar boa internet.
Arrumar pegar notebuk usa cuidado sempre
problema nada
Boa pessoar arrumar criar demora gostou
internet cada vicido.
Deixar Trabalhar cuidado internet ispor vida
internet perigoso nunca deixa internet nós
Controle.
Internet importante precisar cuidado usar.
expor dado pessoais internet nenem os preju-
dicados.

Fonte: (INEP, 2020)

Figura 2 – Redação do ENEM de nível 5 para Competência 1



Na mitologia grega, Sísifo foi condenado por Zeus a rolar uma enorme pedra morro acima eternamente. Todos os dias, Sísifo atingia o topo do rochedo, contudo era vencido pela exaustão, assim a pedra retornava à base. Hodiernamente, esse mito assemelha-se à vida cotidiana de indivíduos que possuem seu comportamento alterado pelos meios de comunicação. Nesse contexto, essa situação aflitiva persiste no corpo social seja pela negligência governamental, seja pela pela naturalização da sociedade frente à problemática.

A priori, a Lei Carolina Dieckman garante segurança de dados privados na internet, porém o Poder Executivo não efetiva esse direito. Consoante Aristóteles em seu livro "Ética a Nicômaco", a política serve para garantir o bem-estar dos cidadãos, logo, é notório que esse conceito encontra-se deturpado no Brasil, à medida que há manipulações do comportamento individual por meio da internet.

A posteriori, é válido destacar a obediência influenciada como um fator enraizado na sociedade. Tristemente, a existência de uma manipulação disfarçada é reflexo da realização de padrões criados pela consciência coletiva. No entanto, segundo o pensador e ativista francês Michel Foucault, é preciso mostrar às pessoas que elas não mais livres do que pensam para romper com ideais manipuladores e errôneos construídos em algum momento histórico.

Portanto, medidas são necessárias para solucionar a problemática. Cabe ao Ministério da Educação, por intermédio das instituições escolares, promover debates e palestras em aulas interdisciplinares afim de proporcionar maior consciência crítica e maior participação juvenil no combate à manipulação do comportamento do usuário pelo controle de dados na internet. Dessa forma, a semelhança com o mito ~~gr~~ grego será rompida e os Sísifos brasileiros vencerão o desafio de Zeus.

Fonte: (INEP, 2020)