UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE COMPUTAÇÃO
CURSO DE CIÊNCIA DA COMPUTAÇÃO

LEVI MATHEUS MARTINS SANGE

CLASSIFICADOR DE LEGIBILIDADE DE TEXTOS EM LÍNGUA INGLESA

TRABALHO DE CONCLUSÃO DE CURSO

MEDIANEIRA

2021

LEVI MATHEUS MARTINS SANGE

CLASSIFICADOR DE LEGIBILIDADE DE TEXTOS EM LÍNGUA INGLESA

Trabalho de Conclusão de Curso apresentado ao Departamento Acadêmico de Computação da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de "Bacharel em Ciência da Computação".

Orientador: Prof. Arnaldo Candido Junior

MEDIANEIRA

2021

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Diretoria de Graduação e Educação Profissional
Coordenação do Curso de Ciência da Computação

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

TERMO DE APROVAÇÃO

CLASSIFICADOR DE LEGIBILIDADE DE TEXTOS EM LÍNGUA INGLESA

Por

LEVI MATHEUS MARTINS SANGE

Este Trabalho de Conclusão de Curso foi apresentado às 10:10h do dia 17 de Agosto de 2021 como requisito parcial para a obtenção do título de Bacharel no Curso de Ciência da Computação, da Universidade Tecnológica Federal do Paraná, Câmpus Medianeira. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Arnaldo Candido Junior
UTFPR - Câmpus Medianeira

Prof. Evando Carlos Pessini
UTFPR - Câmpus Medianeira

Prof. Pedro Luiz de Paula Filho
UTFPR - Câmpus Medianeira

Prof. Jorge Aikes Junior
UTFPR - Câmpus Medianeira

A folha de aprovação assinada encontra-se na Coordenação do Curso.

RESUMO

SANGE, Levi Matheus Martins. CLASSIFICADOR DE LEGIBILIDADE DE TEXTOS EM LÍNGUA INGLESA. 47 f. Trabalho de Conclusão de Curso – Curso de Ciência da Computação, Universidade Tecnológica Federal do Paraná. Medianeira, 2021.

A língua inglesa alcançou o nível de língua global ou globalizada, ou seja, escolhida como intermediadora entre as comunicações mundialmente devido as suas características como vocabulário extenso, junção de outras línguas, facilidade de aprendizado, além de ter sido aceita com maior apreço. Devido à esse status e poder alcançado, a fluência nessa língua tem sido requisito em diversos setores e áreas. Consequentemente, gerou-se um aumento no número de pessoas interessadas em obter proficiência e domínio. Algumas das maneiras de melhorar as habilidades são através do ler e escutar, leituras e textos permitem a descoberta de diversos aspectos da língua, entretanto, procurar uma leitura em que há proximidade com o nível e conhecimento da língua inglesa do leitor, sem ajuda, pode ser muito desmotivante. Porém, esta busca tem sido facilitada com o avanço da inteligência artificial e das técnicas de processamento de língua natural, que permitem, em conjunto com *datasets* de textos, gerar resultados como características dos conteúdos, permitindo categorizá-los conforme o nível de conhecimento, por exemplo, de um usuário na língua inglesa. Para o escopo deste trabalho foram utilizadas técnicas de Inteligência Artificial e Aprendizado de Máquina com algoritmos como o *Naive Bayes*, Máquina de vetor suporte e Árvores de decisão, para gerar classificações e Processamento de Língua Natural sobre dois *datasets* disponíveis gratuitamente na internet, estas possuem milhares de definições de palavras. O objetivo principal do trabalho foi desenvolver um classificador de legibilidade de textos em língua inglesa, a partir da aplicação dos algoritmos de aprendizagem de máquina supracitados. Foram analisadas métricas e características textuais, extraídas de cada artigo do *dataset Wikipedia* e *Simple Wikipedia*. Foram alcançados através do treinamento dos algoritmos, com destaque para o algoritmo J48 de Árvore de decisão, uma acurácia de 94,17%, realçando como atributos textuais importantes, a frequência de palavras complexas, o índice de *Gunning Fog*, verbos auxiliares e *to be*. Alguns itens como *datasets* pré-processados e *scripts* foram gerados e disponibilizados gratuitamente em repositório[1] online com o objetivo de contribuir para pesquisas e trabalhos futuros na área. Através da utilização do classificador desenvolvido é possível a construção de, por exemplo, ferramentas e sistemas de recomendação de conteúdo para usuários aprendizes da língua inglesa como segunda língua ou para pessoas interessadas em desenvolver sua capacidade de leitura. Com estes resultados, tem-se então, mais espaço para pesquisas e desenvolvimento de ferramentas gratuitas complementares na área de legibilidade e inteligibilidade de textos através do Processamento de Língua Natural e Aprendizado de Máquina.

Palavras-chave: Língua inglesa, Inteligência Artificial, Aprendizado do computador

[1] https://github.com/LeviMatheus/tcc-readability-score-level

ABSTRACT

SANGE, Levi Matheus Martins. ENGLISH TEXT READABILITY CLASSIFIER. 47 f. Trabalho de Conclusão de Curso – Curso de Ciência da Computação, Universidade Tecnológica Federal do Paraná. Medianeira, 2021.

The English language has reached the level of a global or globalized language, that is, chosen as an intermediary among communications worldwide due to its characteristics such as extensive vocabulary, combination of other languages, ease of learning. Due to this status and power achieved, fluency in this language has been a requirement in several sectors and areas. Consequently, there was an increase in the number of people interested in obtaining proficiency and mastery. Some of the ways to improve skills are through reading and listening, readings and texts allow the discovery of various aspects of the language, however, looking for a reading that is close to the reader's level and knowledge of the English language, without help, can be very demotivating. However, this search has been facilitated with the advancement of artificial intelligence and natural language processing techniques, which allow, together with text datasets, to generate results as content characteristics, allowing them to be categorized. For the scope of this work, Artificial Intelligence and Machine Learning techniques were used with algorithms such as Naive Bayes, Support Vector Machine and Decision Trees, to generate classifications and Natural Language Processing on two datasets freely available on the Internet. These datasets have thousands of words. The main objective of this work was to develop a readability classifier for texts in English, based on the application of the aforementioned machine learning algorithms. Metrics and textual characteristics were analyzed, such as number of syllables, frequency of long and complex words, readability formulas, extracted from each articles in the Wikipedia and Simple Wikipedia datasets. An accuracy of 94,17% was achieved through the training of the algorithms, with emphasis on the J48 Decision Tree algorithm, highlighting as important textual attributes, the frequency of complex words, the Gunning Fog index, auxiliary and to be verbs. Some items such as pre-processed datasets and scripts were also generated and made available for free in an online repository in order to contribute to future research and work in the area. Through the use of the developed classifier, it is possible to build, for example, tools and content recommendation systems for users who learn English as a second language or for people interested in developing their reading skills. With these results, more space was opened up for research and development of complementary free tools in the area of readability and intelligibility of texts through Natural Language Processing and Machine Learning.

Keywords: Artificial Intelligence, English language, Machine Learning

A Deus por ser tudo em minha vida, sem Ele nada existiria e nada teria sentido em existir.

A Universidade Federal Tecnológica câmpus Medianeira, que disponibilizou todo o conhecimento necessário para que eu pudesse seguir em frente com meu sonho.

A minha família e a minha maravilhosa esposa pelo amor, dedicação, apoio e motivação.

AGRADECIMENTOS

Aos meus professores, amigos e companheiros de curso.

Ao meu orientador do Trabalho de Conclusão do Curso Arnaldo Cândido Júnior, por ter me auxiliado na realização do trabalho, ter fornecido informações indispensáveis para o desenvolvimento.

A Universidade Tecnológica Federal do Paraná campus Medianeira por todo suporte deste trabalho.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

ARFF	*Attribute-Relation File Format*
ASL	*Average sentence length*
ASW	*Average syllable per Word*
CSV	*Comma-separated values*
KDD	Knowledge-discovery in databases
PLN	Processamento de Língua Natural
SVM	Support Vector Machine
URL	*Uniform Resource Locator*
WEKA	*Waikato Environment for Knowledge Analysis*

SUMÁRIO

1 INTRODUÇÃO

Proveniente do ocidente germânico, a língua inglesa possui um vocabulário extenso formada a partir da incorporação de outras línguas. As inúmeras transformações e eventos ocorridos mundialmente impulsionaram a língua inglesa à condição de língua globalizada. Segundo Baker (2018), alguns dos fatores que também contribuíram para a língua inglesa ser utilizada globalmente são sua propagação geográfica e diversidade cultural dos que a falam. Como segunda língua aprendida por uma pessoa permite a comunicação quase mundialmente. Reconhecendo a importância da língua inglesa, o mercado de trabalho têm, muitas vezes, adicionado o conhecimento dela como requisito para oportunidades de emprego. O que antes era um diferencial no currículo de uma pessoa que está a procura de um emprego, vaga ou setor para atuar, atualmente quase é um pré-requisito para, por exemplo, quem atua na Indústria, Finanças, Recursos humanos, área de TI e Comunicação. Acredita-se que cientes desta exigência por parte do mercado de trabalho as pessoas tendem a procurar mais e diferentes formas de obter conhecimento e proficiência na língua inglesa. Porém, conforme British Council (2014), os especialistas, professores e até governos reconhecem que o ensino de inglês na educação básica, privada ou pública, no Brasil, têm formado poucos estudantes com proficiência no idioma, portanto o ensino tem sido resumido a noções iniciais das regras gramaticais, leitura de textos curtos e desenvolvimento da habilidade de testes de múltipla escolha voltados para exames avaliativos.

Tendo em vista este cenário em relação ao aprendizado da língua inglesa, seria de grande benefício o desenvolvimento de ferramentas capazes de complementar processos já existentes de aprendizagem, por exemplo a leitura, principal processo de aprendizagem citado neste trabalho. Nesse sentido, existem formas de utilizar máquinas para alcançar um objetivo, como a classificação de textos, esse objetivo de classificação pode ser alcançado através de treinamento de algoritmos. Um algoritmo de classificação é uma categoria de aprendizado de máquina supervisionado que possui potencial para, por exemplo, rotular uma leitura ou texto em níveis com base em características presentes em textos, que podem ser obtidas através de bibliotecas e fórmulas. Estas auxiliam os algoritmos de aprendizado na tarefa de classificar a dificuldade de leitura e compreensão de textos. Com o desenvolvimento de um classificador de

legibilidade de textos é possível expandir o número de ferramentas gratuitas e de fácil acesso para a área de Processamento de Língua Natural e de aprendizagem da língua inglesa. sendo mais um passo positivo para a educação e pesquisa.

1.1 OBJETIVOS GERAL E ESPECÍFICOS

Esse trabalho teve como objetivo desenvolver um classificador de legibilidade de textos capaz de classificar textos em níveis de dificuldade de leitura, utilizando algoritmos de classificação como J48, LibSVM e NaiveBayes. Esse objetivo principal pode ser dividido nos seguintes objetivos específicos:

- Selecionar *datasets* públicos em língua inglesa, sendo estes, o *Wikipedia* e *Simple Wikipedia*;
- Validar e tratar os *datasets* linguísticos.
- Treinar os algoritmos de aprendizado de máquina para a classificação. Os algoritmos selecionados para o treinamento são o *Naive Bayes*, Árvore de decisão e Máquina de vetor suporte;
- Analisar os resultados dos algoritmos e aplicabilidade do classificador desenvolvido.

1.2 JUSTIFICATIVA

As principais dificuldades encontradas pelos Brasileiros no processo do ensino público da língua inglesa são, a pouca estrutura para um ensino adequado da língua, turmas com número elevado de alunos, duração do curso muito extensa ou carga horária insuficiente e dificuldade de encontrar professores com formação adequada conforme demonstra a Figura 1. Então o ensino, conforme o British Council (2014) mostra, acaba focando somente em noções básicas da língua, leituras leves e testes de múltipla escolha que, muitas vezes são apenas focados em vestibulares. Existem parâmetros nacionais que dizem como será o currículo das matérias, mas que no dia a dia não conseguem ser plenamente aplicados, unindo tal problema a falta de estrutura e recursos, alguns dos exercícios de comunicação acabam sendo prejudicados.

Existem ferramentas para apoiar alunos e professores na aprendizagem e no ensino, sendo, a maior parte delas focadas nas habilidades de leitura (*reading*) e compreensão de falas (*listening*), entretanto, se incluído o aspecto de considerar o nível de conhecimento do usuário sobre a língua inglesa e a questão de ser uma ferramenta de acesso gratuito, esta quantidade irá reduzir significativamente. Com o desenvolvimento de um classificador de fácil acesso e gratuito sobre *datasets* apropriados é possível classificar a legibilidade de textos em níveis sendo mais uma contribuição para a área de Processamento de Língua Natural e de aprendizagem da língua inglesa. Possibilitando a construção de ferramentas e sistemas de recomendação de conteúdo que reduzem o tempo de pesquisa e procura por parte dos usuários em busca de textos e leituras niveladas ao seu conhecimento na língua inglesa.

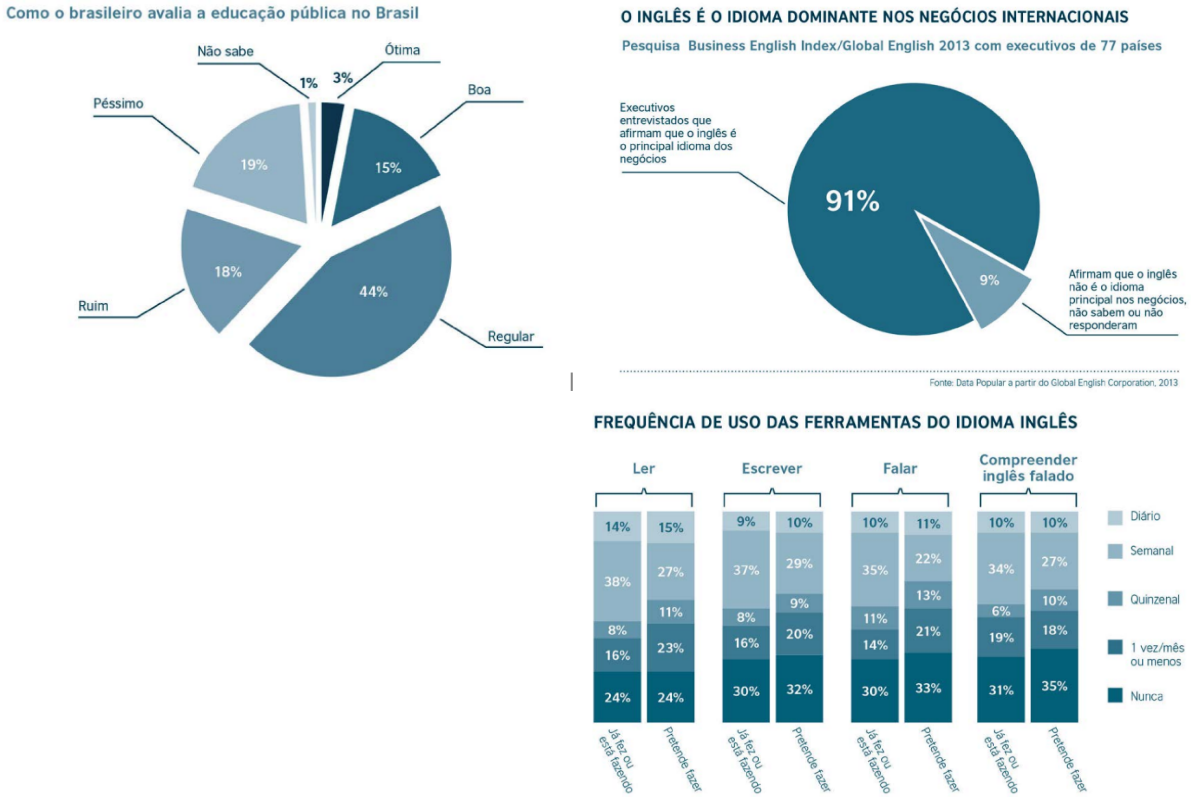

Figura 1 – Data Popular a partir da Pesquisa de Opinião Pública/Data Senado, 2011.

Fonte: Adaptado de Data Senado (2011)

2 REFERENCIAL TEÓRICO

Neste capítulo serão descritos o papel e importância da língua inglesa (Seção 2.1), técnicas de Processamento de Língua Natural (Seção 2.2) úteis no contexto da língua inglesa, aprendizado de máquina (Seção 2.3) área base para entendimento de como é possível para os computadores, aprender e representar conhecimento, assim também gerar resultados e tomada de decisões, e alguns algoritmos de classificação (Seção 2.4) que são técnicas e abordagens diferentes que utilizam dados para gerar o aprendizado de máquina.

2.1 LÍNGUA INGLESA

A língua inglesa tornou-se uma língua global devido a vários fatores, sendo os principais os aspectos sociais, históricos e políticos. Segundo Lopes (2008), o papel da língua é explicado pela importância do império Britânico nos séculos passados, e pela predominância da economia dos Estados Unidos a partir da segunda guerra mundial.

Uma língua atinge nível global quando tem seu papel específico reconhecido em diversos países ao redor do mundo Melitz (2016). Neste sentido, a língua inglesa destaca-se devido a não ser falado apenas no país de origem, mas sim como uma segunda língua que foi aos poucos sendo falada em diferentes países ao redor do mundo, ganhando seu lugar especial nas comunidades globalmente. Setores do governo, mídia e redes sociais a utilizam como intermediária com foco de atingir um público maior devido a aceitação, além de muitos países escolherem aplicá-la na educação como uma língua estrangeira. A língua estrangeira mais falada mundialmente, em cerca de 100 países como China, Rússia, Alemanha, Espanha, Egito e Brasil (MELITZ, 2016).

2.2 PROCESSAMENTO DE LÍNGUA NATURAL

Conforme Covington (2013) Processamento de Língua Natural (PLN) é o uso de computadores para entender as línguas humanas (naturais), como o inglês. É uma área da Inteligência Artificial que estuda como representar, reconhecer, extrair conhecimento e compreender linguagem natural de forma automática pelas máquinas utilizando técnicas, algoritmos e soluções.

> O Homo sapiens é diferente das demais espécies pela capacidade de linguagem. Em algum lugar a cerca de 100.000 anos, humanos aprenderam a falar, e a 7.000 anos aprenderam a escrever. Embora chimpanzés, golfinhos e outros animais tenham mostrado vocabulários de centenas de sinais, apenas os humanos podem comunicar de maneira confiável em um número ilimitado de diferentes mensagens sobre qualquer tópico usando sinais discretos. (RUSSELL; NORVIG, 2013, p. 860).

Com base na definição exposta, pode-se observar que a tarefa do PLN não é tão fácil para a máquina devido a sua complexidade, sua ambiguidade e diversos outros fatores que dificultam o processamento de uma língua natural. Entretanto, é uma tarefa viável, e para entendimento deste trabalho, serão apresentadas nesta seção as tarefas necessárias e técnicas que podem ser utilizadas.

2.2.1 Pré-processamento de dados

Segundo Addrians e Zantinge (1996) *Knowledge-discovery in databases* (KDD) é a extração não trivial de conhecimento previamente desconhecido e potencialmente útil de um banco de dados.

A Figura 2 apresenta as atividades necessárias para o KDD. A partir de uma base de dados é feita a seleção inicial de dados, gerando um *subdataset* ou conjunto de dados reduzido o qual será efetivamente utilizado para a geração da descoberta do conhecimento. Como uma segunda etapa é necessário o pré-processamento dos dados, conforme Alves (2003) diz, o pré-processamento de dados é uma etapa semi-automática em que depende da capacidade da pessoa que conduz a identificação de problemas nos dados e sua natureza, utilizando diferentes métodos para solucioná-los, e segundo Fayyad et al. (1996) pré-processar dados consistem em modificá-los para um formato mais apropriado, remover ruídos, impurezas, tratar dados

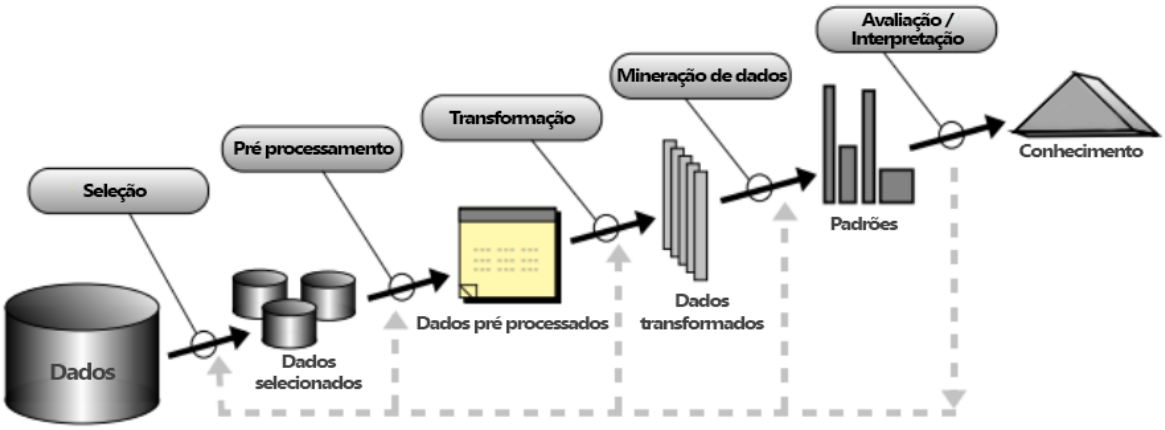

Figura 2 – Etapas do processo de descoberta de conhecimento.
Fonte: Adaptado de Fayyad et al. (1996)

duplicados, entre outras operações, a etapa de limpeza, consiste em analisar e tratar a base de dados a procura por informações corrompidas, imprecisas ou desnecessárias de acordo com o objetivo estabelecido. Isso é feito, removendo ou modificando dados a fim de tornar a base mais consistente e assim evitar resultados falsos ou imprecisos após a mineração. A redução dos dados, etapa na qual é feita a transformação de um conjunto de dados para uma forma adequada, de forma ordenada e simples, reduzindo o número de dados, se preciso, gerando um volume de dados que for necessário e significativo. Este processo pode envolver formatações, edições, redimensionamentos entre outros tipos de transformações. Estas etapas do pré-processamento e transformação, se realizadas corretamente, permitem a aplicação de técnicas e algoritmos na base de dados as quais tornam possível o KDD, diferentes algoritmos podem ou não exigir diferentes formatações e configurações da base, tais mudanças facilitam a próxima etapa, o processo de mineração dos dados (CARVALHO; SAMPAIO, 2000). A etapa de mineração dos dados é o processo o qual efetivamente se aplica os métodos e procedimentos, como a classificações, regressões, clusterizações, sumarizações gerando como resultado informações e modelos que serão avaliados para o objetivo. E como processo final após a obtenção dos resultados existe a análise e interpretação dos padrões gerados, etapa onde o conhecimento é, enfim, extraído ou descoberto, interpretar padrões minerados pode permitir retornar as etapas anteriores para mais iterações (BRACHMAN; ANAND, 1994).

Para realizar o PLN, algumas etapas adicionais são necessárias para que o documento possa ser representado e consequentemente processado.

2.2.2 *Bag of words*

Segundo Sebastiani (2002) os textos não podem ser interpretados diretamente por um classificador, consequentemente é necessário um procedimento de indexação que mapeia o texto em uma representação do seu conteúdo, esta etapa precisa ser aplicada nas fases de treinamento e validação. *Bag of words* ou na tradução, saco de palavras, é uma dessas formas de representação baseada no modelo atributo-valor mais comumente utilizada, em que o documento é representado como um vetor de palavras ocorridas no documento (MARTINS et al., 2003). Como mostra a Tabela 1 abaixo.

Tabela 1 – Exemplo de representação textual em modelo *Bag of words*

Frase	**Representação**
Levi likes to play games, Maria likes too.	{”Levi”:1,”likes”:2,”to”:1,”play”:1,”games”:1,”Maria”:1,”too”:1}

Fonte: Autoria Própria

Conforme discutido anteriormente é necessário realizar o pré-processamento nas bases de dados antes de utilizá-las para treinamento dos algoritmos, nesta fase existem diferentes processos e métodos que podem ser aplicados e que se suas aplicações variam muito de acordo com a capacidade de identificação dos problemas nas bases por parte do responsável por este processo. Para este trabalho ajustes como a tokenização (separar palavras em unidades), segmentação de sentenças, tratamento de caixa alta para caixa baixa, remoção de caracteres especiais, substituição de termos e caracteres que não facilitam o treinamento dos algoritmos por um *token* representativo são aplicados, além da remoção de palavra vazia, aplicação de stemização e lematização.

Remoção de palavras vazias, é a remoção de palavras muito frequentes, por exemplo, os artigos “a” e “o”, as preposições “de”, “para”, entre outros. Este processo é importante se essas palavras não tem relevância para o objetivo em que o corpus será utilizado porque segundo Korfhage (1997) as palavras vazias tem potencial para afetar diretamente a eficiência na obtenção de características do texto.

O processo de stemização consiste em uma normalização linguística em que formas derivadas de um termo podem ser reduzida a forma comum que recebe o nome de *stem* ou “raiz” (MARTINS et al., 2003). Por exemplo, as seguintes palavras, “andei”, “andaram”, “andarão”, podem ser transformadas na raiz[1] “and”, a raiz seria o elemento mórfico mais simples a que pode ser reduzida uma palavra. Este processo agrupa palavras com sentido similar, reduzindo a

[1]Em Processamento de Língua Natural, é comum o uso de uma definição relaxada do termo “raiz” de uma palavra ao invés de definição oficial utilizada na Linguística

dimensionalidade do problema. Erros de pré-processamento podem ocorrer gerando situações em que palavras com diferentes sentidos são reduzidas a uma mesma raiz.

Lucca (2002) diz que a lematização é o ato de representar as palavras através do infinitivo dos verbos e masculino singular dos substantivos e adjetivos. Já conforme Galisson e Coste (1983) consiste em encoontrar uma forma representativa de todas as formas que uma palavra ou palavras compostas pode tomar. Para o exemplo citado em *Stemming*, a palavra "anda", seria lematizada para "andar" assim como outras conjugações "andei", "andaram", "andarão". Estas técnicas que modificam a representação do documento são muitas vezes um passo anterior à classificação. Para esta nova etapa serão apresentadas características que podem ser analisadas por ferramentas, para então, fazer a classificação.

2.2.3 Características rasas e profundas

Segundo Scarton e Aluísio (2010), para a compreensão de um texto, alguns aspectos são importantes, como legibilidade (apresentação gráfica do texto) e a inteligibilidade (uso de palavras frequentes e estruturas sintáticas mais simples). Esses fatores afetam a complexabilidade da compreensão de textos. A microestrutura e macroestrutura de texto, coesão e coerência, são características que afetam a facilidade de entendimento do texto. O conceito do texto sensível ao leitor apresenta características que podem facilitar a compreensão como proximidade na anáfora, o uso de marcadores discursivos entre as orações, a preferência por definições explícitas ou a apresentação de informações complexas (LEFFA, 1996).

Existem muitas e diferentes fórmulas para avaliar a legibilidade de um texto, em sua maioria possuem focos diferentes, como textos fundamentais e escolares, textos médicos e de saúde, textos que focam em termos tecnológicos, manuais e entre outros. De acordo com Greenfield (2004) de maneira simplificada, as fórmulas de legibilidade são múltiplas equações de regressão em que a variável dependente, isto é, o valor que deseja-se descobrir representa a dificuldade de leitura prevista de um texto ou leitura, as variáveis independentes ou preditoras são duas ou mais características diretamente estimáveis do texto, como o número de letras por palavra e o número de palavras por frase. Pensando em características rasas, pode-se citar a *Flesch Reading Ease* e a *Flesch-Kincaid Grade Level*, estas fórmulas se baseiam no número de palavras das sentenças e no número de sílabas por palavra.

Segundo Scarton e Aluísio (2010), a Figura 3 demonstra exemplos de sentenças, em

a) *Sometimes you did not pick the right letter. You did not click on the letter 'd'.*
b) *Sometimes you did not pick the right letter. For example, you did not click on the letter 'd'.*
c) *Sometimes you did not pick the right letter. You did not, for example, click on the letter 'd'.*
d) *Sometimes you did not pick the right letter – you did not click on the letter 'd', for example.*
e) *You did not click on the letter 'd'. Sometimes you did not pick the right letter.*
f) *Sometimes you did not pick the right letter. For instance, you did not click on the letter 'd'.*

Figura 3 – Exemplo dos problemas do índice *Flesch*.
Fonte: (WILLIAMS, 2004).

inglês, com o índice Flesch, citando, em ordem de inteligibilidade, os itens (a) e (e) da Figura, seguidos por (b) e (c) como intermediários, e (f) e (d) em últimos, como piores. Nota-se que (a) e (e) não possuem marcadores de discurso, palavras que guiam a interpretação do texto e a relação das ideias, o leitor não sabe qual a relação entre elas, logo, características superficiais ou rasas captam de maneira falha a dificuldade de um texto, entretanto, são formulas úteis para aspectos superficiais como número de palavras, tamanho médio das palavras, sentenças e parágrafos.

A fórmula de legibilidade *Flesch reading Ease* conforme Flesch (1948) tem saída entre 0 e 100, sendo 100 uma leitura mais fácil e 0 uma leitura mais difícil. O parâmetro *Average sentence length* (ASL) representa o tamanho médio de sentenças (número de palavras dividido pelo número de sentenças), o *Average syllable per Word* (ASW) representa o número médio de sílabas por palavra, a fórmula proposta por Flesch, é da maneira:

$$RE(Readability\ Ease) = 206.835 - (1.015 x ASL) - (84.6 x ASW)$$

A *Flesch-Kincaid Grade Level* proposta por Kincaid et al. (1975) converte o índice *Flesch reading Ease* para uma série, sendo $(0.39 x ASL) + (11.8 x ASW) - 15.59$ uma série dos Estados Unidos e $248.835 - (1.015 x ASL) - (84.6 x ASW)$ uma adaptação para o português (MARTINS; MOREIRA, 2012). Os valores para esta fórmula também ficam entre 0 e 100, sendo 100 muito fácil e 0 muito difícil. Esta fórmula ainda resulta, além do *Score* ou pontuação, o *Grade Level* que é, por exemplo, a série escolar a qual é proximamente necessária para que o leitor consiga compreender o texto a ser lido, esta série porém é baseada no nível escolar Norte Americano.

Como é possível observar através das fórmulas citadas, com as características rasas de um texto é possível classificá-lo, entretanto, para o objetivo deste trabalho, a intenção é, também a utilização de características aprofundadas de um texto, as quais envolvem *word embeddings*, características sintáticas como número de orações subordinadas e coordenadas, tipos de orações, características semânticas como número de sinônimos, hiperônimos, sentidos, entre outros.

2.3 APRENDIZADO DE MÁQUINA

Saber o funcionamento de uma máquina permitiu com que fosse possível abstrair código de nível máquina para ficar mais próximo a linguagem natural humana. Assim sendo possível dar instruções para que os programas de computadores executem comandos, conjuntos de instruções. Entretanto, quando se fala em fazê-los aprender ainda é uma tarefa um pouco difícil e abstrata se comparado com o aprendizado humano, porém, esta tarefa vem, nas últimas décadas, sendo muito explorada e estudada ao ponto de muitas soluções serem desenvolvidas para problemas específicos como nas áreas financeiras, estatísticas, médicas, econômicas e diversas outras, ao passo que, tais algoritmos já são capazes de aprender mais rapidamente do que um ser humano.

Segundo Mitchell (1997) um programa de computador aprende com a experiência E em relação a alguma classe de tarefas T e medida de desempenho P, se seu desempenho nas tarefas T, medido por P, melhorar com a experiência E.

Figura 4 – Hierarquia de aprendizado.
Fonte: Adaptado de Faceli et al. (2011).

Segundo Faceli et al. (2011) o aprendizado de máquina é um processo indutivo no qual se busca obter generalizações de informações conhecidas, assim, gerando conhecimento e

novas informações, este processo pode ser supervisionado, não supervisionado ou por reforço.

Para ficar mais claro a estrutura da Figura 4 e cada uma das abordagens elas serão explicadas a seguir.

2.3.1 Aprendizado supervisionado

Santos (2005) define o aprendizado supervisionado como aprender a fazer predições considerando um conjunto de dados em que os rótulos são observados. Essa categoria de aprendizado de máquina recebe este nome devido a interação que o ser humano tem durante o processo de treinamento, atuando como um professor, instruindo o algoritmo durante o processo fornecendo, como exemplo, dados de entrada rotulados corretamente, isto é, com a resposta ou resultado esperado.

Para este tipo de aprendizado os algoritmos estão definidos em quais fazem regressão e classificação, a partir de um conjunto de treinamento de N exemplos de pares de entrada e saída $(x_1, y_1), (x_2, y_2), ...(x_n, y_n)$ onde cada y_j é gerado por uma função desconhecida $y = f(x)$, descobre-se a função h que aproxima a real função f (HOUSE; RADO, 2013). A classificação procura definir uma dada instância em uma classe com base em atributos, e portanto, pode apresentar erros. Existem diversos problemas de classificação que podem ser linearmente separáveis ou não. A regressão faz um processo parecido com a classificação, entretanto, procura estimar o valor de variáveis tendo como base dados numéricos. Alguns dos algoritmos mais conhecidos para aprendizado de máquina são, algoritmos de Árvores de decisão, Máquina de Vetor Suporte e algoritmo Bayes. Os algoritmos de aprendizado supervisionado selecionados para este trabalho serão apresentados em seus tópicos respectivos para a noção necessária do escopo do trabalho.

2.3.2 Aprendizado não supervisionado

O aprendizado não supervisionado é um método no qual, diferente do aprendizado supervisionado, não possui diretamente a dependência da interação do ser humano, pois tem como objetivo obter a relação entre variáveis do conjunto de dados, agrupando-os em caso de

similaridade e associando caso houver padrões entre os vários atributos. Segundo Bruce e Bruce (2019) o aprendizado não supervisionado pode ter objetivos diferentes, seja a criação de uma regra preditiva caso não houver uma resposta rotulada, ou reduzir a dimensão dos dados para um conjunto de variáveis mais fácil de gerir e utilizá-los como entrada em um modelo preditivo. Obtendo uma visão interna melhor do comportamento dos dados e das relações das variáveis.

2.3.3 Aprendizado por reforço

O aprendizado por reforço, é como uma mistura do aprendizado supervisionado e do não supervisionado, o papel do ser humano neste tipo é semelhante a de um analisador, avaliando a ação tomada entretanto sem indicar qual deveria ser a ação. Utiliza-se de estados e reforços para verificar as ações realizadas a procura de melhor rendimento. Segundo Sutton e Barto (2017) o objetivo deste tipo de aprendizado é extrapolar ou generalizar as respostas para atuar corretamente em situações não presentes no conjunto de treinamento, pode parecer semelhante ao aprendizado não supervisionado, mas o objetivo deste último tipo é maximizar uma "recompensa" em vez de tentar encontrar uma estrutura, relação oculta. O ponto deste aprendizado é o equilíbrio na decisão, no sentido de ações anteriores que deram certo e ações que ainda não foram tomadas, que podem consequentemente maximizar a recompensa.

2.4 ALGORITMOS DE CLASSIFICAÇÃO

Nesta seção serão apresentados algoritmos de classificação, que possuem como objetivo tentar determinar a classe de instâncias de um conjunto baseado nos atributos, para o aprendizado supervisionado é necessário que o os dados de treinamento estejam rotulados para que seja possível o treinamento.

2.4.1 Naive Bayes

Segundo Mitchell (1997) o classificador *Naive Bayes* é aplicado a tarefas em que cada instância é descrita por uma conjunção de valores de atributo, onde a função objetivo pode assumir qualquer valor de algum conjunto finito c. A diferença entre este método e os demais é que ele é formado sem pesquisa, pois baseia-se na contagem e frequência das combinações de dados de treinamento. O algoritmo Naive Bayes é baseado no teorema de Bayes, dado pela Equação 1. O teorema diz que a probabilidade do evento A ocorrer dado que o evento B ocorreu é calculada da seguinte forma. Primeiro, é obtido o produto entre a probabilidade do evento B ocorrer dado que A ocorreu e a probabilidade do evento A ocorrer. Na sequência, esse produto é dividido pela probabilidade de B.

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)} \tag{1}$$

No contexto de Aprendizado de Máquina, deseja-se calcular a probabilidade da classe c de uma determinada instância dados os seus valores de atributos $a_1, a_2, ..., a_n$, conforme descrito na Equação 2.

$$p(c|a_1, a_2, ..., a_n) = \frac{p(a_1, a_2, ..., a_n|c)P(c)}{p(a_1, a_2, ..., a_n)} \tag{2}$$

Mais precisamente, deseja-se saber qual é a classe mais provável, o que pode ser feito utilizando-se a Equação 3.

$$argmax_c \frac{p(a_1, a_2, ..., a_n|c)P(c)}{p(a_1, a_2, ..., a_n)} \tag{3}$$

Considerando-se que é a possível classe que varia e não os atributos da instância, o denominador é constante e pode ser eliminado, resultando na Equação 4.

$$argmax_c p(a_1, a_2, ..., a_n|c)P(c) \tag{4}$$

O algoritmo também conta com a aplicação da hipótese ingênua, descrita na Equação 5, que considera que os atributos são condicionalmente independentes entre si, o que nem sempre é válido, mas fornece uma estimativa da probabilidade real. O cálculo então pode ser simplificado como na Equação 6.

$$p(A, B) = p(A)p(B) \tag{5}$$

$$argmax_c p(c) \prod_{i=1}^{n} p(a_i|c) \tag{6}$$

2.4.2 Máquina de Vetor Suporte

Segundo Goodfellow et al. (2016) as Support Vector Machines (SVMs) foram inicialmente criadas para resolver problemas linearmente separáveis de classificação binária utilizando da função $wx + b$, predizendo que a classe é positiva caso o resultado desta função seja positivo, assim também a classe é negativa caso o resultado seja negativo. O algoritmo busca determinar o hiperplano separador ótimo entre instâncias de ambas as classes. Para isso, são utilizadas instâncias de apoio, próximas a fronteira de decisão. O processo é ilustro na Figura 5, onde em (a) são mostrados vários hiperplanos separadores e (b) indica o hiperplano separador ótimo usando as três instâncias circuladas como apoio.

A busca pelo hiperplano separador ótimo pode ser representada como um problema de otimização quadrática. O algoritmo pode ser estendido para problemas não-lineares com o uso de funções *kernel*. Esse tipo de função aumenta, de forma não linear, a dimensionalidade dos dados. Uma função *kernel* pode mapear as instâncias de tal maneira que as classes se tornem linearmente separáveis após a sua aplicação. O processo é ilustrado na Figura 6. Adicionalmente, problemas de classificação não binários (multi classes) podem ser decompostos em problemas binários com a indução de diferentes SVMs, usando por exemplo matrizes de código.

2.4.3 Árvores de Decisão

Segundo Mitchell (1997), Árvores de Decisão é um dos métodos mais largamente utilizados para o processo indutivo. Esse método classifica as instâncias organizando-as em forma de árvore, do nó raiz até os nós folhas, cada nó especifica um teste de cada atributo da instância, cada ramo de um nó corresponde a um possível valor para o atributo, e a árvore é construída com base nas instâncias. As instâncias são testadas a partir do nó raiz, após criar um

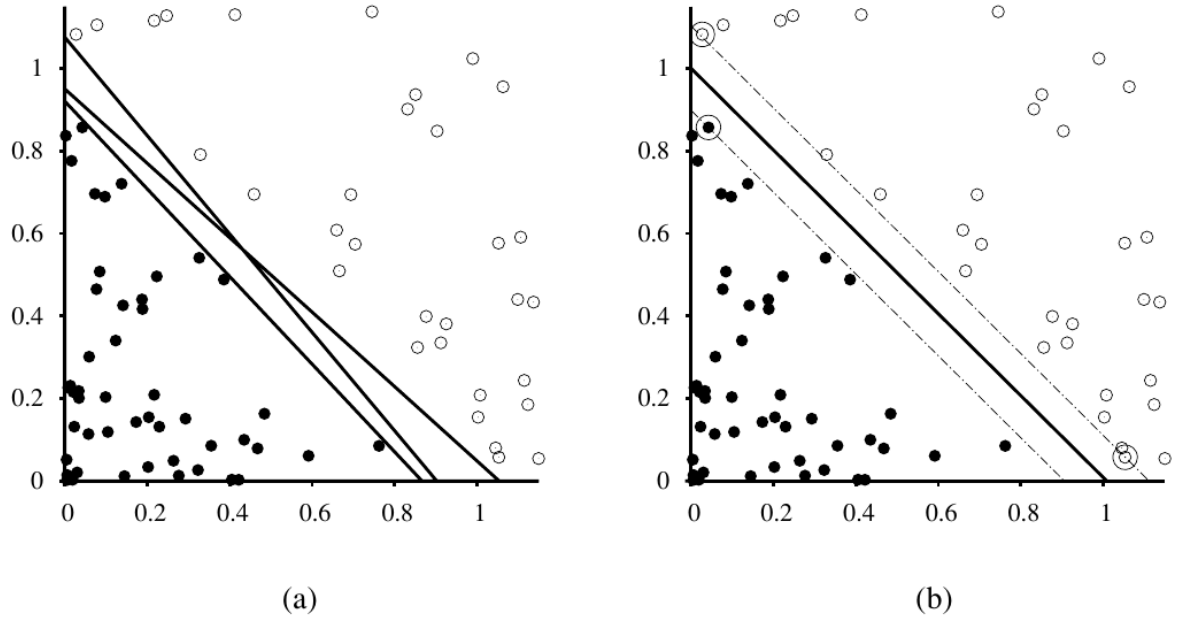

Figura 5 – Hiper Plano separador ótimo.

Fonte: (RUSSELL; NORVIG, 2013)

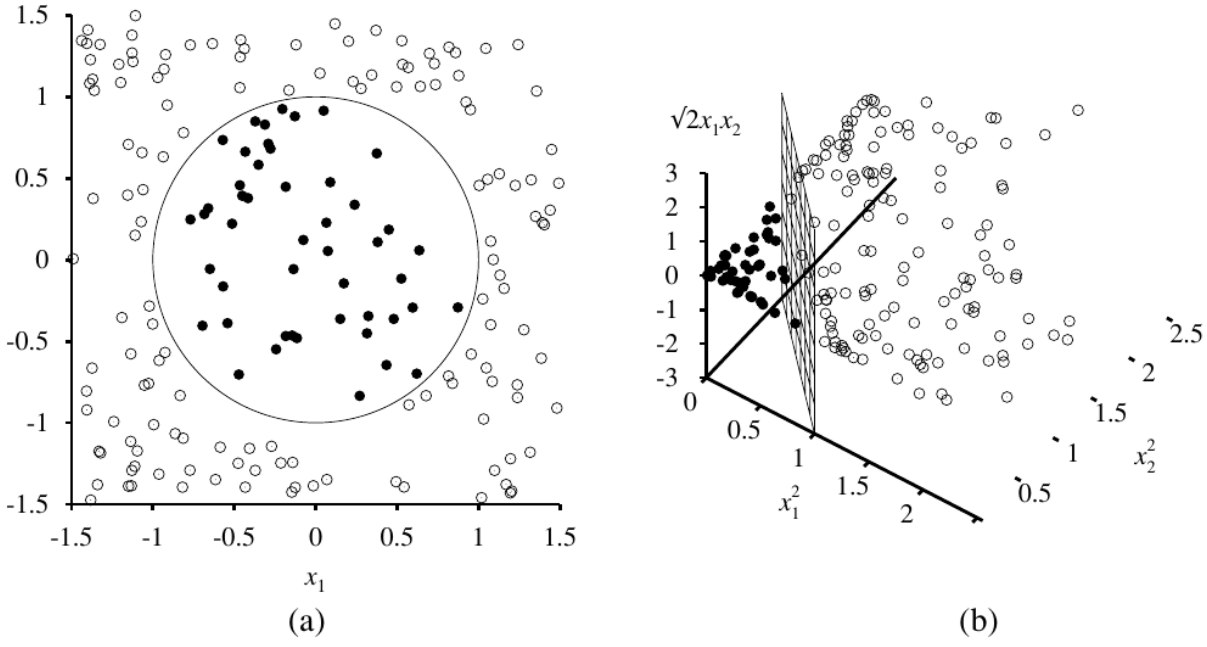

Figura 6 – Mapeamento para um problemas não linearmente separável usando funções *kernel*.

Fonte: (RUSSELL; NORVIG, 2013)

ramo para o valor do atributo, o processo se repete para este ramo criado, sendo considerado como um novo nó raiz na árvore. A Figura 7 apresenta a estrutura de uma árvore de decisão

para verificar se uma manhã de sábado está boa para jogar tênis.

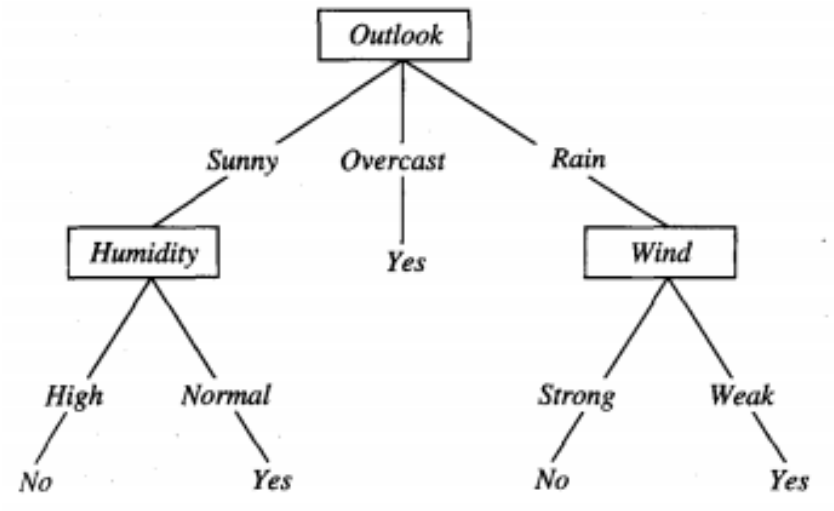

Figura 7 – Conceito de uma árvore de decisão.
Fonte: (MITCHELL, 1997)

Uma medida muito utilizada para auxiliar a determinar e medir a pureza e impureza de um subconjunto é a entropia, que procura dizer o quão diferentes ou iguais elementos são entre si, com valores entre 0 e 1. As Equações 7 e 8 apresenta a fórmula associada à medida de entropia, sendo V uma variável aleatória com v_k valores, cada uma com percentual $p(v_k)$ em relação ao subconjunto.

$$e(V) = \sum_k p(v_k) log_2 \frac{1}{p(v_k)} \tag{7}$$

$$e(V) = -\sum_k p(v_k) log_2 p(v_k) \tag{8}$$

Com base na entropia é possível aplicar outras medidas para auxiliar no processo de seleção do melhor atributo para ser criado um nó em dado nível da árvore. Uma das medidas é o ganho de informação, dado pela Equação 9, em que S representa o conjunto, A representa o atributo, v os valores de um conjunto, um somatório aplicado a valores de um conjunto, analisando cada valor iterativamente, a proporção da frequência de v sobre todo o conjunto multiplicado pela entropia de subconjunto. Permitindo criar índices e assim dizer qual melhor atributo. A entropia avalia a impureza de um *subdataset* enquanto o ganho de informação avalia a impureza dentro de uma divisão.

$$Gain(S,A) = Entropia(S) - \sum_{v=Valores(A)} \frac{|S_v|}{|S|} Entropia(S_v) \tag{9}$$

2.5 TRABALHOS RELACIONADOS

Nesta seção serão apresentados e descritos os principais trabalhos relacionados ao tema de legibilidade e inteligibilidade de textos e PLN, que inspiraram o desenvolvimento deste trabalho.

2.5.1 Coh-Metrix

A ferramenta computacional *Coh-Metrix*, conforme os autores Graesser et al. (2004), analisa textos com aproximadamente 200 medidas de coesão, linguagem e legibilidade, utilizando de módulos lexicais, analisadores sintáticos e análise semântica latente, além de outros componentes que são muito utilizados na área de processamento linguístico. O funcionamento da ferramenta é relativamente simples, recebe como parâmetro de entrada um texto qualquer em língua inglesa e retorna ao usuário métricas e medidas as quais podem ser salvas em arquivos. O interessante desta ferramenta é que, além das fórmulas tradicionais de legibilidade e dificuldade de compressão de textos ainda são considerados aspectos mais sensíveis e aprofundados tais como características linguísticas e de discursos. Esta ferramenta é considerada como estado da arte por pesquisadores da área e amplamente utilizada como base para muitos outros trabalhos e pesquisas como, por exemplo, *Coh-Metrix-Port* de Scarton e Aluísio (2010) e *Pylinguistics* de Castilhos (2017) os próximos dois apresentados a seguir. A análise sobre esta ferramenta é de grande auxílio para a validação deste trabalho que visa obter métricas de texto em língua inglesa.

2.5.2 *Coh-Metrix-Port*

Coh-Metrix-Port é uma ferramenta desenvolvida por Scarton e Aluísio (2010) que adapta métricas da ferramenta Coh-Metrix desenvolvida por Graesser et al. (2004) para o português brasileiro, em seu trabalho, as autoras apresentam avaliações de textos jornalísticos e

também adaptação destes textos para crianças, desta forma, ressaltando as principais diferenças de compreensão e legibilidade entre textos simples e complexos através da utilização de classificadores binários. Foram obtidos, através do classificador SVM, resultados com precisão de aproximadamente 97% desta forma separando com bom desempenho textos dedicados a adultos e crianças que foram o objetivo da pesquisa. Observando a metodologia aplicada e os resultados obtidos pelas autoras conclui-se que a utilização de classificadores de aprendizado de máquina treinados com características obtidos de textos são viáveis para alcançar o objetivo de predizer, com bom desempenho, classes de dificuldade de, por exemplo, legibilidade de textos e leituras, abrindo espaço para pesquisas na área como este trabalhos.

2.5.3 *Pylinguistics*

No trabalho de Castilhos (2017) foi desenvolvida uma biblioteca de código aberto para a avaliação de inteligibilidade de textos escritos em Português. O autor ainda apresenta avaliações comparativas obtendo resultados relevantes. Para validar uma das aplicações da ferramenta desenvolvida o autor também apresenta uma análise prática da inteligibilidade do jornalismo científico brasileiro. Foram selecionados, de maneira aleatória, 20 artigos do corpus da Fapesp [2], posteriormente processados na ferramenta desenvolvida *Pylinguistics*, os resultados das métricas obtidas foram comparadas com os resultados obtidos pelo *Coh-Metrix-Port*, a comparação avaliativa deram indicativos suficientes para a validação da ferramenta embora as formas de validação não terem sido abrangidas no trabalho. Com base no trabalho de Castilhos (2017), se torna mais simples desenvolver a proposta deste trabalho.

[2] Site: https://www.fapesp.br/

3 MATERIAIS E MÉTODOS

Embasado nos conhecimentos apresentados nas seções anteriores, serão apresentados os materiais e métodos utilizados para o desenvolvimento deste trabalho e realização dos objetivos propostos.

3.1 MATERIAIS

Nesta seção serão apresentados todos os materiais necessários e que serão utilizados para o desenvolvimento deste trabalho.

3.1.1 Waikato Environment for Knowledge Analysis

Para o treinamento e seleção dos algoritmos de classificação utilizou-se o WEKA[1] versão 3.8.5, o qual é composto por um interface gráfica (Figura 8) e uma biblioteca Java open source para mineração de dados.

Essa ferramenta foi selecionada para ser o ambiente em que os algoritmos são treinados devido a sua grande utilização no contexto de mineração de dados e *machine learning* globalmente. Por ser muito utilizado, o Weka possui uma comunidade que desenvolve inúmeros algoritmos e ferramentas que podem ser instalados facilmente nele e serem utilizadas. A interface possui diversas opções, a que será utilizada neste trabalho é a "Explorer" onde, ao acessar, uma nova tela é aberta fornecendo novas opções como *Preprocess*, *Classify*, *Cluster*, *Associate*, *Select Attributes* e *Visualize*, portanto, disponibilizando várias técnicas no ambiente

[1] Site: https://www.cs.waikato.ac.nz/ml/weka/

gráfico. Utiliza como padrão arquivos com extensão ".arff" (ARFF), porém permitindo outros diversos.

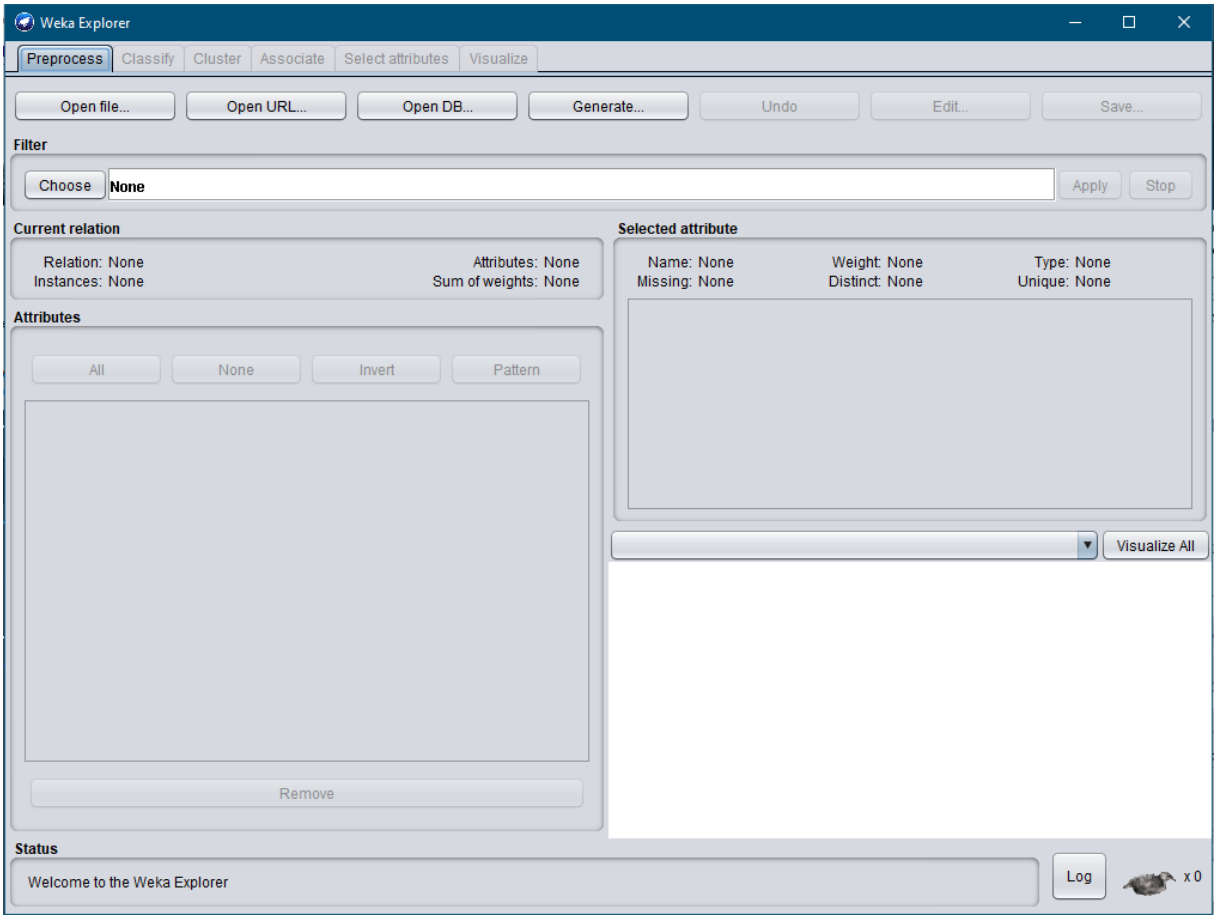

Figura 8 – Interface gráfica do WEKA.

Fonte: Autoria própria

.

3.1.2 Python

Python[2] é uma linguagem de programação interpretada, com tipagem forte e dinâmica. Essa linguagem de alto nível tem crescido muito no mercado e comunidades desde 2012 segundo mostra a Figura 9. Despertou um interesse maior ainda nos programadores ao lançar sua versão 3.0, e tem sido largamente utilizada em áreas como aprendizado de máquina e análise

[2]Site: https://www.python.org/

de dados.

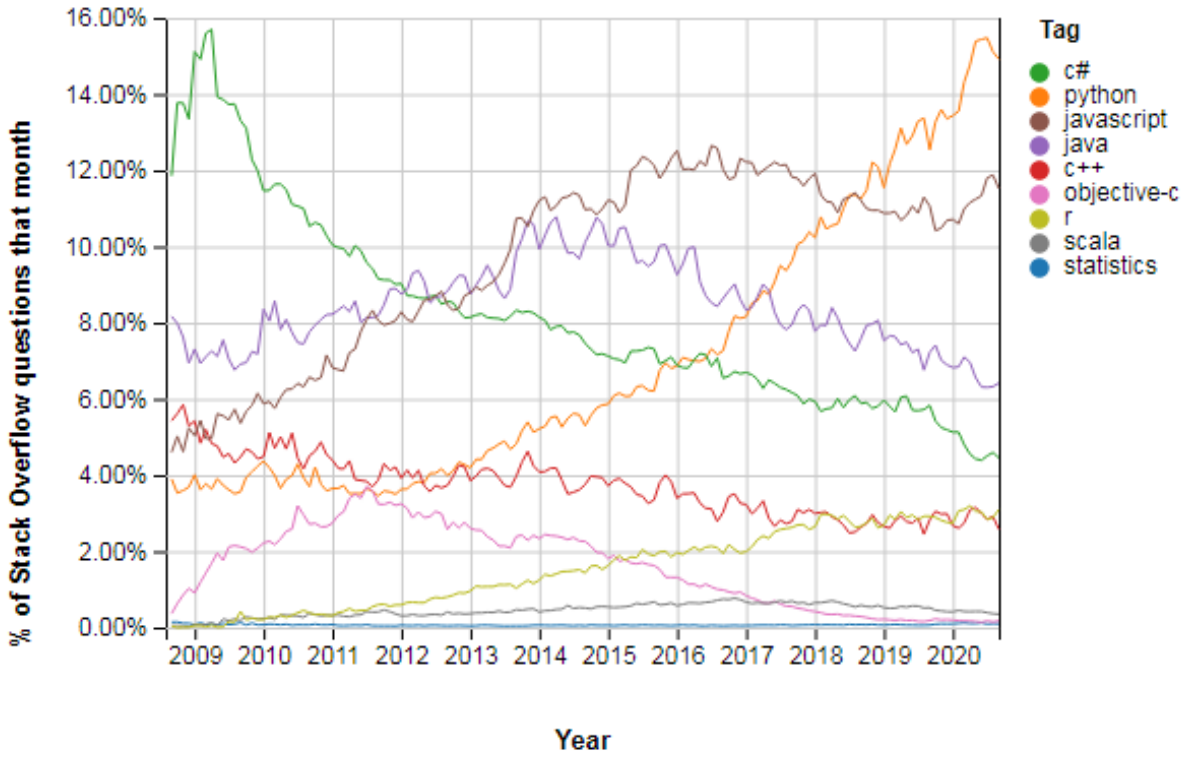

Figura 9 – Porcentagem de questões sobre as linguagens no Stack Overflow

Fonte: Adaptado de Stack Overflow (2021)

.

Conforme a Python Software Fundation (2021) a linguagem python possui como características a flexibilidade, podendo se comunicar com outras linguagens, acessibilidade por ser código aberto e possuir uma comunidade grande e aberta em auxiliar e desenvolver novas soluções, simples pois apresenta uma grande facilidade de aprendizado, tanto que sua popularidade tem aumentado entre estudantes. Python apresenta diversas características que a diferencia de muitas outras consideradas líderes de mercado. Mas o principal motivo da escolha desta linguagem para o desenvolvimento deste trabalho é que muitas soluções, ferramentas e tecnologias para aprendizado de máquina já estão feitas e portanto, ajuda na solução de problemas intermediários.

3.1.3 Bibliotecas

A linguagem Python já possui diversos pacotes e bibliotecas de código aberto disponíveis na internet e em diversas comunidades. Utilizando desse aspecto positivo da linguagem serão apresentados a seguir os principais pacotes utilizados para o desenvolvimento deste trabalho.

3.1.4 NTLK

O projeto NLTK[3] é liderado por Steven Bird e Lilling Tan (BIRD, 2021), porém cada pacote é mantido individualmente por diferentes colaboradores ao redor do mundo. É um programa de código aberto e também uma plataforma para a construção de programas em Python para trabalhar com dados de linguagem humana como, o PLN, fornecendo interfaces com recursos lexicais, pacotes de bibliotecas de processamento de texto para classificação, tokenização, lematização, e diversas outras funções.

3.1.5 Pandas

Pandas[4] é um pacote Python que fornece suporte e ferramentas para o trabalho com dados estruturados de maneira simplificada. É também uma ferramenta de análise de dados de código aberto amplamente utilizada e disponível em muitos idiomas. Um conjunto de dados pode ser armazenado em uma estrutura de dados do Pandas, sendo as duas principais a *Series* e *Dataframe* (*PYDATA DEVELOPMENT TEAM*, 2021). Através do Pandas pode-se, assim como utilizado neste trabalho, remodelar, unir e separar conjuntos de dados, rotular eixos, carregar e ler dados de diferentes tipos e extensões de arquivos como valores separados por vírgulas, também conhecido como CSV, e arquivos do programa Excel da Microsoft. O pacote também possui um repositório[5] disponível na Internet na plataforma Github com seu conteúdo

[3] Site: https://www.nltk.org/
[4] Site: https://pandas.pydata.org/
[5] Site: https://github.com/pandas-dev/pandas

disponível para acesso, além de muitos guias e tutoriais de utilização.

3.1.6 Readability

O pacote Readability[6] para a linguagem Python, conforme a definição do autor Andreas van Cranenburgh em seu repositório[7] disponível do Github[8] é uma implementação de métricas, regressões lineares de legibilidade tradicionais utilizando como base características rasas como número de palavras, sílabas e frases. Suporta idiomas como a língua Inglesa, Alemã e Holandesa, porém as fórmulas foram desenvolvidas para a língua inglesa. A biblioteca recebendo como parâmetro um texto qualquer previamente aplicado no processo de tokenização apresenta métricas de legibilidade, como por exemplo, o índice Flesch Reading Ease, Kincaid Grade Level, DaleChall e entre outros. Extrai do texto características como caracteres por palavra, sílabas por palavra e palavras por sentença, informações sobre o uso de palavras como verbos auxiliares, conjunções e pronomes, além de quantidade de preposições e subordinações de início de frase (CRANENBURGH, 2021).

3.1.7 Cleantext

A biblioteca *Cleantext*[9] é um pacote de código aberto para a linguagem Python utilizada para limpar dados de textos brutos. Possui dois métodos principais *clean* e *clean_words*, o primeiro é utilizado para limpar todo o texto bruto e retornar o texto limpo, o segundo é bem semelhante, entretanto, retorna uma lista de palavras limpas. Para o processo de limpeza algumas operações podem ser aplicadas, entre elas, remoção de espaços em branco extras, conversão do texto para caixa baixa, remoção de dígitos, pontuações, números, e-mail e palavras de parada (GUDIWADA, 2021).

[6] Site: https://pypi.org/project/readability/
[7] Site: https://github.com/andreasvc/readability/
[8] Plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git.
[9] Site: https://pypi.org/project/cleantext/

3.1.8 Equipamentos

Para o treinamento dos algoritmos de classificação apresentados serão utilizados os seguintes equipamentos de *hardware*:

Tabela 2 – Especificações de hardware do computador a ser utilizado no treinamento

Especificações	**Notebook**
Processador	AMD Ryzen 7 1700
Memória RAM	16 GB
Placa gráfica	Nvidia GeForce GTX 970
Disco rígido	SSD 240GB
Sistema Operacional	Windows 10 Pro 64 bits 19042

Fonte: Autoria Própria

Para o escopo deste trabalho a máquina de uso pessoal já apresenta os requisitos de hardware mínimos necessários para o desenvolvimento das etapas citadas no fluxograma.

3.1.9 Wikipédia e Simple Wikipedia

A Wikipédia[10], segundo a própria definição disponível no site oficial, é um projeto de enciclopédia multilíngue de licença livre, escrita de maneira colaborativa e sem fins lucrativos. Possui um acervo de definições, palavras em grande escala, cada uma destas definições possuem bastante informação relevante sobre o escopo do assunto, como contexto histórico, origem, entre outros aspectos. Segundo informações oficiais do site, em Novembro de 2020 a Wikipédia em português possuía 1.047.992 artigos válidos, páginas de domínio principal. Por este motivo foi decidido por utilizar um *dataset* relacionado a esta enciclopédia devido a grande quantidade de dados e informações detalhadas de maneira completa que podem ser perfeitamente utilizadas no escopo do trabalho.

A Simple English Wikipédia[11] é uma versão da Wikipédia escrita apenas em inglês simplificado, em Novembro de 2020 possuía 170 mil artigos, sendo a 50ª maior Wikipédia. Seu diferencial é utilizar sentenças curtas, palavras e gramáticas mais simples do que a Wikipédia

[10] Site: https://www.wikipedia.org/
[11] Site: https://simple.wikipedia.org/

completa. Seu *dataset* também será utilizado no objetivo do trabalho para treinar os algoritmos em representar a classe de textos com conteúdo mais fácil de compreender, por isso a escolha desse *dataset*.

3.2 MÉTODOS

A Figura 10 apresenta o fluxograma e sequência das atividades necessárias para a realização do trabalho.

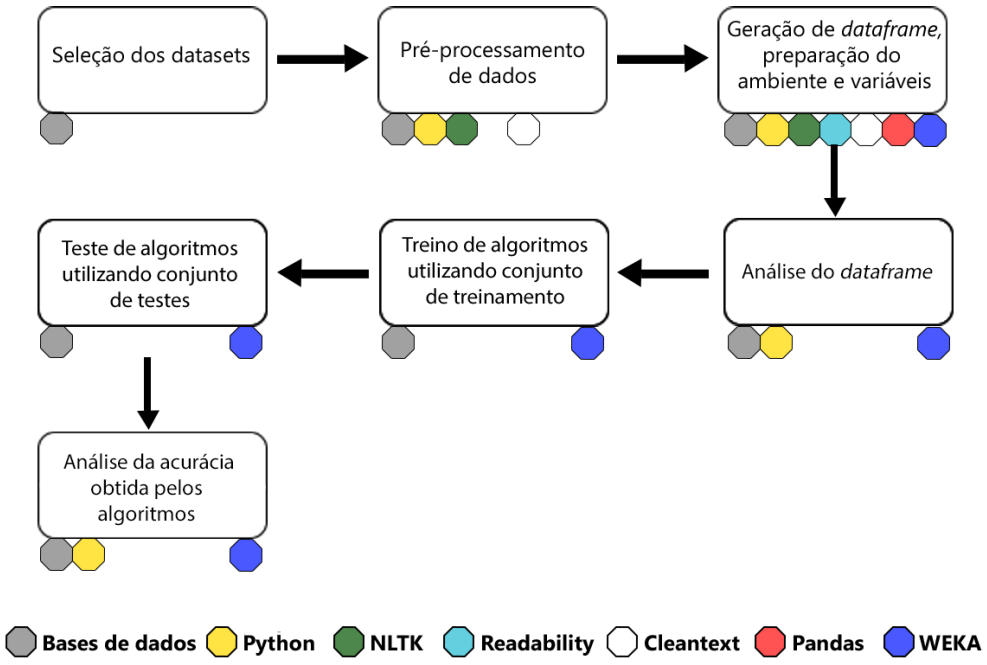

Figura 10 – Diagrama de fluxo de trabalho.

Fonte: Autoria própria.

Para início a seleção dos *datasets* em inglês públicos e gratuitos foram necessárias pesquisas na Internet. Foram necessários *datasets* que possuíssem definições em diferentes graus de legibilidade para auxiliar o aprendizado dos algoritmos de classificação. Após muitas pesquisas, comparações e seguindo a indicação do orientador deste trabalho, chegou-se a conclusão da utilização dos *datasets Wikipedia* e *Simple* Wikipédia, a diferença entre eles permite com êxito aos algoritmos diferenciar uma definição complexa de uma simples. O primeiro possui definições e conteúdos sobre diversos temas, palavras, de maneira formal e aprofundada, enquanto o segundo possui quase o mesmo conteúdo, porém, define a mesma

coisa de maneira mais simples e sem termos que venham a complicar o entendimento e compreensão por parte de leigos em relação ao assunto pesquisado. Após definidos os *datasets* é necessário realizar o tratamento e pré processamento dos dados, com o objetivo de modificação para aprimoramento dos resultados do treinamento dos algoritmos. Neste sentido, as bibliotecas citadas serão úteis para a realização do PLN.

Os *datasets* utilizados são partes menores dos *datasets Wikipedia*[12] e *Simple Wikipedia*[13]. Possuem quantidades diferentes de artigos, sendo 19.555 presentes no *dataset Wikipedia*, e 49.754 no *dataset Simple Wikipedia*. Posteriormente serão descritos processos utilizados para igualar a quantidade de dados em ambos *datasets*. O *dataset Wikipedia* utilizado para este trabalho apresenta originalmente um formato e estrutura com diversas *tags* de Linguagem de Marcação de Hipertexto, o que dificulta que este seja aplicado ou utilizado diretamente no processo de extração de características textuais e obtenção das métricas, além dos artigos estarem contidos em um único arquivo, sendo necessário a separação de cada artigo em seu respectivo arquivo nomeado. Já o *dataset Simple Wikipedia* foi encontrado em formato de texto plano, ou seja, sem marcadores de linguagens de hipertexto, porém também todo reunida em um único arquivo de texto. Analisando estas condições foram desenvolvidos dois *scripts* que respectivamente modelam e modificam ambos em seus devidos formatos, com os arquivos separados e nomeados com os títulos dos artigos. A Figura 11 apresenta o formato dos arquivos dos *datasets* depois de serem aplicados nos *scripts*.

```
A, or a, is the first letter and the first vowel letter of the modern English
alphabet and the ISO basic Latin alphabet. Its name in English is "a" (pronounced
), plural "aes". It is similar in shape to the Ancient Greek letter alpha, from
which it derives. The uppercase version consists of the two slanting sides of a
triangle, crossed in the middle by a horizontal bar. The lowercase version can be
written in two forms: the double-storey a and single-storey ɑ. The latter is
commonly used in handwriting and fonts based on it, especially fonts intended to be
read by children, and is also found in italic type.
```

Figura 11 – Artigo de exemplo do *dataset* depois do processamento do texto para texto plano

Fonte: Autoria Própria

Da maneira como estão apresentados, já é quase possível aplicar diretamente as bibliotecas de extração de métricas nos *datasets*. Porém, alguns processos de PLN extras são aplicados antes da ativação dessas bibliotecas. São aplicados os métodos *remove_stopwords* para retirar as palavras vazias das cadeias de caracteres, *tokenization* para separar as palavras em unidades e o *clean* do pacote *Cleantext* para modificar, retirar e substituir todos os conteúdos

[12] Site: http://wikipedia.c3sl.ufpr.br/enwiki/20210620/enwiki-20210620-pages-meta-current1.xml-p1p41242.bz2

[13] Repositório: https://github.com/LGDoor/Dump-of-Simple-English-Wiki

presentes nos textos que não são úteis para a classificação, por exemplo, são aplicadas a remoção de e-mail, número de celular, moeda corrente, Localizadores Uniformes de Recursos (URL) e substituindo-os por *tokens* representativas, por exemplo, "<URL>", além da mudança nas palavras de caixa alta para caixa baixa. Um exemplo de texto de saída processado pelos métodos citados é apresentado na Figura 12.

```
("['a', 'zoological', 'garden', ',', 'zoological', 'park', ',', 'zoo', "
 "'place', 'different', 'species', '(', 'types', ')', 'animals', 'kept', "
 "'held', 'bad', 'conditions', '.', 'they', 'kept', 'small', 'cages', ',', "
 "'bored', 'sick', '.']")
```

Figura 12 – Exemplo de artigo processado
Fonte: Autoria Própria

Para utilizar algoritmos para extração de características e métricas textuais também foi desenvolvido um *script* Python que utiliza dos textos processados e os enviam como parâmetro para o método *readability*, este obtém as diversas métricas e características textuais e retorna um dicionário.

Após a etapa de pré processamento foi necessário a geração do *dataset* de treinamento, este foi construído em torno das métricas obtidas, nome dos artigos, resultados de fórmulas de legibilidade e nome do *dataset* de origem do artigo. A estrutura do *dataset* é apresentado conforme a Figura 13. O *dataset* está em formato de arquivo *Comma-separated values* (CSV), separando valores com vírgulas. A primeira linha deste arquivo de saída contém o nome dos atributos utilizados, a partir da segunda linha estão dispostos nas colunas os valores para cada atributo listado, sendo o valor da última coluna a classe, ou seja, o nome de qual *dataset* a instância originalmente pertence, desta forma, cada linha representa uma instância que será utilizada para treinamento nos algoritmos de classificação. Também foi necessário configurar o ambiente Weka e as variáveis dos algoritmos de classificação J48, LibSVM, e Naive Bayes, porém somente o parâmetro chamado número mínimo de instâncias por folha do algoritmo J48 é alterado, todos os demais mantém o padrão.

Com este aspecto devidamente organizado, é possível realizar o treinamento e teste dos algoritmos de classificação utilizando dos *dataset* no ambiente Weka. Com os processos de treinamentos e testes finalizados, foi possível obter as estatísticas e acurácias de cada um dos algoritmos e compará-los. A utilização do programa Weka foi escolhida devido a ser uma ferramenta completa que já possui implementação para os algoritmos escolhidos, sendo necessário somente alterar o parâmetro citado. Na etapa final com os algoritmos devidamente treinados no objetivo de classificação, é possível fazer a extração do modelo para utilização,

```
characters_per_word,syll_per_word,words_per_sentence,...,classes
0.17639876715210306,0.2519110021059938,0.030066200809121,...,enwikipedia
0.15367011865406388,0.220557500119716,0.04067058967757754,...,enwikipedia
0.15064364896907834,0.2194588771288639,0.13816354051734708,...,enwikipedia
0.11844273551222192,0.21442389923529756,0.0003677822728944465,...,enwikipedia
0.10285816505008746,0.1649994471977734,0.0008581586367537084,...,enwikipedia
0.0998515028931333,0.1566585292983958,0.00150177761431899,0.0,...,enwikipedia
0.15167695118626812,0.24055329676081408,0.12299252176045114,...,enwikipedia
0.15269933009326153,0.21125614012794497,0.0003984307956356503,...,enwikipedia
0.1568673018824076,0.214631473193899,0.0008888071594949123,...,enwikipedia
0.1808331396699977,0.2511644899077541,0.000490376363859262,...,simplewiki
0.17865442714385316,0.2580144305416019,0.000490376363859262,...,simplewiki
0.17647571461770856,0.24431454927390625,0.000490376363859262,...,simplewiki
0.16558215198698584,0.22376472737236272,0.000490376363859262,...,simplewiki
0.16478989288656964,0.22251928362075404,0.0003677822728944465,...,simplewiki
0.1851905647222868,0.2580144305416019,0.000490376363859262,...,simplewiki
```

Figura 13 – Exemplo de formato do *dataset* de treinamento

Fonte: Autoria Própria

entretanto, esta etapa fica aberta para pesquisas e trabalhs futuros. Como classificador para os usuários tem como entrada outros *datasets* no mesmo formato apresentado anteriormente, sendo necessário passar pelas mesmas etapas citadas pelo fluxograma da Figura 10, e apresentará como saída através do programa Weka, em forma de matriz de confusão, a classe prevista das instâncias, ou seja, o nível de legibilidade do texto presente no *dataset*, além da acurácia do algoritmo selecionado.

4 RESULTADOS

Nesta seção serão apresentados os resultados das etapas anteriores e do desenvolvimento deste trabalho.

4.1 ANÁLISE DO DATASET

Com o *dataset* gerado resultante dos *scripts* pode ser feita a análise sobre as instâncias e valores. Através de gráficos representativos do *dataset* é possível observar algumas características das instâncias de diferentes classes. A Figura 14 apresenta graficamente alguns dos atributos do *dataset* com valores normalizados entre 0 e 1, listados como entre os mais importantes, segundo algoritmos de seleção de atributos do ambiente WEKA para a classificação das instâncias. A quantidade de instâncias presentes no *dataset* apresentam-se diferentes para cada classe, sendo 49.750 instâncias presentes para a classe *Simple Wikipedia* e 19.554 instâncias presentes para a classe *Wikipedia*. Com o objetivo de aprimorar os resultados do processo de classificação por parte dos algoritmos essas quantidades foram redimensionadas para que ambos *datasets* tenham o mesmo número de instâncias. Após o redimensionamento as *datasets* ficaram com 19.544 instâncias para cada classe.

No *dataset Simple Wikipedia* quase toda a frequência normalizada de palavras complexas se concentra próxima ao valor zero, assim não presentes tantas palavras complexas que é seu objetivo, enquanto o *dataset Wikipedia* já tem as frequências mais distribuídas, pois o objetivo da base é conter definições mais complexas dos artigos presentes. A frequência normalizada de palavras longas, apresenta-se de maneira bem similar ao comportamento da frequência de palavras complexas. Palavras muito longas podem atrapalhar a leitura do usuário sobre um texto qualquer, ciente deste aspecto o *dataset Simple Wikipedia* foi construído visando não utilizar palavras longas e por este motivo a frequência fica bem próxima do valor zero, enquanto para o *dataset Wikipedia* esta frequência normalizada é mais distribuída. O índice

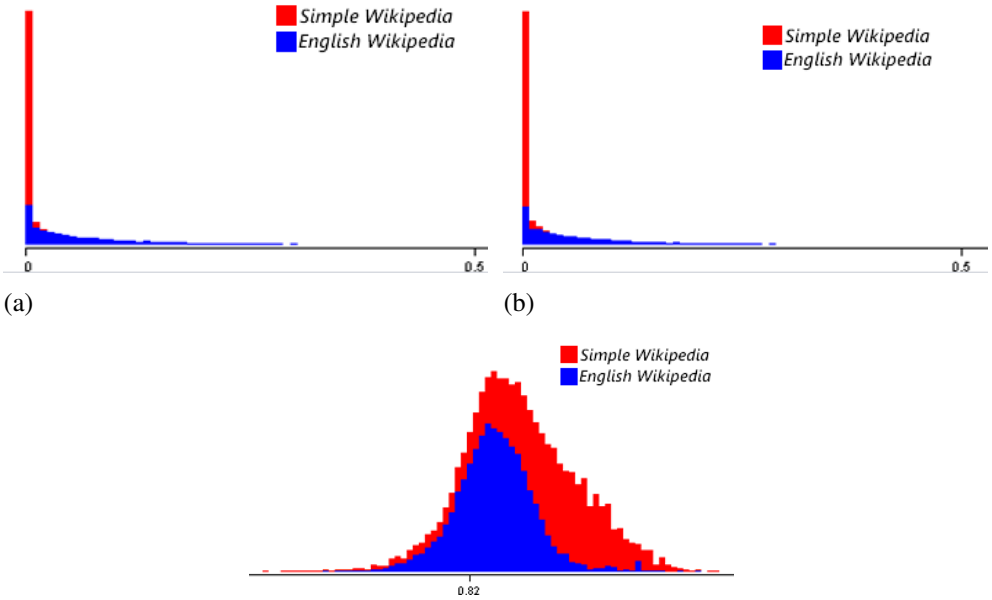

Figura 14 – Gráfico (a) frequência de palavras complexas, Gráfico (b) frequência de palavras longas, Gráfico (c) índice de Miyazaki.

Fonte: Autoria Própria

Miyazaki Greenfield (2004) é semelhante aos índices Kincaid e Flesch. O valor normalizado desse índice, para os artigos do *dataset Simple Wikipedia* apresentou maior frequência de valores próximos a um, ou seja, leituras mais fáceis do que o *dataset Wikipedia*.

4.2 APRENDIZADO DE MÁQUINA

Os algoritmos de aprendizado de máquina são aplicados no *dataset* através do ambiente WEKA na versão 3.8.5. Para este trabalho são aplicados os algoritmos Máquina de Vetor Suporte (*LibSVM*), Naive Bayes e árvore de decisão (J48). A característica do problema proposto é uma classificação binária e os algoritmos de aprendizado de máquina selecionados lidam bem sem exigir tempos extremos de processamento ou *hardwares* robustos.

A Tabela 3 apresenta os resultados do treinamento dos algoritmos de classificação.

Todos os algoritmos utilizaram parâmetros padrões do programa, exceto pela árvore de decisão J48, a qual o parâmetro chamado número mínimo de instâncias por folha foi

Tabela 3 – Acurácias obtidas pelos algoritmos de treinamento.

Algoritmo	Categoria	Acurácia	Método de reamostragem
J48	Árvore de decisão	94,17%	Divisão de porcentagem
J48	Árvore de decisão	90,12%	Validação cruzada
LibSVM	Máquina de vetor suporte	87,67%	Divisão de porcentagem
LibSVM	Máquina de vetor suporte	87,33%	Validação cruzada
NaiveBayes	*Naive Bayes*	85,63%	Divisão de porcentagem
NaiveBayes	*Naive Bayes*	85,34%	Validação cruzada

Fonte: Autoria Própria

configurado com valor igual a 800, desta forma, realiza-se uma pré-poda com o objetivo de gerar uma árvore menor. Através do programa é possível acessar a estrutura visual da árvore formada pelo algoritmo de Árvore de decisão, com base nisso a interpretação sobre as instâncias classificadas se torna um processo mais visível, a estrutura resultante da construção da árvore para as instâncias do algoritmo é demonstrada pela Figura 15.

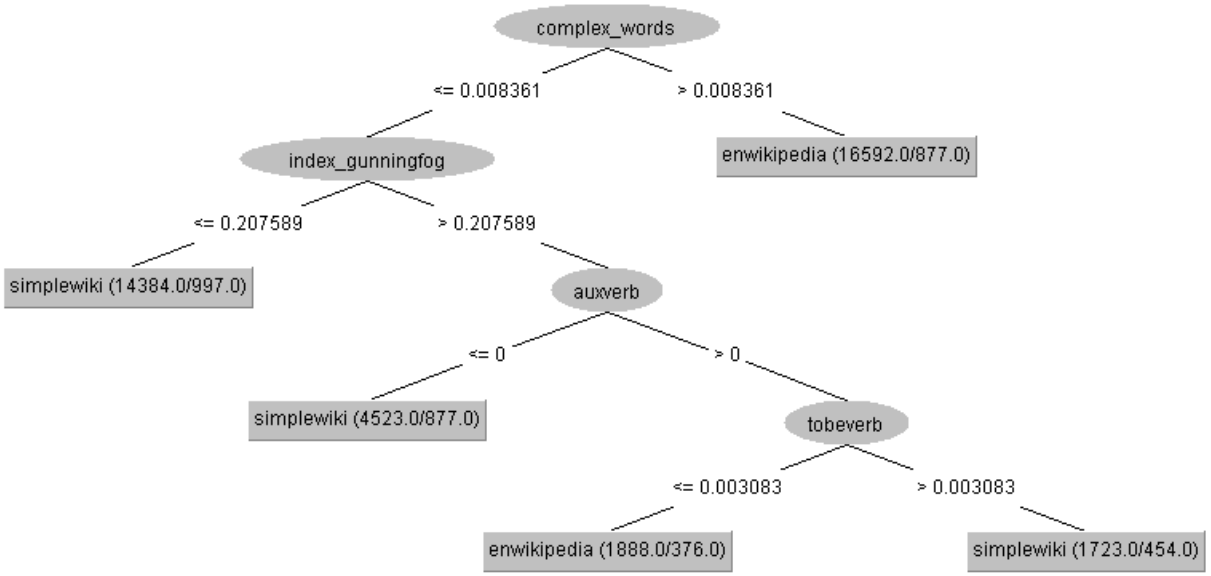

Figura 15 – Visualização da estrutura da árvore construída para o *dataset.*

Fonte: Autoria Própria

A Figura demonstra que, para este algoritmo, grandes quantidades de instâncias podem primeiramente ser classificadas como pertencentes a classe *Wikipedia*, ou seja, um texto com maior índice de dificuldade de leitura, levando em consideração a frequência de palavras complexas presentes no texto. Já o índice de Gunning Fog pode ser um separador de grandes quantidades de instâncias pertencentes a classe *Simple Wikipedia*, textos com menor dificuldade

de leitura. A ausência de verbos auxiliares e a maior frequência de *to be verbs* demonstraram serem aspectos de artigos do *dataset Simple Wikipedia*. *To be verbs* são uma categoria de verbos mais simples da língua inglesa, e portanto, a maior frequência deste tipo de verbo no *dataset Simple Wikipedia* era esperada devido a sua característica. Este algoritmo também, de todos os testados, foi o que apresentou maior acurácia em ambos os métodos de reamostragem. Em segundo lugar na avaliação de acurácia dos algoritmos, o SVM, apresentou também um desempenho condizente com trabalhos da área, como o de Scarton e Aluísio (2010), no qual o algoritmo havia demonstrado a melhor acurácia e resultado comparado aos demais algoritmos. A escolha de diferentes características, principalmente no objetivo de classificação textuais em níveis de dificuldade de leitura, pode ser um fator muito impactante nos resultados obtidos pelos algoritmos. A escolha de um algoritmo probabilístico simples baseado no teorema de Bayes foi proposital para a análise da dificuldade do objetivo de classificação. Mesmo um algoritmo que parte ingenuamente de suposições de fortes independências entre as características, consegue alcançar uma acurácia relevante no objetivo de classificação com as características selecionadas e utilizadas neste trabalho.

O método de árvore de decisão J48 é o que obteve maior acurácia, segundo as características e parâmetros usados no objetivo de classificação das instâncias corretamente. Outros algoritmos podem ser aplicados de maneira similar, tanto de aprendizado de máquina como de redes neurais, podendo apresentar diferentes resultados e acurácias. A utilização de características não citadas neste trabalho também podem se tornar aspectos relevantes para o resultado de classificação dos algoritmos, abrindo espaço para outras pesquisas e trabalhos na área.

5 CONCLUSÕES

Este trabalho apresentou a classificação de textos em níveis de dificuldade de leitura através da utilização de algoritmos clássicos de aprendizado de máquina e bases de dados gratuitas e disponíveis na Internet que continuam a serem atualizadas constantemente como meio de treinamento. O algoritmo de Árvore de decisão apresentou o melhor resultado, utilizando parâmetros padrões do programa WEKA alcançou acurácia de 94,17%, apresentando os atributos frequência de palavras complexas, frequência de verbos auxiliares, *to be verb* e índice de Gunning Fog como importantes para a diferenciação das classes, entretanto, não foram realizados cálculos estatísticos comparativos para explicar o motivo do algoritmo obter este resultado.

Este classificador pode ser utilizado como base para diferentes finalidades, por exemplo, sistemas de recomendação de conteúdo para aprendizes da língua inglesa como segunda língua, uso individual para pessoas com baixo letramento no objetivo de desenvolver a leitura em diferentes contextos, além de base para construção de ferramentas de leitura para auxílio de pessoas com distúrbios cognitivos que afetam a fala, nos aspectos de expressão, entendimento da linguagem e aprendizagem da leitura, como a afasia e dislexia. Possibilita e abre espaço para criação de diversas outras ferramentas e aplicações que auxiliam usuários, complementando ferramentas gratuitas e disponíveis na área de legibilidade e inteligibilidade de textos através do Processamento de Língua Natural. Visa-se também através deste trabalho, auxiliar estudos mais aprofundados sobre aspectos que possam compreender e envolver leituras simples e complexas. Outros resultados obtidos também por este trabalho é a criação e divulgação de bases de dados previamente processadas que podem auxiliar inúmeros trabalhos e pesquisas, além dos *scripts* que também ficarão disponíveis para estudantes e quaisquer interessados na área para livre acesso e uso em repositório[1] online.

[1] https://github.com/LeviMatheus/tcc-readability-score-level

5.1 TRABALHOS FUTUROS

Como dito previamente, diferentes características e métricas textuais podem ser obtidas e utilizadas como atributos para a formação de uma base de dados, sendo uma difícil etapa durante a realização deste trabalho e que exige aprofundamento teórico em, por exemplo, áreas científicas da Linguística e PLN. Abre-se espaço para, como trabalhos futuros, a utilização do classificador desenvolvido como base para a construção de uma ferramenta de recomendação de leituras para usuários aprendizes da língua inglesa como segunda língua.

Por fim é possível que a incorporação do gênero textual na base de dados abra espaço para análises da dificuldade de leitura de textos por categoria e gênero textual.

REFERÊNCIAS

ADDRIANS, P.; ZANTINGE, D. **Data Mining**. Addison-Wesley, 1996. ISBN 9780201403800. Disponível em: <https://books.google.com.br/books/about/Data_Mining.html?id=vbsZAQAAIAAJ&redir_esc=y>.

ALVES, G. E. A. B. P. D. Pré-processamento de Dados em Aprendizado de Máquina Supervisionado. **American Journal of Distance Education**, p. 1 – 204, 2003.

BAKER, W. English as a lingua franca and intercultural communication. **The Routledge Handbook of English as a Lingua Franca**, v. 17, n. 3, p. 25–36, 2018.

BIRD, S. **Cleantext**. 2021. Disponível em: <http://www.nltk.org/>. Acesso em: 24 jun. 2021.

BRACHMAN, R.; ANAND, T. The process of knowledge discovery in databases: A first sketch. In: . [S.l.: s.n.], 1994. p. 1–12.

British Council. Demandas de aprendizagem de inglês no brasil. **São Paulo**, 2014.

BRUCE, A.; BRUCE, P. **Estatística Prática para Cientistas de Dados**. Alta Books, 2019. ISBN 9788550813004. Disponível em: <https://books.google.com.br/books?id=v4-wDwAAQBAJ>.

CARVALHO, J. V. D.; SAMPAIO, M. C. Reconhecimento de Caracteres Manuscritos Utilizando Regras de Associação. 2000.

CASTILHOS, S. Pylinguistics : an open source library for readability assessment of texts written in Portuguese . **Revista de Sistemas de Informação da FSMA**, v. 18, n. July, p. 2–7, 2017.

COVINGTON, M. A. **Natural Language Processing for Prolog Programmers**. [S.l.]: Prentice-hall, Inc., 2013.

CRANENBURGH, A. van. **Readability**. 2021. Disponível em: <https://github.com/andreasvc/readability/>. Acesso em: 23 jun. 2021.

Data Senado. **Educação pública no Brasil**. 2011.

FACELI, K.; LORENA, A. C.; GAMA, J.; CARVALHO, A. C. P. d. L. F. d. **Inteligência artificial: uma abordagem de aprendizado de máquina**. [S.l.]: LTC, 2011.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; PADHRAIC, S. From data mining to knowledge discovery in databases. **AI Magazine**, v. 17, n. 3, p. 37–53, 1996. ISSN 07384602.

FLESCH, R. **A New Readability Yardstick**. [s.n.], 1948. Disponível em: <https://books.google.com.br/books?id=C0xkNQEACAAJ>.

GALISSON, R.; COSTE, D. **Dicionário de didáctica das Línguas**. [S.l.]: Livraria Almedina, 1983.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.], 2016.

GRAESSER, A. C.; MCNAMARA, D. S.; LOUWERSE, M. M.; CAI, Z. Coh-Metrix: Analysis of text on cohesion and language. **Behavior Research Methods, Instruments, and Computers**, v. 36, n. 2, p. 193–202, 2004. ISSN 07433808.

GREENFIELD, J. Readability Formulas For EFL. **JALT Journal**, v. 26, n. 1, p. 5, 2004.

GUDIWADA, P. **Cleantext**. 2021. Disponível em: <https://pypi.org/project/cleantext/>. Acesso em: 23 jun. 2021.

HOUSE, R. W.; RADO, T. An approach to artificial intelligence. **IEEE Transactions on Communication and Electronics**, v. 83, n. 70, p. 111–116, 2013. ISSN 0536-1532.

KINCAID, J. P. J.; JR, R. F.; ROGERS, R. L.; CHISSOM, B. S. B.; Fishburne Jr, R. P.; ROGERS, R. L.; CHISSOM, B. S. B. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel (No. RBR-8-75). Naval Technical Training Command Millington TN Research Branch. **Naval Technical Training Command Millington TN Research Branch**, 1975. Disponível em: <http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA006655>.

KORFHAGE, R. R. **Armazenamento e recuperação de informações**. [S.l.: s.n.], 1997. ISSN 0471143383.

LEFFA, V. J. Aspectos da leitura. n. October, 1996. Disponível em: <http://www.ufrgs.br/textecc/traducao/teorias/files/aspectos_leitura.pdf>. Acesso em: 06 jul. 2021.

LOPES, L. P. d. M. Inglês e globalização em uma epistemologia de fronteira: Ideologia lingüística para tempos híbridos. **DELTA Documentacao de Estudos em Linguistica Teorica e Aplicada**, v. 24, n. 2, p. 309–340, 2008. ISSN 01024450.

LUCCA, J. L. D. Lematização versus Stemming. p. 16, 2002.

MARTINS, C. A.; MONARD, M. C.; MATSUBARA, E. T. PreTexT : uma ferramenta para pré-processamento de textos utilizando a abordagem bag-of-words. **Instituto de Ciências Mateáaticas e de Computação**, n. April, p. 57, 2003.

MARTINS, C. B. M. J.; MOREIRA, H. O campo CALL (Computer Assisted Language Learning): Definiçõ es, escopo e abrangência. **Calidoscopio**, v. 10, n. 3, p. 247–255, sep 2012. ISSN 21776202.

MELITZ, J. English as a global language. In: **The Palgrave Handbook of Economics and Language**. [S.l.]: Palgrave Macmillan, 2016. p. 583–615. ISBN 9781137325051.

MITCHELL, T. M. T. M. **Machine Learning**. [S.l.: s.n.], 1997. 414 p. ISBN 0070428077.

PYDATA DEVELOPMENT TEAM. **Pandas**. 2021. Disponível em: <https://pypi.org/project/pandas/>. Acesso em: 23 jun. 2021.

Python Software Fundation. **What is Python? Executive Summary**. 2021. Disponível em: <https://www.python.org/doc/essays/blurb/>. Acesso em: 21 jun. 2021.

RUSSELL, S.; NORVIG, P. **A AI I PRENTICE HALL SERIES IN ARTIFICIAL INTELLIGENCE**. [S.l.], 2013.

SANTOS, T. M. D. **Aprendizado de máquina: uma abordagem estatística**. [S.l.: s.n.], 2005. 99–117 p. ISBN 9786500024104.

SCARTON, C. E.; ALUÍSIO, S. M. Análise da Inteligibilidade de textos via ferramentas de Processamento de Língua Natural : adaptando as métricas do Coh-Metrix para o Português. **Linguamatica**, v. 2, p. 45–62, 2010. ISSN 16470818.

SEBASTIANI, F. Machine Learning in Automated Text Categorization. **ACM Computing Surveys**, v. 34, n. 1, p. 1–47, 2002. ISSN 03600300.

Stack Overflow. **Stack Overflow Trends**. 2021. Disponível em: <https://insights.stackoverflow.com/trends?tags=r%2Cstatistics%2Cc%23%2Cpython%2Cjavascript%2Cjava%2Cc%2B%2B%2Cobjective-c%2Cscala>. Acesso em: 24 jun. 2021.

SUTTON, R. S.; BARTO, A. G. No Play, Bad Work, and Poor Health. **The Lancet**, v. 258, n. 6685, p. 675–676, 2017. ISSN 01406736.

WILLIAMS, S. Natural Language Generation (NLG) of discourse relations for different reading levels. 2004. Disponível em: <https://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.408787>.