UNIVERSIDADE DE CAXIAS DO SUL

CAIO GUSTAVO RODRIGUES DA SILVA

AVALIAÇÃO DA QUALIDADE DA INFORMAÇÃO NA ÁREA DA SAÚDE: APLICAÇÃO DE ALGORITMOS DE APRENDIZADO DE MÁQUINA

CAXIAS DO SUL

2017

CAIO GUSTAVO RODRIGUES DA SILVA

AVALIAÇÃO DA QUALIDADE DA INFORMAÇÃO NA ÁREA DA SAÚDE: APLICAÇÃO DE ALGORITMOS DE APRENDIZADO DE MÁQUINA

Monografia apresentada como requisito para a obtenção do Grau de Bacharel em Sistemas de Informação da Universidade de Caxias do Sul

Orientador: Profª. Carine Geltrudes Webber.

CAXIAS DO SUL

2017

AGRADECIMENTOS

Quero expressar meus agradecimentos a todas as pessoas que de alguma forma colaboraram para que este trabalho fosse realizado. Em especial a minha orientadora, Profª. Carine Geltrudes Webber, pela sua competência, paciência e orientação durante todo o desenvolvimento desta monografia. Agradeço de forma toda especial, aos meus familiares pela compreensão e apoio não só durante o período de desenvolvimento deste projeto, mas em todos os momentos durante a minha graduação, aos meus amigos e colegas, por sempre me incentivarem durante anos vividos juntos na universidade, e a minha namorada, que sempre me ajudou, incentivou e sempre esteve ao meu lado nos momentos de alegrias, felicidades, frustações e dificuldades.

RESUMO

A evolução contínua das tecnologias de informação proporciona às pessoas e aos pacientes um acesso cada vez mais fácil e instantâneo a artigos, receitas e dicas relacionadas à saúde e às condições de vida. Frente a isso, a necessidade de revisão e garantia da qualidade destas informações existentes na Internet é de grande importância, uma vez que decisões baseadas em informações equivocadas podem provocar consequências graves e permanentes ao usuário/paciente. O conceito de análise textual automática é derivado dos estudos baseados em Mineração de Textos, podendo ser caracterizado pela descoberta e extração de padrões relevantes e informações úteis a partir de elementos textuais. O objetivo principal deste trabalho é aplicação de técnicas e conceitos relacionados à Mineração de Textos em uma ferramenta Web, com o propósito de se automatizar e alcançar resultados semelhantes aos de especialistas humanos no processo de classificação de textos da área da saúde. Para alcançar o objetivo proposto foram estudados algoritmos de aprendizagem de máquina (J48, Naïve Bayes, Support Vector Machines, K-Nearest Neighbor) que fizeram parte de estudos recentes ligados a saúde em outros idiomas, a fim de utilizá-los como base para testes e para o desenvolvimento da ferramenta. Complementando estes estudos, examinou-se os benefícios proporcionados pela aplicação da estrutura de um Ensemble de algoritmos de aprendizado de máquinas. Após a realização destes estudos, coletou-se junto a especialistas humanos amostras de dados textual. Estas amostras foram divididas em categorias, que segundo os especialistas, são determinantes para a classificação final de um texto da área da saúde, sendo assim os arquivos foram divididos entre: Descrição do tratamento, Benefícios do tratamento, Consequências do tratamento, Influência na qualidade de vida do paciente, Riscos do tratamento. A fim de se assemelhar às percepções dos especialistas, foi implementado na ferramenta Web um Ensemble de algoritmos. Cada algoritmo do Ensemble representa uma das categorias definidas pelos especialistas. Após esta implementação, foi possível realizar os testes utilizando as amostras selecionadas. Após a realização dos testes, os algoritmos que obtiveram os melhores resultados foram utilizados na versão final da ferramenta. Com a utilização destes algoritmos foi possível obter uma taxa de 90,75% de convergência entre as classificações realizadas pela ferramenta e as classificações realizadas pelos especialistas. Como conclusão, considera-se que os resultados são promissores e evidenciam a viabilidade de uso de técnicas de aprendizado automático no tratamento de textos da área da saúde.

Palavras-chave: Avaliação automática de textos. *Ensemble* de algoritmos. Mineração de Textos. Aprendizado de Máquina. *Weka*.

ABSTRACT

The continuos evolution of information techonology provides people and patients an instant and easier acess to articles, prescriptions and tips related to health and life conditions. Against this, the necessity of reviewing and insuring the quality of the information on the Internet is of great importance, once decisions based on misguided informations can cause severe and permanent consequences to the reader/patient. The concept of automatic textual analysis is derived from studies based on Data Mining, which can be characterized by discovery and extraction of relevant patterns and useful information from textual elements. The main goal of this paper is to apply techniques and concepts related to Data Mining in a Web tool, with the purpose to automate and achieve similar results as human experts when assorting health area texts. To reach the proposed goal, machine learning algorithmes were studied *(J48, Naïve Bayes, Support Vector Machines, K-Nearest Neighbor)* which were part of recent studies related to health in other languages, in order to use them as basis for tests and for the tool development. In adittion to this studies, the benefits provided by the aplication of an Ensemble of machine learning algorithmes were examined. After this, samples of textual data were colected with human experts. Those samples were divided in categories, which according to experts are determinant to the final assortment of a health text, so the files were divided between: Treatment description, Treatment benefits, Treatment consequences, Influence os Patient's quality life, Treatment risks. An Ensemble of algorithmes was implemented in the web tool in order to resemble itself to the experts perception. Each algorithm represents one of the categories established by experts. After the implementation, it was possible to do tests using the selected samples. After conducting tests, the algorithmes that got the best results were used on the final version of the tool. Using those algorithmes it was possible to obtain a 90,75% convergence rate between the assortments made by the tool and by the experts. In conclusion, the results can be considered promising and they evidence the viability of using autimatic learning techniques on treatment of health area texts

Keywords: Automatic textual analysis. *Ensemble* of Algorithms. Text Mining. Machine learning. *Weka*.

LISTA DE FIGURAS

LISTAS DE QUADROS

LISTA DE TABELAS

LISTA DE ALGORITMOS

LISTA DE SIGLAS

Sigla	Significado
ML	*Machine Learning*
IA	Inteligência Artificial
MT	Mineração de Textos
KDT	*Knowledge Discovery from Text*
PLN	Processamento de Linguagem Natural
RI	Recuperação de Informação
SVM	*Support Vector Machine*
KNN	*K-Nearest Neighbor*
SQL	*Structured Query Language*
Weka	*Waikato Environment for Knowledge Analysis*
GNU	*General Public License*
DLL	*Dynamic-link Library*
JAR	*Java Archive*
.NET	*DotNET*
URL	*Uniform Resource Locator*

SUMÁRIO

1. INTRODUÇÃO

A expansão constante da utilização das tecnologias de informação, auxiliam a disseminação e compartilhamento de informações em todos os setores da sociedade. Na área da medicina, estas ferramentas se tornaram um meio fácil e prático para a obtenção de receitas, dicas e informações relacionadas as condições de saúde e vida das pessoas.

Conforme Wilkes (2014), muitas informações existentes na Internet relacionadas à saúde não podem ser consideradas corretas e confiáveis. Este cenário provê a necessidade da avaliação destes textos a fim de garantir uma tomada de decisão correta por parte do paciente, uma vez que informações incorretas podem ocasionar acidentes graves e irreversíveis. Porém, a análise e determinação da qualidade e coerência da informação, disponíveis em Português, só deve ser realizada por especialistas neste assunto, aqueles que possuem a capacitação necessária para tal.

Este cenário demonstra a necessidade da utilização das tecnologias de informação no auxílio a sociedade, sem grande expertise na área da saúde, nesta análise da qualidade da informação descoberta na Internet. Tornando muito importante o desenvolvimento de uma ferramenta que proporcione a avaliação de textos relacionados a saúde de maneira automática.

1.1 TEMA E PROBLEMA DE PESQUISA

O seguinte projeto visa o refinamento e aprimoramento de algoritmos de extração e avaliação textual em uma ferramenta Web, a fim de garantir resultados satisfatórios e conclusivos através da utilização de *Machine Learning* (ML) para a avaliação da qualidade de textos na área da saúde. Tendo em vista o cenário apresentado, chega-se a seguinte questão problema: “Como algoritmos de aprendizado de máquina podem auxiliar na avaliação da qualidade em textos da área da saúde?”

1.2 OBJETIVOS

1.2.1 Objetivo geral

Aplicar técnicas de aprendizagem de máquina em uma ferramenta na plataforma Web para avaliar a qualidade textual de documentos referentes a área da saúde, refinando assim seus resultados e a sua visualização gráfica.

1.2.2 Objetivos específicos

Para atingir o objetivo geral serão propostos os seguintes objetivos específicos:

a) Pesquisar os estudos recentes referentes a qualidade da informação em saúde.
b) Identificar algoritmos de aprendizagem automática supervisionada, válidos para análise de dados textuais e suas configurações.
c) Definir critérios de qualidade para o trabalho.
d) Aplicar e avaliar o uso de algoritmos para a tarefa de avaliação de informação em saúde.
e) Fornecer recursos de visualização de dados sobre a qualidade da informação em saúde.

1.3 METODOLOGIA

O presente projeto apresenta uma pesquisa com implementação, contemplando fases de pesquisa, estudos e implementação da ferramenta proposta. A organização do projeto está dividida em quatro fases sequenciais que visam atender à os objetivos estabelecidos.

Primeiramente, a etapa de pesquisa e estudo de conteúdo bibliográfico para formação da base de conhecimento que deve ser a base pra o desenvolvimento do capítulo de fundamentação teórica, ao final desta etapa.

Posteriormente, a segunda etapa abrange a elaboração da proposta de solução e os critérios de avaliação para validação da proposta assim como especificação dos algoritmos e a modelagem do sistema.

A terceira etapa atende ao quesito da implementação. Nesta etapa será desenvolvido junto ao sistema as diretrizes propostas na solução anterior, junto aos testes e validações do software.

Por fim, a última etapa contempla a realização da comparação entres os resultados obtidos pela ferramenta com os resultados dos especialistas, a fim de se avaliar a taxa de acerto do software e sua confiabilidade.

1.4 ESTRUTURA DO ESTUDO

Este projeto está estruturado da seguinte maneira: no Capítulo 2 estão presentes os estudos relacionados a avaliação textual qualitativa de forma automática, com foco especialmente da utilização desta técnica na área da saúde. O Capítulo 3 apresenta as alterações e implementações realizas no software, tal como o método de pesquisa, arquitetura e modelagem do sistema, ferramentas de auxílio no desenvolvimento e as métricas e testes realizados para a validação do sistema. O capítulo 4 conclui esta monografia e apresenta trabalhos futuros.

2. AVALIAÇÃO E CLASSIFICAÇÃO AUTOMÁTICA DE TEXTOS DA SAÚDE

Os processos de avaliação e classificação automática de informações textuais consistem na utilização de algoritmos de aprendizagem de máquina para a exploração e análise de maneira inteligente e automática de grandes volumes de dados textuais com a intenção de encontrar algum conhecimento útil. Com propósito de auxiliar o entendimento do conceito de avaliação textual, o presente capítulo apresenta técnicas e abordagens utilizadas para a avaliação automática da qualidade de informações textuais da área da saúde.

2.1 QUALIDADE DA INFORMAÇÃO NA SAÚDE

Informação na saúde pode ser caracterizada como todo conteúdo relacionado a condições de vida e morte de indivíduos e populações, além de conteúdos sobre comportamentos, produtos e serviços relacionados ao corpo e a saúde (MORAES, 2002, apud MENDONÇA e NETO, 2015). Com a globalização das tecnologias de informação, a disseminação e o acesso aos conteúdos presentes na internet, tornou-se algo simples a qualquer pessoa, sendo possível que qualquer indivíduo publique algo independente de sua veracidade, coerência e coesão.

Segundo Ballou et. Al (BALLOU, MADNICK e WANG, 2004), qualidade da informação é a sua aptidão para uso. Dizem ainda que a qualidade da informação deve ter uma alta prioridade, e as consequências de não tê-la pode ser devastador. A própria existência de uma organização pode ser ameaçada pela má qualidade da informação. Na saúde, a qualidade da informação é uma das mais importantes características a serem consideradas (LOPES, 2004). Segundo Wilkes (2015), pacientes podem adotar medidas incorretas com base em informações de má qualidade, podendo levar a consequências graves e irreversíveis.

Através de um estudo realizado em 2015, Mendonça e Neto sugerem o agrupamento dos principais critérios, desenvolvidos por iniciativas nacionais e internacionais, de avaliação da qualidade de informações disponíveis na internet na área da saúde. Com base nestes estudos criaram três dimensões de avaliação: conteúdo, técnica e design.

2.1.1 Conteúdo

A dimensão conteúdo envolve três critérios: abrangência, acurácia e inteligibilidade. Abrangência diz respeito a quantidade de informações abrangidas e detalhadas que o site apresenta. O grau de concordância entre as informações oferecidas pelo texto é caracterizado pelo critério da acurácia. A dimensão conteúdo ainda abrange o critério da inteligibilidade, que visa avaliar o grau de compreensão das informações obtidas no texto.

2.1.2 Técnica

A dimensão técnica abrange os critérios: credibilidade, segurança e privacidade das informações.

Em relação aos dois primeiros critérios, os sites devem prover nomes e demais informações referentes ao autor ou instituição responsável pelo texto além da data de atualização da publicação. O objetivo do site deve estar claro, seja ele comercial, informativo ou educacional.

A privacidade do usuário é outro aspecto abordado nesta dimensão. O site deve estar de acordo com as legislações vigentes de cada país, além de solicitar permissão do usuário antes de qualquer obtenção e retenção de dados, informando o motivo da coleta, termos e as políticas de segurança.

2.1.3 Design

Última dimensão apresentada por Mendonça e Neto (2005), qualquer site relacionado a saúde deve oferecer facilidade de uso, navegação e acessibilidade de acordo com as expectativas dos usuários. A usabilidade, interface, rapidez, compatibilidade com os mais diversos navegadores deve ser garantida e validada, garantindo acesso a todas as pessoas independente de sua disponibilidade de recursos.

2.2 TRABALHOS RELACIONADOS

Os estudos em torno da descoberta de conhecimento automática em textos da área da saúde ganham cada vez mais relevância na comunidade de inteligência artificial (IA). Diversos resultados positivos estão sendo publicados e validados tanto pela comunidade de IA quando pelos estudiosos da saúde.

Szlosek e Ferretti (2016), apresentam resultados satisfatórios quanto à utilização da IA na avaliação de arquivos gerados por uma máquina de ressonância para identificação e classificação de concussões na região craniana. O sistema proposto foi validado através de testes com três algoritmos distintos: máquina de vetor de suporte, k-vizinhos mais próximos e árvore de decisão. O algoritmo de máquina de vetor de suporte apresentou os melhores resultados quanto a classificação correta.

Zhang et. Al (2014), propõe a utilização dos algoritmos de aprendizado de máquina para a avaliação da qualidade de páginas web relacionadas ao tratamento da depressão. Utilizando como fonte de conhecimento guias médicos relacionados ao tratamento da doença, é realizado a comparação semântica com os textos encontrados na web, a fim de avaliar a qualidade destes textos. Para garantir a coesão dos resultados, dois avaliadores humanos foram contratados para classificar os guias em positivos e negativos e destacar nos positivos as palavras e termos chaves que os levam a uma classificação positiva. Além da classificação dos guias, os avaliadores humanos realizaram a classificação de algumas páginas web a fim de comparar estes resultados ao do sistema. O sistema foi construído utilizando o algoritmo *Naïve Bayes*. O resultado final apresentado pelo sistema foi considerado efetivo e conclusivo, com a classificação correta dos textos em 84% dos casos.

Yamada et. Al (2015), desenvolveram um sistema de auditoria automática, a fim de verificar a qualidade dos registros médicos de consentimento informado[1] do idioma japonês. Utilizando a técnica de análise morfológica, realizou-se o treinamento do sistema através de um dicionário médico, capaz de oferecer conteúdo suficiente para a classificação dos registros. Para a mineração de textos optou-se pelo algoritmo de máquina de vetores de suporte, classificando os arquivos

[1] Consentimento informado: É o processo pelo qual o prestador de cuidados de saúde revela informações ao paciente para que este possa fazer a escolha voluntária de aceitar ou recusar o tratamento (DE BORD, 2007).

em positivos e negativos. O sistema conseguiu avaliar corretamente 89.4% dos textos empregues na etapa de validação.

Os estudos e resultados apresentados indicam um cenário positivo e promissor quanto à utilização dos algoritmos de aprendizado de máquina na avaliação textual de arquivos na área da saúde, dando sustentação a adaptação destes princípios a um sistema específico para à língua portuguesa.

2.3 APRENDIZADO DE MÁQUINA

Os estudos sobre *Machine Learning* (ML) ou Aprendizado de máquina ganham cada vez mais destaque na comunidade da computação. O aumento contínuo da quantidade de dados disponíveis em todas as áreas organizacionais e sociais fornecem razões capazes de justificar o conceito de que a análise inteligente de dados deve se tornar ainda mais difundida e uma ferramenta chave para o desenvolvimento tecnológico. (SMOLA e VISHWANATHAN, 2008)

Aprendizado de máquina pode ser definido como uma área de estudo da IA que produz algoritmos capazes de proporcionar conhecimento a computadores utilizando-se de dados de eventos passados. O aprendizado de máquina é capaz de evidenciar características e padrões que dificilmente seriam perceptíveis de maneira clara para um humano, utilizando técnicas triviais de análise de dados. (AMARAL, 2016, p. 2)

O estudo dos algoritmos de aprendizado de máquina pode ser divido em três grupos: algoritmos de aprendizado supervisionado, não-supervisionado e semi-supervisionado.

2.3.1 Algoritmos supervisionados

Segundo Shalev-Shwartz e Ben-David (2014), algoritmos supervisionados utilizam experiência para ganhar conhecimento. Os algoritmos de aprendizado supervisionado são treinados através de rótulos e características pré-definidas, fazendo o reconhecimento e a classificação das informações de acordo com o aprendizado adquirido na fase de treinamento. Estes podem ser divididos entre os de classificação, algoritmos que classificam as informações em categorias, ou regressão, que resulta em valores contínuos.

São exemplos de algoritmos supervisionados os algoritmos: máquina de vetor de suporte, árvore de decisão, k-vizinhos mais próximos e *Naïve Bayes* (LE, 2016).

2.3.2 Algoritmos não supervisionados

Algoritmos não supervisionados são utilizados quando não se possui rótulos prévios para se classificar os dados (AMARAL, 2016, p 9). O processamento dos dados de entrada ocorre com o objetivo de criar um resumo ou uma versão compactada destes (SHALEV-SHWARTZ e BEN-DAVID, 2014, p 23).

Segundo Brownlee (2016), algoritmos não supervisionados possuem dados de entrada e nenhuma resposta como saída, sua utilização resulta em grupos *(clusters)* de dados com características em comum, estes geralmente necessitam de uma análise para a extração de conhecimento de cada agrupamento (MONARD E BARANAUSKAS, 2003).

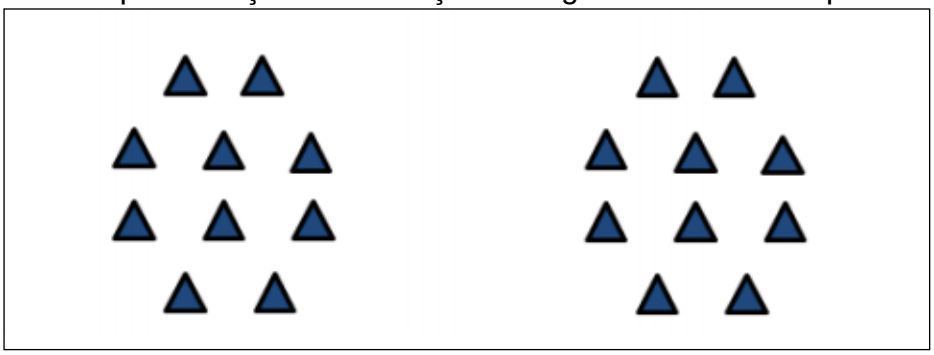
São exemplos de algoritmos não supervisionados os algoritmos: Algoritmos baseados em Centroid, Probabilístico, Redução da Dimensionalidade e Redes Neurais (LE, 2016).

2.3.3 Semi-supervisionados

De acordo com Filho (2009), os algoritmos semi-supervisionados são uma abordagem intermediária entre os algoritmos de aprendizagem supervisionada e não supervisionada. Sua característica principal é a capacidade de aprender a partir de dados rotulados e não rotulados, podendo exercer as tarefas de classificação e agrupamento. São exemplos de algoritmos semi-supervisionados os algoritmos: *COP-k-means, SEEDED-k-means e Co-Training* (SANCHES, 2003).

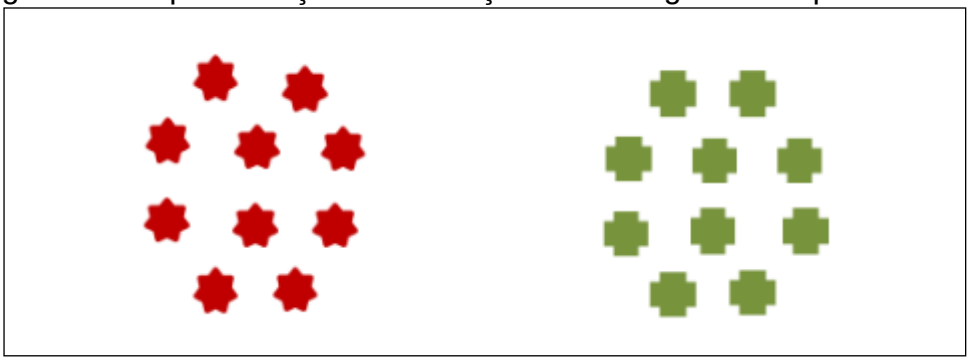
As Figuras 1, 2 e 3, apresentadas por Filho (2009), apresentam a diferença entre os três tipos de abordagens. A Figura 1 representa o resultado da aplicação de um algoritmo não supervisionado, formando dois agrupamentos de dados. Já a Figura 2, demonstra o produto final de um algoritmo supervisionado, classificando os dados em verdes e vermelhos. Por fim, a Figura 3, demonstra a utilização de um algoritmo semi-supervisionado, realizando a classificação e agrupamento dos dados.

Figura 1 – Representação da utilização de algoritmo um não supervisionado



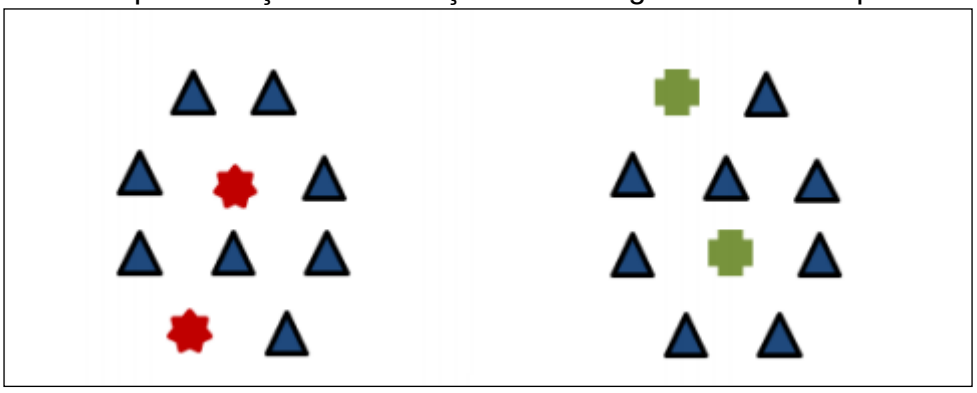
Fonte: Filho (2009)

Figura 2 – Representação da utilização de um algoritmo supervisionado



Fonte: Filho (2009)

Figura 3 – Representação da utilização de um algoritmo semi-supervisionado



Fonte: Filho (2009)

2.4 MINERAÇÃO DE TEXTOS

De acordo com Ambrósio e Morais (2007), o campo de estudo de Mineração de textos (MT) ou *text minning*, tem origem relacionada a área de Descoberta de conhecimento em Textos (*Knowledge Discovery from Text* - KDT). Segundo Beppler et al. (apud AMBRÓSIO E MORAIS, 2007), KDT engloba maneiras inteligentes e

automáticas que auxiliem a análise de grandes volumes de dados textuais com a intenção de encontrar algum conhecimento útil. A busca de informações específicas em documentos, a análise qualitativa e quantitativa de grandes volumes de textos, e a melhor compreensão de textos disponíveis em documentos são elementos essenciais do KDT, que contribuem para o desenvolvimento da área de MT (AMBRÓSIO E MORAIS, 2007).

Já Aranha (2006), relaciona os estudos em torno da MT com a mineração de dados, que se destina a encontrar padrões em bancos de dados estruturados, enquanto a mineração de textos objetiva a extração de conhecimento em dados não-estruturados (ARANHA, 2006).

Lopes (2004) define para a mineração de texto a finalidade de extração de padrões relevantes e não triviais ou a descoberta de conhecimento a partir de elementos textuais não-estruturados. Aranha (2006) descreve o estudo da mineração de textos como a extração de regularidades, padrões e tendências de conteúdos textuais, geralmente para objetivos específicos. A aplicação da mineração de textos compreende a utilização de algoritmos computacionais que analisam e identificam informações, em textos, frases ou palavras, que normalmente não seriam recuperadas através da utilização das técnicas comuns de consulta, tendo em vista o formato não-estruturado em que estes estão descritos (AMBRÓSIO E MORAIS, 2007).

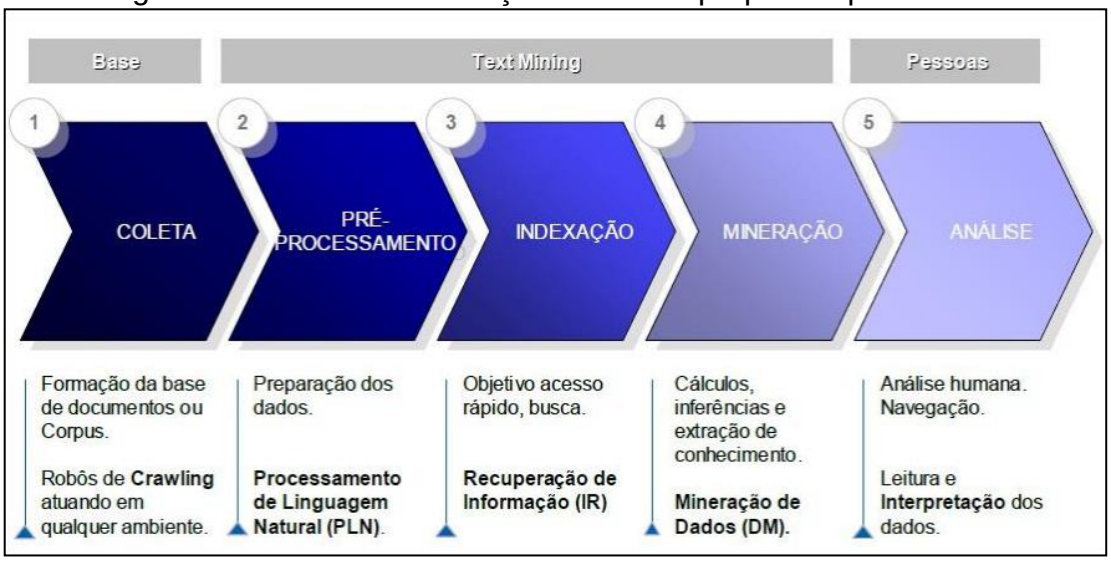
De acordo com Silva (2010), dados estruturados são caracterizados por conterem uma estrutura lógica de armazenamento, como um banco de dados relacional. Os dados não estruturados, não contém nenhum tipo de estrutura lógica de armazenamento, aspecto comum entre conteúdos textuais.

A mineração de textos pode ser considerada um campo multidisciplinar por envolver várias áreas de conhecimento, entre elas a informática, estatística, linguística e ciência cognitiva (ARANHA, 2007). Estima-se que cerca de 80% das informações disponíveis nas organizações encontram-se no formato de textos (HE et. al, 2013).

Aranha (2007, p. 19), propõe um processo de MT dividido em cinco fases. A Figura 4 é um modelo didático proposto por Aranha (2007, p. 19), com o propósito de representar as cinco fases do processo de MT: coleta, pré-processamento, indexação, mineração e análise. A Figura 4, evidencia a multidisciplinaridade do processo de MT, contentando etapas compostas pelas áreas de Recuperação de

Informação e Linguística Computacional, além de demonstrar a interação existente entre os resultados do processo com os humanos, estes responsáveis pela análise final dos resultados.

Figura 4 – Fases da mineração de dados propostas por Aranha



Fonte: Aranha (2007)

As próximas seções visam apresentar detalhadamente cada etapa do processo.

2.4.1 Coleta

A etapa de coleta de dados tem como objetivo formar a base de conteúdo textual a sustentar todo o trabalho, podendo esta ser estática, os dados são sempre os mesmos, ou dinâmica, há atualização dos conteúdos a todo o momento através da remoção e edição de arquivos ou substituição da base por uma completamente nova (ARANHA, 2007, p. 42).

Conforme Gomes (2009), a coleção de textos pode ser oriunda de diversas fontes diferentes, como documentos pré-armazenados de uma forma organizada ou elementos textuais de diversas fontes, sem nenhuma estrutura de armazenamento. De acordo com Aranha e Passos (2006), a base de dados, também é conhecida como corpus.

2.4.2 Pré-processamento

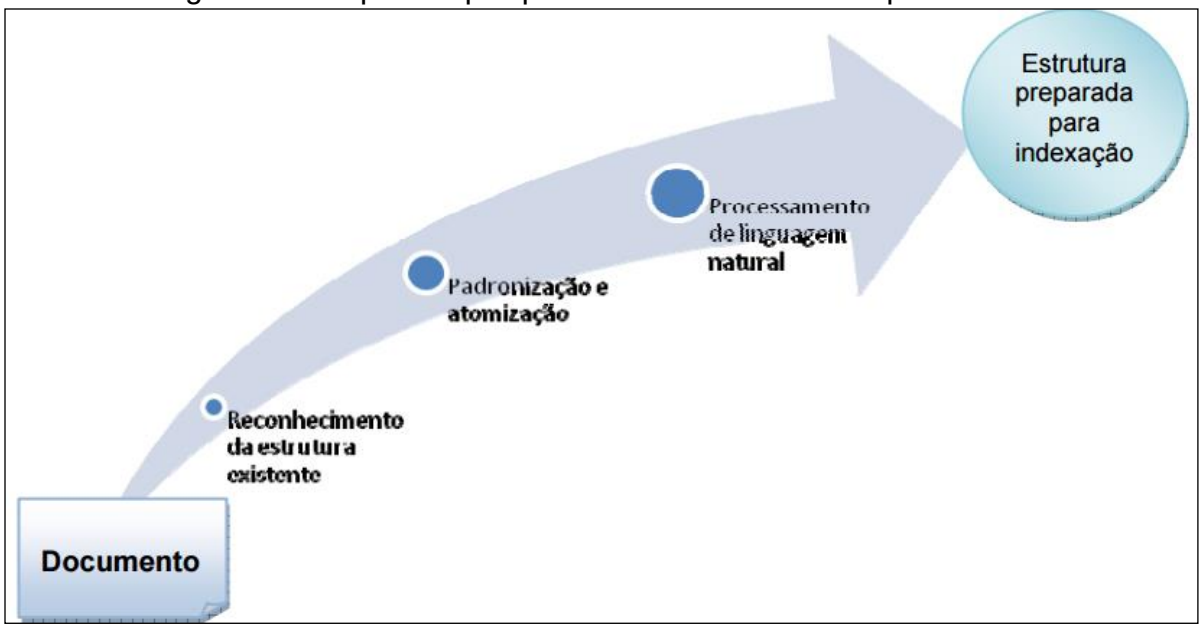
Esta etapa consiste na organização e transformação do conjunto de textos coletados, contemplando a identificação, compactação, tratamento de dados corrompidos e atributos irrelevantes em uma coleção de dados estruturados com representação de atributo-valor (ARANHA, 2007, p 42), possibilitando a alimentação dos algoritmos de aprendizado de máquina. A fase de pré-processamento deve prover uma redução dimensional de dados, além de tentar identificar similaridades em função da morfologia ou do significado dos termos no texto (EBECKEN, 2003, apud. AMBRÓSIO e MORAIS, 2007).

Uma das técnicas utilizadas nesta etapa é a remoção de palavras sem relevância para a análise e descoberta do conhecimento no ambiente desejado. A remoção destas palavras é conhecida como *stopwords*, estas geralmente são adicionadas a um dicionário de palavras conhecido como *stoplist.* Durante a fase de análise e extração das características dos textos estas serão desconsideradas (GOMES, 2009).

De acordo com Gomes (2009), modelos mais elaborados que incorporam as técnicas de Processamento de Linguagem Natural (PLN) também podem ser aplicadas durante a etapa de pré-processamento, com a desvantagem de que a ferramenta possa se tornar dependente de algum idioma e suas especificidades.

Gomes (2009) descreve a etapa de pré-processamento, com a utilização das técnicas de PLN, conforme e Figura 5.

Figura 5 – Etapas de pré-processamento definidas por Gomes



Fonte: Gomes 2009.

2.4.2.1 *Identificação de termos*

Esta técnica tem como objetivo identificar os termos existentes no texto, sejam eles simples ou compostos (AMBRÓSIO E MORAIS, 2007).

Segundo Ambrósio e Morais (2007), na identificação de termos simples devem ser identificados as palavras existentes no documento, eliminando símbolos e formatação. Nesta etapa pode ocorrer a verificação de erros ortográficos, conversão das palavras para maiúsculo ou minúsculo, além da padronização de números e datas.

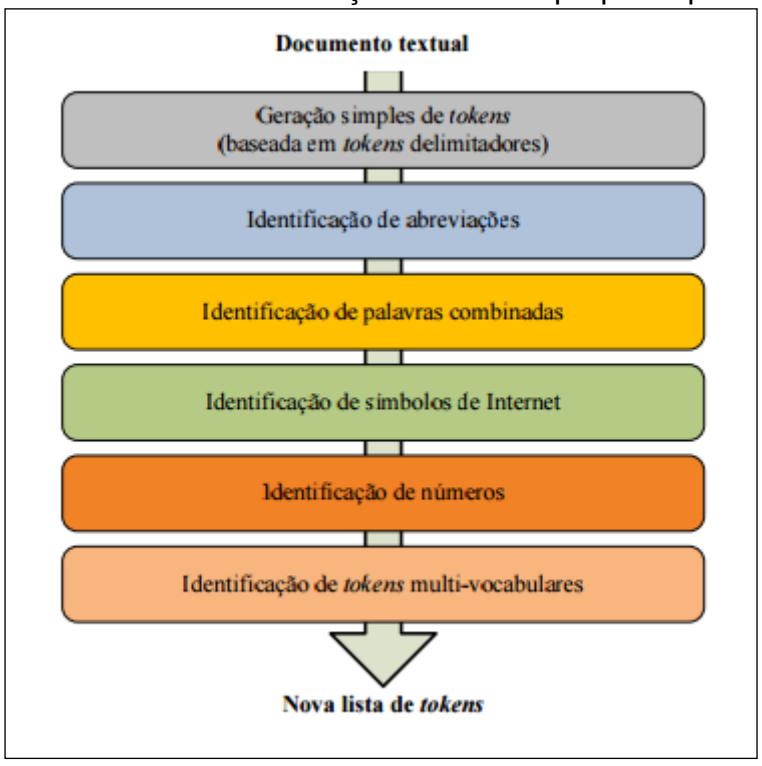
Palavras utilizadas em conjunto podem ter significados diferentes de quando utilizadas sozinhas, isto pode ocorrer pois existem conceitos que somente são descritos por meio da utilização de duas ou mais palavras adjacentes (AMBRÓSIO E MORAIS, 2007). A identificação destes termos compostos pode ocorrer de duas formas. A primeira maneira, apresentada por Ambrósio e Morais (2007), envolve a identificação dos termos que aparecem com grande frequência em uma coleção de documentos. Estes termos devem ser apresentados ao usuário e ele deve escolher aquelas consideradas relevantes. A segunda compreende a criação de um dicionário de termos, que contará com os termos compostos mais expressivos para a análise.

2.4.2.2 *Tokenização*

A *tokenização* é primeiro estágio da fase de pré-processamento de elementos textuais. O texto a ser analisado deve ser agrupado conforme as delimitações impostas seja ela espaços em branco, vírgulas, pontos etc (ARANHA, 2007). De acordo com Aranha (2007), cada agrupamento ou termo encontrado é nomeado como *token.* Uma das dificuldades da aplicação desta técnica é a ambiguidade dos delimitadores. O “ponto”, por exemplo, pode ser utilizado como delimitador de uma sentença e como um abreviador de palavras (GOMES, 2009).

A Figura 6 apresenta a metodologia de criação de *tokens* proposta por Konchady (2006 apud SOARES, 2008), com a utilização de dicionários de dados e regras para a formação de palavras, visando manter o mesmo nível semântico existente antes do processo.

Figura 6 – Processo de identificação de *tokens* proposto por Konchady



Fonte: Soares (2008).

2.4.2.3 *S*temming

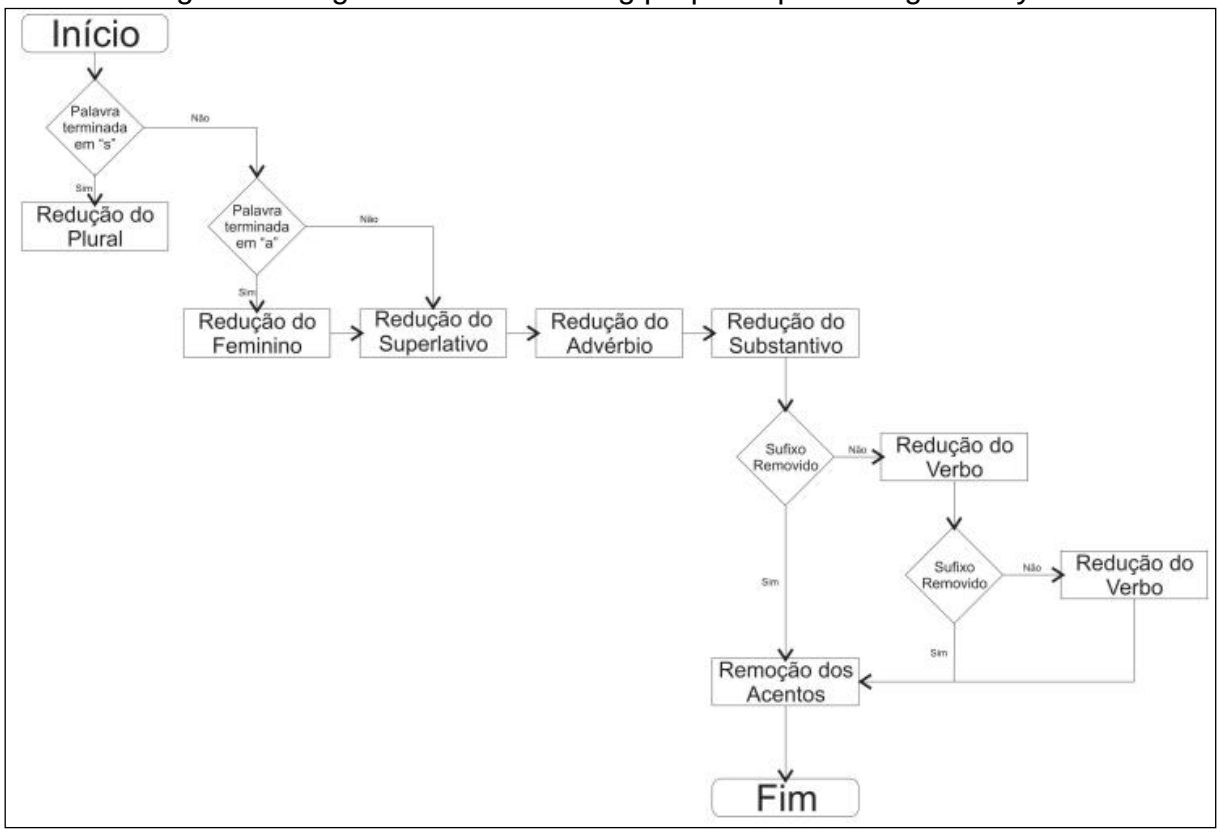
Compreende a unificação da representação de palavras associadas ao mesmo conceito, propondo o agrupamento destas palavras apenas em uma ou em seu radical, indicando que possuem o mesmo significado (ARANHA, 2007, p 65).

A utilização desta técnica pode ser exemplificada pela utilização das palavras, “doença’, “doenças” e “doentio”, que através da normatização pode ser reduzida ao radical “doen”, evitando que alguma destas palavras seja ignorada na análise por estar escrita de maneira diferente (SILVA, 2010).

Esta técnica proporciona a redução do vetor de *token* a ser explorado, fornecendo maior rapidez e coesão na análise realizada pela ML.

Segundo Gomes (2009), cada idioma possui suas próprias regras para a retirada de aumentativos, diminutivos, pluralização, tempos verbais, entre outras características. Orengo e Huyck (2001, apud AMBRÓSIO E MORAIS, 2007), apresentam um algoritmo de *stemming*, composto de 8 etapas, adaptado para a língua portuguesa conforme a Figura 7.

Figura 7 – Algoritmo de Stemming proposto por Orengo e Huyck



Fonte: Orengo e Huyck (2001 apud MORAIS e AMBRÓSIO, 2007).

As etapas propostas por Orengo e Huyck (2001), são descritas por Ambrósio e Morai (2007) a seguir:

1. Remoção do plural: remoção da letra “s” do final das palavras, com a presença de uma lista de exceções de palavras terminadas em “s” e que não estão no plural (exemplo: lápis).
2. Remoção do feminino: Palavras no formato feminino são transformadas para o correspondente em masculino (exemplo: Americana -> Americano)
3. Remoção de advérbio: Etapa mais simples do processo, uma vez que o sufixo que caracteriza um advérbio é “mente”. Neste caso, também existe uma lista de exceções. (exemplo: Lamentavelmente -> Lamentavel)
4. Remoção do aumentativo e diminutivo: Remoção dos sufixos que denotam aumentativo e diminutivo. (exemplo: Carrinho -> Carr)
5. Remoção de sufixos em nomes: Esta etapa conta com uma lista de 61 sufixos para substituição e adjetivos passiveis de remoção.
6. Remoção de vogais: Esta etapa consiste em remover a última vogal das palavras que não foram examinadas pelas etapas 4 e 5. (exemplo: Coluna -> Colun)
7. Remoção de sufixos em verbos: As formas verbais são reduzidas ao seu radical correspondente. (exemplo: vertebrado -> vertebr)
8. Remoção de acentos: Etapa necessária devido ao fato de que algumas variantes são acentuadas e outras não, como em psicólogo e psicologia. Após esta etapa, ambas as formas serão convertidas para psicolog.

2.4.3 Indexação

Oriundo do ramo de recuperação de informação (RI), as técnicas de indexação de informações visam fornecer procedimentos que possibilitem o armazenamento dos documentos de forma conveniente, visando a redução de tempo e esforços na busca do conteúdo no momento da análise (GOMES, 2009).

Os estudos em busca de meios de catalogação de informações textuais de forma automática são antigos e o avanço tecnológico proporciona ferramentas cada vez mais rápidas e precisas, tornando este processo mais produtivo (ARANHA, 2007). De acordo com Aranha (2007), as técnicas de mineração de textos necessitam de operações bem mais complexas do que a simples busca de palavras chaves em um texto. Para isso, as técnicas de indexação dentro da área de MT, promovem informações mais extensas, identificando fatos e padrões similares a aquelas obtidas pelas percepções humanas. Estes procedimentos atendem aos propósitos de categorização de documentos, identificação do significado semântico de expressões dentro do documento e a leitura de uma grande quantidade e arquivos textuais necessários para o rápido desempenho durante a etapa de análise. Em outras palavras, a indexação visa atribuir a cada conteúdo textual palavras e termos chaves que facilitaram a busca destes arquivos na base de dados.

2.4.4 Mineração

Segundo Gomes (2009), a fase de mineração tem como objetivo a extração de algum tipo conhecimento relevante dos conteúdos textuais. A escolha do algoritmo a ser utilizado durante esta etapa depende do objetivo específico da aplicação e do seu nível de complexidade e confiabilidade de acerto, não existindo um algoritmo ideal para todas as aplicações, tendo em vista as especificidades de cada cenário e suas variáveis. Uma das técnicas para aumentar a confiabilidade da mineração de dados é a utilização de vários classificadores em vez de um único classificador (ARANHA, 2007), técnica conhecida como *Ensemble*.

De acordo com Sharkey (1998), os sistemas de *Ensemble* ou multi-redes, proporcionam soluções para problemas que não podem ser resolvidos por sistemas de uma única tarefa ou que podem ser resolvidas de maneira mais eficaz por um sistema de multi-redes. Além disto, a utilização desta técnica permite maior confiabilidade em relação ao resultado final da etapa de mineração, pois o conceito de utilizar dois ou mais algoritmos para resolver um mesmo problema adiciona redundância ao sistema, tornando possível a desconsideração dos algoritmos que tenham realizado previsões incorretas (GIACOMEL, 2016). Ao final de uma operação com a utilização de *Ensemble*, algum método estatístico deve ser utilizado para definição do resultado final.

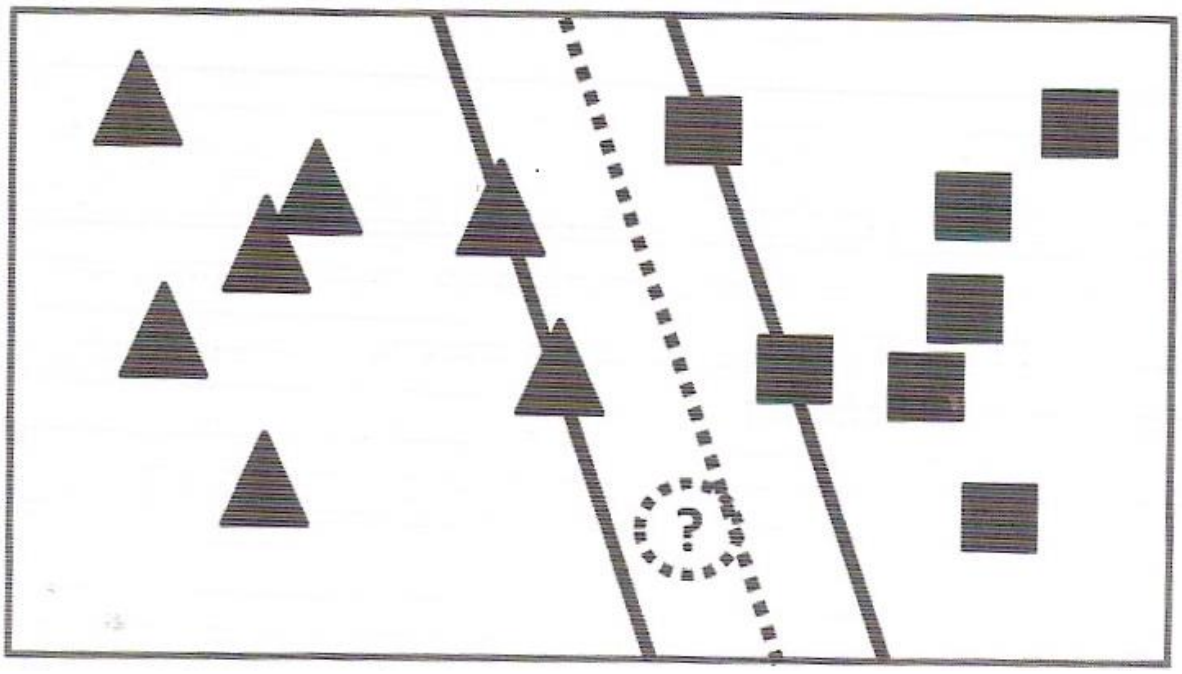
A literatura recente em torno da extração de conhecimento em textos da área da medicina apresenta a tendência de melhores resultados com a utilização de alguns algoritmos específicos. Nas seções seguintes, serão apresentadas estas técnicas e suas características.

2.4.4.1 *Máquina de Vetor de Suporte ou Support Vector Machine (SVM)*

De acordo com Amaral (2016), máquinas de vetor de suporte são algoritmos de classificação que aproximam as margens de uma instância a ser classificada com as instâncias mais próximas. O algoritmo de *support vector machine* apresenta grande capacidade de generalização e robustez, possibilitando sua aplicação em vetores de grandes dimensões (CARVALHO e LORENA, 2003).

Amaral (2016) apresenta a representação da classificação de um item a partir da utilização do SVM, conforme a Figura 8. São apresentados dois vetores de classificação, um para triângulos e outro para quadrados, representados na imagem por uma linha contínua. A classificação de um novo item (?) é efetuada através da criação de um novo vetor linear, representado na Figura por uma linha pontilhada, classificando o novo item como triângulo.

Figura 8 – Representação do algoritmo SVM



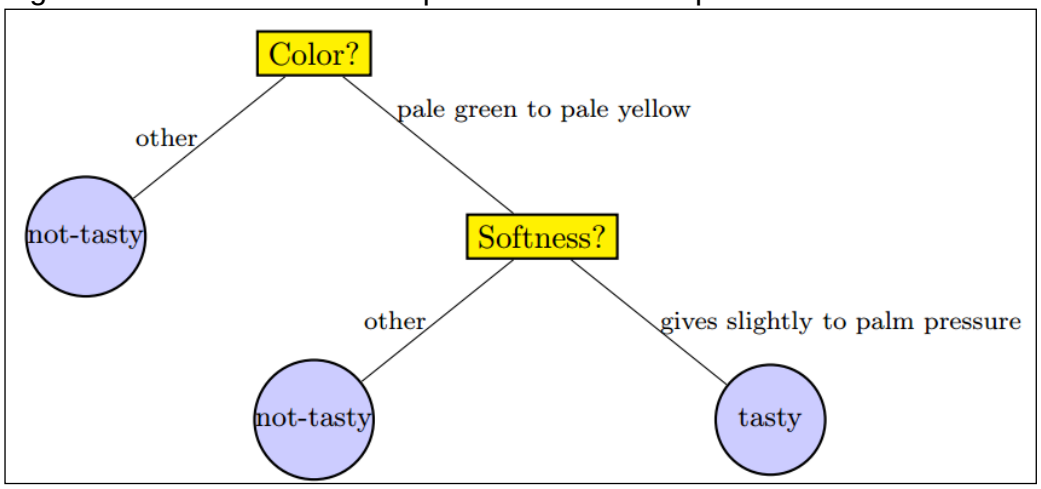
Fonte: Amaral (2016).

2.4.4.2 Árvores de decisão *ou Decision Tree Classifier*

O modelo de árvore de decisão se caracteriza por ser um modelo estatístico que possui um treinamento supervisionado para a classificação e previsão dos dados (DA SILVA, 2005). Uma árvore pode conter diversas ramificações, de acordo com a quantidade de atributos e valores existentes. Em uma árvore são executadas tarefas que visam identificar os atributos e os valores que fornecem a maioria das informações e remover aqueles raros, que não forneceram uma análise relevante (MICROSOFT, 2016). De acordo com Da Silva (2005), para a realização das partições e criações dos nodos da árvore, é avaliado a utilidade do atributo para a classificação, ou seja, qual o ganho de informação que cada atributo proporciona. O atributo escolhido como base é aquele que garante maior ganho de informação, e a partir deste, inicia-se um novo processo de partição.

Conforme a Figura 9, Shalev-Shwartz e Ben-David (2014), apresentam o exemplo de uma árvore de decisão para a determinação do sabor de uma fruta.

Figura 9 – Árvore de decisão para determinar a qualidade de um mamão



Fonte: Shalev-Shwartz e Ben-David (2014)

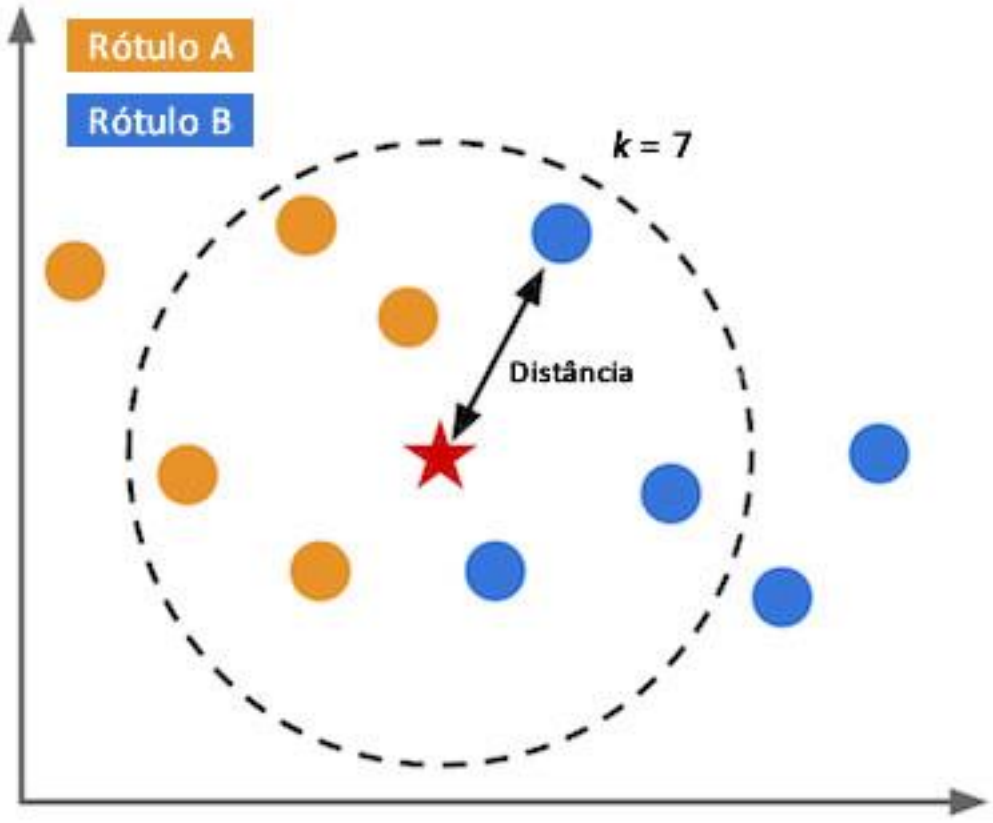
2.4.4.3 *K-Vizinhos mais próximos (K-Nearest Neighbor - KNN)*

O algoritmo k-vizinhos mais próximos é baseado na aprendizagem em instância, isto significa que os dados de treinamento são armazenados e a classificação de um novo item é realizado através da comparação entre as similaridades do item a ser classificado com os dos dados de teste (BUANI et al., 2009).

De acordo com Amaral (2016), a implementação do algoritmo KNN é realizado por meio da distância euclidiana, baseado no teorema de Pitágoras. Quando menor for a distância euclidiana entre duas instâncias, mais similaridades estas instâncias terão. Logo, quanto menor o valor da distância euclidiana maior a eficiência na classificação de uma nova instância (BROWER e ZAR, 1977 apud BORGES et al., 2007), que receberá como classificação o classificador do vizinho mais próximo. A parametrização da quantidade de vizinhos a serem considerados na análise é representando por K, que é muito menor a N, o tamanho da amostra (ALPAYDIN, 2014, p 190).

Na Figura 10, é apresentado por Pacheco (2017), a classificação de um novo item perante os 7 vizinhos mais próximos (k=7). Por existir 4 vizinhos do rótulo A e apenas 3 vizinhos do rótulo B a nova instância será classificada pelo rótulo A.

Figura 10 – Representação da classificação por meio do algoritmo KNN



Fonte: Pacheco (2017).

2.4.4.4 *Naïve Bayes*

O algoritmo *Naïve Bayes* é um algoritmo bayesiano, baseado na teoria das probabilidades, que assume que cada atributo influenciará de forma independente a classificação de uma nova instância (AMARAL, 2016, p 41).
Na abordagem *Naïve Bayes*, fazemos a suposição generativa (bastante ingênua) de que, dadas as características, estas são independentes uma das outras (SHALEV-SHWARTZ e BEN-DAVID, 2014, p 347).

Segundo Amaral (2016, p 41), durante a etapa de treinamento, é estabelecido uma tabela de valores e atribuído um peso para cada atributo em cada uma das classes de classificação. Ao submeter uma nova instância para classificação, o modelo somará os pesos de cada atributo em cada uma das classes, sendo que a classe que somar o maior peso será a classe classificadora do novo item.

2.4.5 Análise

Última etapa do processo de mineração de textos, os resultados finais gerados pelo processo todo devem ser interpretados por um analista. O analista deve verificar se o classificador atingiu as expectativas (ARANHA, 2007, p 59). Para Gomes (2009), é vantajoso a utilização de alguma forma de pós-processamento que permita a apresentação gráfica dos dados com possibilidade de navegação e iteração com os resultados, ajudando a fornecer um entendimento muito mais rápido e efetivo das conclusões geradas pela mineração.

2.5 MÉTRICAS DE AVALIAÇÃO

Para garantir a qualidade e coesão dos resultados gerados por um processo de classificação de textos, algumas métricas de avaliação devem ser analisadas. Conforme propostas por Alpaydin (2014), o Quadro 1 apresenta as principais métricas para avaliação do modelo de classificação treinado. Para representação das fórmulas são utilizadas algumas abreviações, exemplificadas abaixo:

a) VP: verdadeiros positivos, é uma instância que foi corretamente classificada como positiva;
b) VN: Verdadeiro negativo, é uma instância que foi corretamente classificada como negativa;
c) FP: Falso positivo, é uma instância que deve se negativa, mas foi classificada como positiva;
d) FN: Falso negativo, é uma instância que deve ser positiva, mas foi classificada como negativa.

Quadro 1 – Métricas de avaliação

Métrica	**Descrição**	**Fórmula**
Erros	Total de erros	(FP+FN) /Total
Acurácia	Total de acertos	1 - Erros
TP-Rate	Média de Positivos corretamente previstos	VP/(VP+FN)
FP-Rate	Média de Negativos corretamente previstos	FP/(FP+VN)
Precisão	Quantos registros de fato são positivos	VP/(VP+FP)
Sensitividade	Positivos corretamente previstos	TP-Rate
Especificidade	Negativos corretamente previstos	1 – (FP-Rate)

Fonte: Alpaydin (2014).

A obtenção destes valores é possível através da geração da Matriz de Confusão, após a aplicação do algoritmo de classificação. Matriz de confusão é uma técnica para resumir o desempenho de um algoritmo de classificação.

2.5.1 Considerações finais

Neste capítulo foram apresentados conceitos e técnicas utilizadas para a avaliação da qualidade de conteúdo textual e sobre mineração de textos.

Os conceitos relacionados à aprendizagem de máquina apresentados representam a base do escopo deste projeto, uma vez que deverão ser utilizados algoritmos de aprendizado de máquina supervisionado para a resolução do problema deste trabalho. A utilização de uma arquitetura *Ensemble* proporciona a possibilidade de utilização de diversos algoritmos de no processo de MT, garantindo maior coesão e confiabilidade nos resultados esperados.

O processo de MT apresentado neste capítulo propõe técnicas relevantes para a correta avaliação e classificação de conteúdos textuais perante a aplicação de algoritmos de aprendizado de máquina. Os algoritmos de aprendizado de máquina descritos neste capítulo são aqueles que apresentam maior coesão quando aplicados em conjuntos de dados textuais (WITTEN e FRANK, 2005) e que nos trabalhos relacionados à descoberta de conhecimento, citados na seção 2.2, em textos relacionados a saúde, se evidenciaram mais assertivos.

O Quadro 2 apresenta as técnicas de MT apresentadas e consideradas relevantes para este projeto, assim como sua respectiva etapa, de acordo os estudos e conceitos apresentados.

Quadro 2 – Técnicas de Mineração de Textos

Etapa	**Técnica**
Pré-processamento	Identificação de termos compostos
Pré-processamento	*Tokenização*
Pré-processamento	*Stemming*
Mineração	Máquina de vector de suporte
Mineração	Árvore de decisão
Mineração	K-Vizinhos mais próximos
Mineração	*Naïve Bayes*

Fonte: Próprio autor.

As técnicas e conceitos, acima evidenciados, fazem parte do escopo deste projeto, com isso, estas devem ser implementadas, testadas e validadas buscando atingir resultados satisfatórios quando a mineração e classificação de dados textuais relacionados a saúde.

3. IMPLEMENTAÇÃO E TESTES

O objetivo deste trabalho é a expansão e aperfeiçoamento de técnicas de aprendizagem de máquina em um software já existente, refinando a avaliação da qualidade textual em textos da área da saúde e visualização gráfica dos resultados. Este capítulo visa apresentar o software desenvolvido, destacando a sua arquitetura, método de pesquisa empregado e as ferramentas utilizadas. Nas seções seguintes são descritas as melhorias desenvolvidas, a fim de atingir objetivo proposto.

3.1 ANÁLISE DO SOFTWARE EXISTENTE

O software "*Minning*" foi desenvolvido com a finalidade de auxiliar a avaliação da qualidade das informações encontradas na internet na área da saúde (ABEL, 2016). A constatação da ausência de um sistema automatizado que avalia a apreensibilidade e qualidade de conteúdos textuais da área da saúde na língua portuguesa permitiu a idealização deste software, com o propósito de preenchimento desta lacuna. A aplicação foi desenvolvida utilizando o *framework ASP.NET*, uma plataforma desenvolvida pela *Microsoft* para o desenvolvimento de aplicações Web. Para a persistência dos dados empregou-se o *SQLServer*, também desenvolvido pela *Microsoft* e com fácil integração ao *ASP.NET*. O software *Weka* é utilizado para a tarefa de mineração de dados. O *Weka* é de domínio público, licença GNU (*General Public License*), e foi desenvolvido em *Java* por pesquisadores do Departamento de Computação da Universidade de Waikato da Nova Zelândia (BOUCKAERT, FRANK, et al., 2016). Para a visualização gráfica dos resultados gerados pelo "*Minning*", optou-se pelas bibliotecas *open source* D3.JS e JqPlot.

Em relação a apreensibilidade, em um primeiro momento foi selecionado a fórmula *Flesch Reading Ease,* por ser a mais utilizada nos estudos relacionados a apreensibilidade na área da saúde. Porém, os resultados obtidos por esta fórmula não foram considerados satisfatórios para a língua portuguesa, com isso foi selecionado a fórmula de *Fernadez-Huerta,* que apresentou resultados superiores para conteúdo em português.

Esta fórmula necessita das seguintes variáveis para mensurar a apreensibilidade de um conteúdo textual: (HUERTA, 1959 apud BARBOZA e NUNES, 2007):

a) A contagem de palavras
b) A contagem de frases
c) A contagem de sílabas

Os dois primeiros itens foram desenvolvidos diretamente no sistema. Já para a contagem de sílabas foi necessário a integração de uma ferramenta *open source*, tendo em vista a inexistência de uma forma universal de separação de sílabas e que cada idioma tem sua peculiaridade. Com todas as variáveis calculadas foi possível aplicar a fórmula de *Fernandez-Huerta*: L = 06.84 - (0.60 * P) - (1.02 * F). Onde L é a apreensibilidade, P a média de sílabas por palavra; F a média de palavras por frases (LAW, 2011).

Já para a avaliação da qualidade foram coletados 68 amostras textuais para treinamento de uma *Machine Learning*. Entre estas amostras estão contidos textos sobre “dor nas costas”, “coluna vertebral”, “hérnia de disco” e “cirurgias de coluna”. Com base nas diretrizes propostas pelo *DISCERN*[2], as amostras foram classificadas em “Negativas”, “Regulares” e “Positivas”. Com isso foram classificadas 39 amostras como “Negativas”, 22 amostras como “Regulares” e 7 amostras como “Positivas”. Antes de iniciar a avaliação dos textos foi necessário normalizá-los e estruturá-los. Para isso, as amostras de treinamento passaram por algumas técnicas de pré-processamento, tais como:

a) Conversão do conteúdo para minúsculas.
b) Remoção dos caracteres numéricos e especiais, como pontuação e outros mais específicos.
c) Substituição dos caracteres com acentuação pelo respectivo caractere sem acentuação, por exemplo “á” é substituído por “a”.
d) *Tokenization*, transformação de um texto em uma lista com todas as palavras. As palavras repetidas são adicionadas quantas vezes forem presentes no texto, pois indicam a frequência das mesmas no texto.
e) Remoção de *stopwords*.

[2] DISCERN é um breve questionário que fornece aos usuários uma maneira confiável de avaliar a qualidade da informação sobre as opções de tratamento para um problema de saúde CHARNOCK (2004).

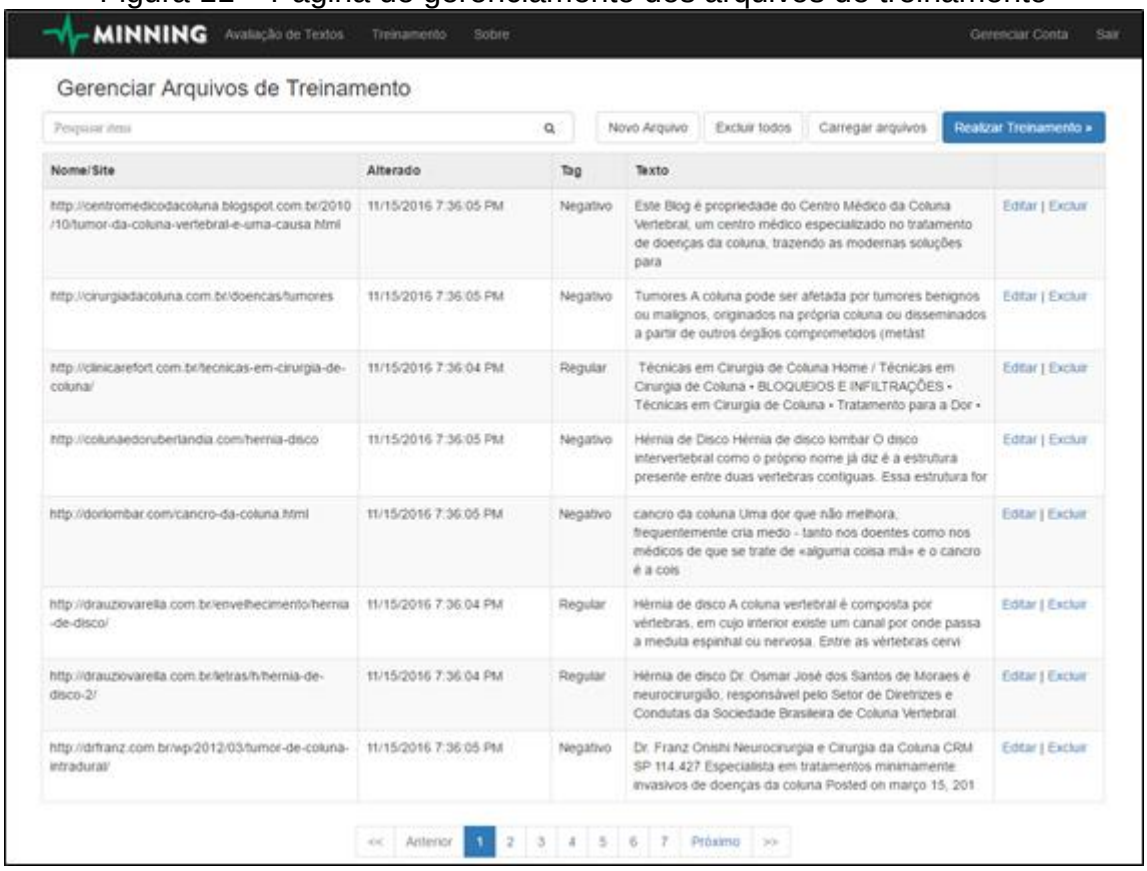
f) *Stemming*, em que as palavras são convertidas para os seus devidos radicais. Essa técnica foi obtida através da utilização de um pacote não oficial do *Weka,* PTStemmer. Essa é uma biblioteca de *stemmer* para a língua portuguesa, desenvolvida por Pedro Oliveira (WEKA, 2016).

Além disso foram aplicadas as técnicas de pré-processamento do *Weka.* A primeira técnica aplicada é a de *StringToWordVector*, que converte o texto em atributos e realiza a criação das classes de treinamento, correspondentes às classificações de “Negativos”, “Regulares” e “Positivos”. Posteriormente foi escolhido o filtro *InfoGainAttributeEval* que é responsável pela técnica de seleção dos termos mais relevantes (*AttributeSelection*), e também é essencial para reduzir o número de atributos, diminuindo a dimensionalidade do problema e eliminando palavras irrelevantes para o processo de classificação. Tendo os textos normalizados e pré-processados foi possível iniciar o treinamento da ML. Foram testados os algoritmos *Bayes Net, Naïve Bayes, Support Vector Machines - SVM, J48, Multilayer Perceptron, K-Nearest Neighbor - KNN* e selecionado aquele que apresentou a maior porcentagem de acertos em suas classificações. O algoritmo *Naïve Bayes* foi o que apresentou os melhores resultados, com uma porcentagem de acertos de 89,71%.

O software “*Minning*” conta com três funcionalidades centrais: treinamento, avaliação de textos e apresentação dos resultados.

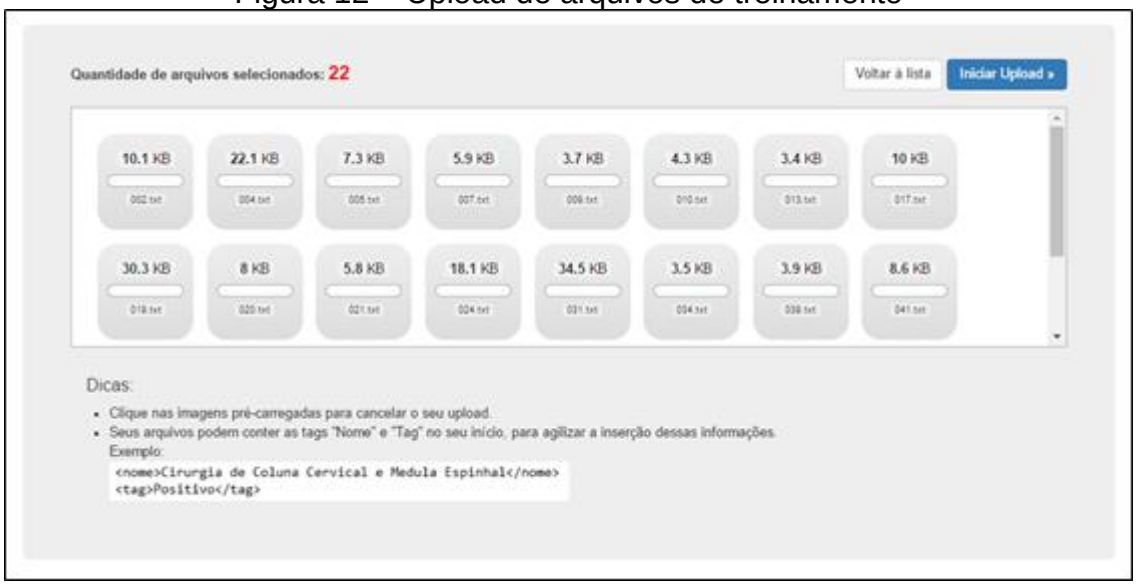
Na página de treinamento, representada pelas Figuras 11 e 12, que é disponível somente para usuários logados ao sistema através do botão “Treinamento” no menu superior, é possível carregar novos arquivos e/ou sites ao sistema. Para agilizar o processo de informar as classificações (tags) de vários arquivos, é possível inclusão de dois atributos dentro de cada arquivo: nome e tag (“Positivo”, “Regular” e “Negativo”) do arquivo. Estes novos arquivos passarão a fazer parte do banco de dados da aplicação e serão utilizados para o refinamento do algoritmo de reconhecimento da *machine learning*.

Figura 11 – Página de gerenciamento dos arquivos de treinamento



Fonte: Abel (2016)

Figura 12 – Upload de arquivos de treinamento

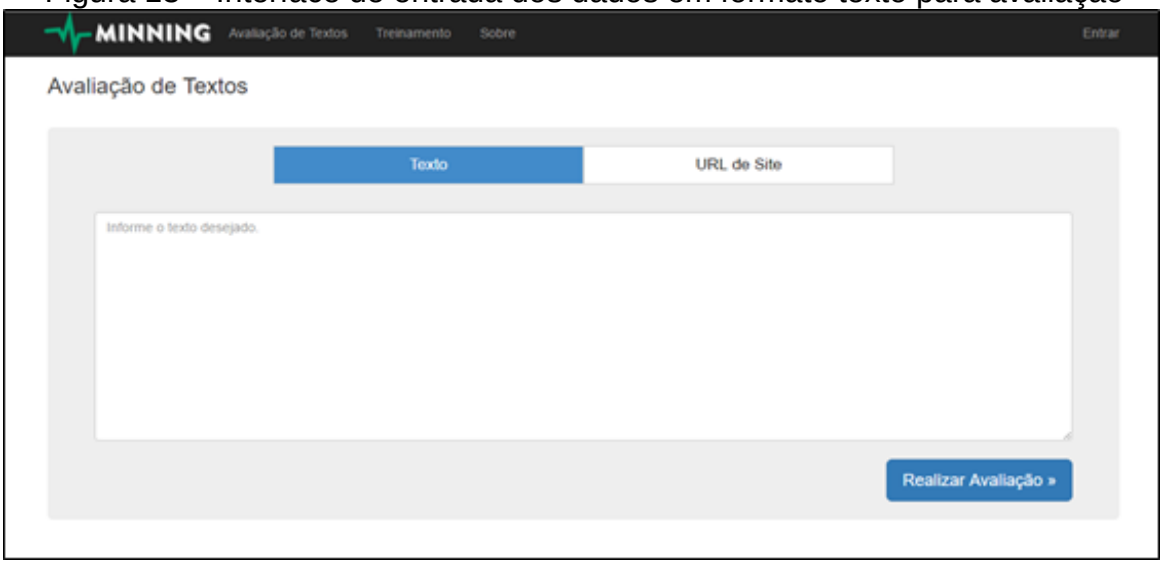


Fonte: Abel (2016)

Na página de avaliações de textos, apresentada pela Figura 13 e pela Figura 14, página inicial da aplicação e também podendo ser acessada pela opção “Avaliação de Textos” no menu, são exibidas para o usuário as opções de informar

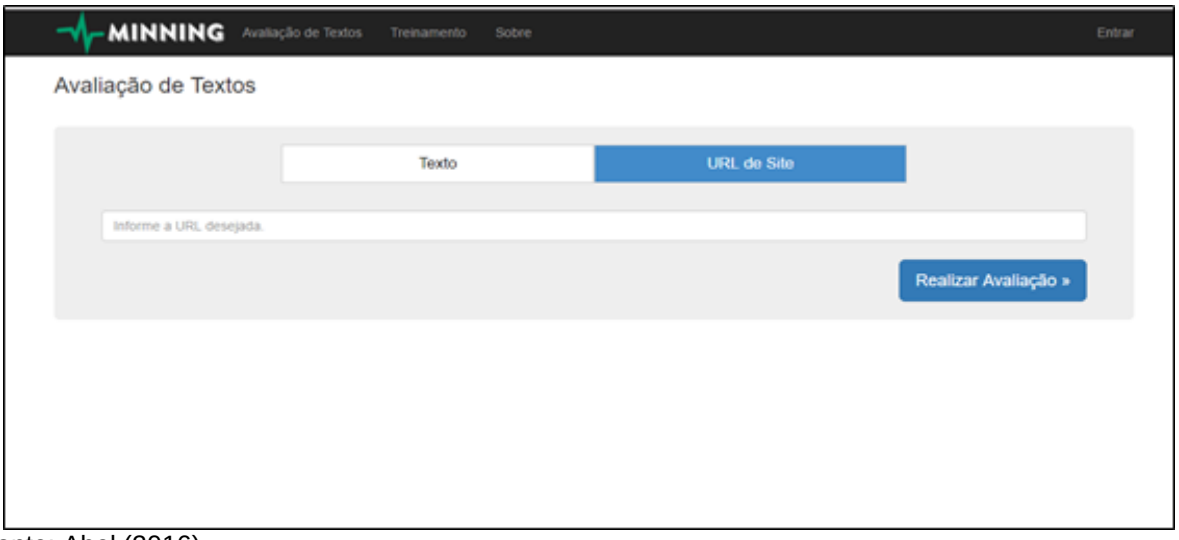
um texto ou a URL de um site. Após clicar em “Realizar Avaliação” o usuário é redirecionado para a página de apresentação dos resultados.

Figura 13 – Interface de entrada dos dados em formato texto para avaliação



Fonte: Abel (2016)

Figura 14 – Interface de entrada dos dados em formato URL para avaliação

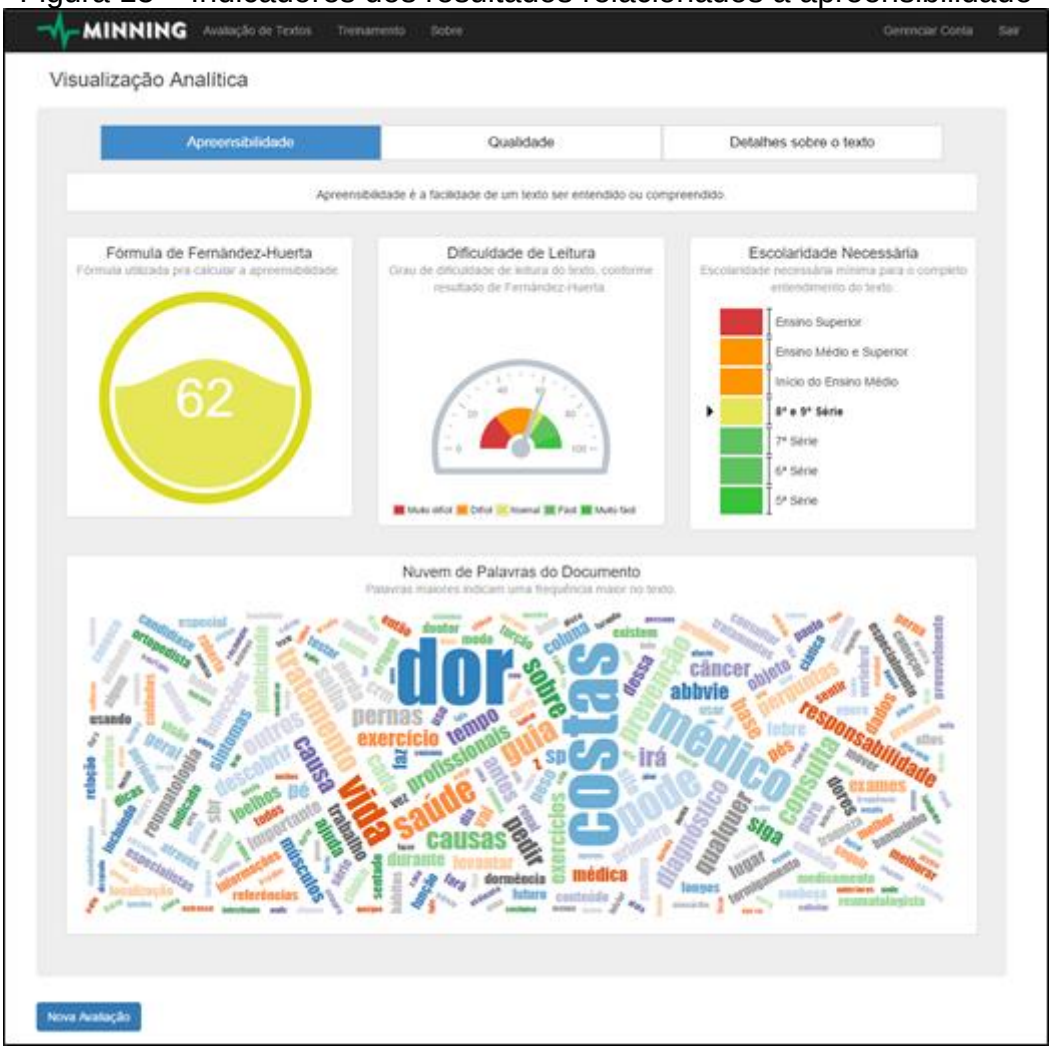


Fonte: Abel (2016)

A Figura 15 apresenta a página de visualização dos resultados, que foi construída em um formato *dashboard*, ou painel de indicadores, esta página é composta por três abas “Apreensibilidade”, “Qualidade” e “Detalhes sobre o texto”. A aba apreensibilidade, apresenta indicadores referentes a apreensibilidade com a dificuldade de leitura e a escolaridade necessária aproximada para entendimento do texto, de acordo com os resultados obtidos através da fórmula de *Fernandez-Huerta.*

Além de um gráfico de nuvem de palavras, apresentando as palavras mais frequentes do texto.

Figura 15 – Indicadores dos resultados relacionados a apreensibilidade

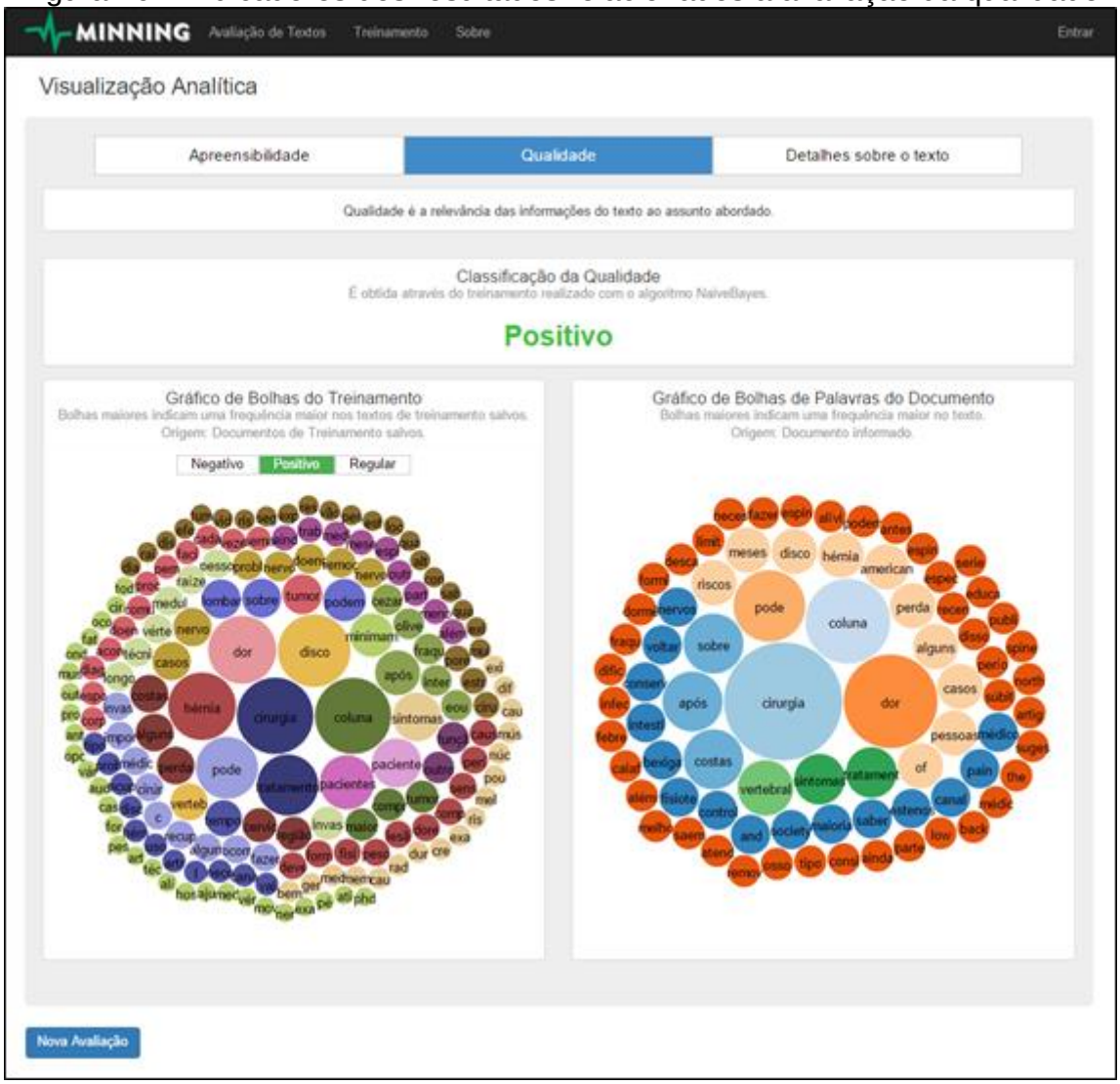


Fonte: Abel (2016)

Referente a aba qualidade, conforme Figura 16,o resultado da classificação é apresentado de maneira literal pelo sistema. Complementando os resultados são apresentados dois gráficos bolhas: o primeiro com as palavras mais frequentes dos arquivos do treinamento, separados pela classificação, o segundo apresenta as palavras mais frequentes do arquivo avaliado.

Na aba “Detalhes sobre o texto” são apresentados detalhes referentes ao arquivo analisado, como o nome/site do arquivo e o seu conteúdo.

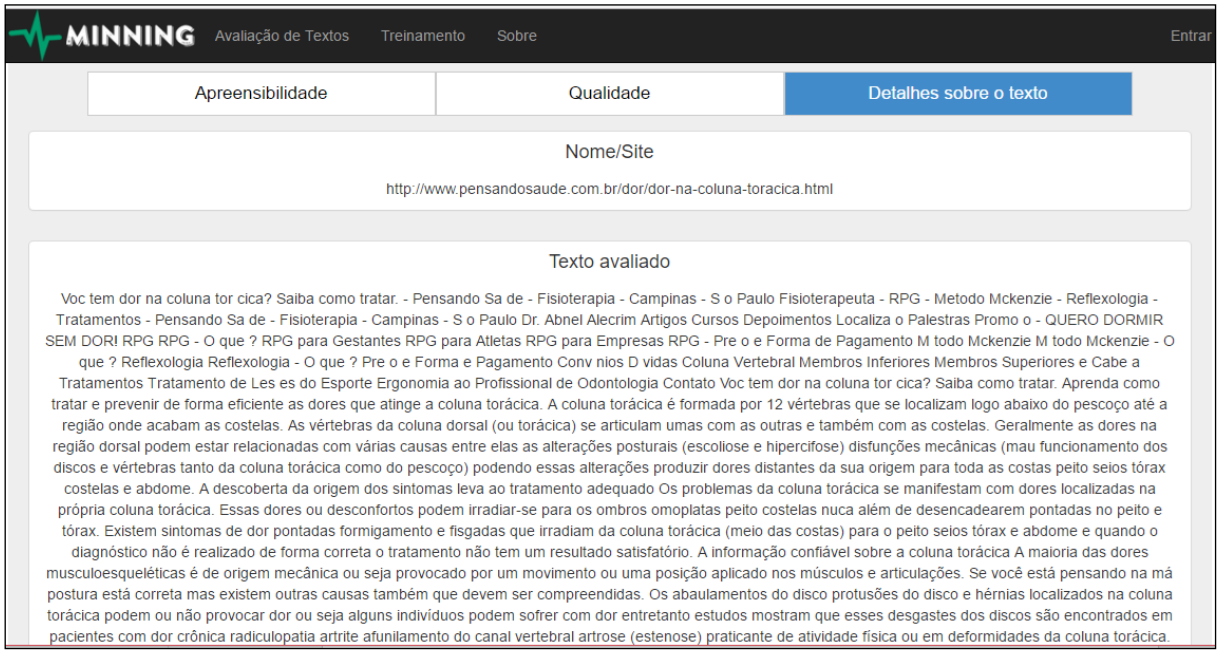
Figura 16 – Indicadores dos resultados relacionados a avaliação da qualidade



Fonte: Abel (2016)

Por fim, conforme a Figura 17, a aba “Detalhes sobre o texto” apresenta detalhes referentes ao arquivo analisado, como o nome/site do arquivo e o seu conteúdo.

Figura 17 – Informações apresentadas na aba “Detalhes sobre o texto”



Fonte: Abel (2016)

3.2 FERRAMENTAS E BIBLIOTECAS DE AUXÍLIO

Para a implementação das funcionalidades da ferramenta foram utilizadas as seguintes ferramentas: o framework ASP.NET, o banco de dados *SQL Server*, a implementação *Weka*, a biblioteca *IKVM.NET* e as bibliotecas de visualização de dados *D3.JS.*

3.2.1 ASP.NET

O ASP.NET foi a framework escolhida para o desenvolvimento da aplicação proposta. Este framework mantido pela Microsoft fornece todas as ferramentas necessárias para o desenvolvimento de uma ferramenta WEB.

A escolha do ASP.NET aconteceu, principalmente, devido ao fato de a primeira versão do sistema já ter sido desenvolvida através deste framework, além de ter um grande volume de documentações e possuir os recursos necessários para o desenvolvimento da aplicação.

3.2.2 SQL Server

Para a persistência de dados optou-se pela utilização do *SQL Server* da Microsoft. Devido ao fato de ser fornecido pela mesma empresa do *ASP.NET*, o *SQL* possui fácil integração e aderência a plataforma de desenvolvimento, garantindo maior produtividade quando comparado a outros bancos de dados.

3.2.3 Weka

O *Weka* é um software *open source,* de domínio público e licença GNU (General Public License), de aprendizado de máquina desenvolvido em Java e mantido pela Universidade de Waikato na Nova Zelândia (AMARAL, 2016). O *Weka* fornece ferramentas para a aplicação das técnicas de pré-processamento, classificação, regressão, agrupamento, associação e visualização.

O software WEKA foi escolhido devido ao fato de fornecer todos os algoritmos e técnicas necessárias para o desenvolvimento da aplicação, além de conter um bom número de técnicas que se aplicam a dados textuais e possuir recursos de teste e visualização dos resultados.

3.2.4 IKVM.NET

A interoperabilidade entre plataformas *Java* e *.NET* só é possível através da conversão dos arquivos com extensão *JAR*, extensão dos arquivos desenvolvidos em *JAVA*, para a extensão *DLL*, extensão reconhecida pela plataforma *ASP.NET*.

Para a realização desta operação de conversão foi integrado ao sistema a ferramenta, *open source*, *IKVM.NET*. Esta ferramenta garante que os arquivos com extensão JAR sejam interpretados e compilados pela plataforma *ASP.NET*.

Com a utilização desta ferramenta foi possível garantir a integração e utilização plena do *WEKA* ao projeto

3.2.5 D3.JS

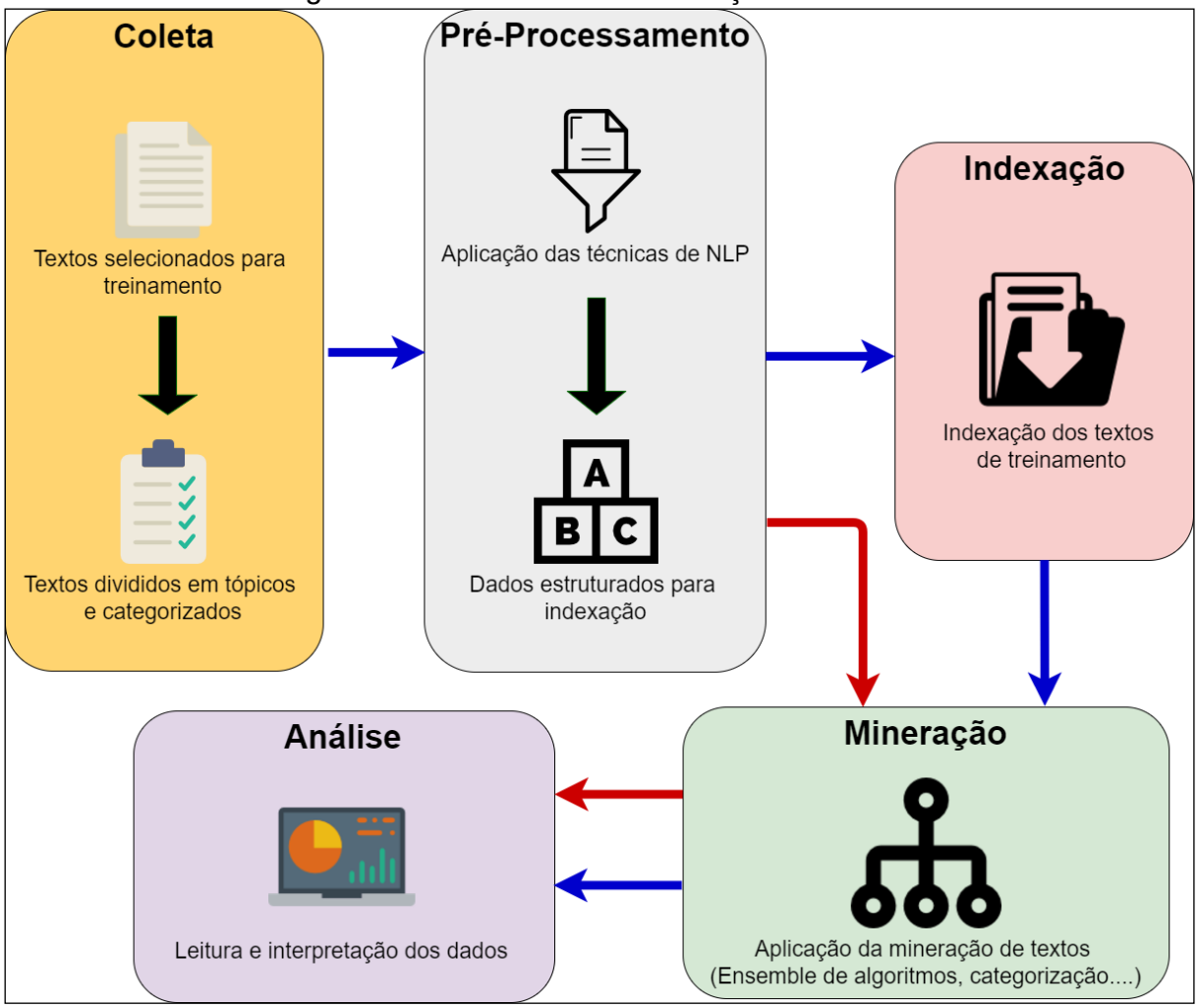
D3.js é uma biblioteca *JavaScript* para a visualização e manipulação de documentos baseada em dados. O D3 ou *Data Driven Documents* se caracteriza por

ser uma ferramenta *open source* que possibilita a criação de efeitos visuais complexos e desenhos interativos (BLEJMAN, 2013). Esta biblioteca possibilitou o desenvolvimento do componente *WordTree*, para a visualização do texto classificado pela ferramenta.

3.3 MÉTODO DE PESQUISA

Com base no processo de mineração de textos proposto por Aranha (2006), o modelo que foi incorporado ao projeto é apresentado pela Figura 18.

Figura 18 – Processo de mineração de textos



Fonte: Próprio autor.

Representado pelas setas azuis, o primeiro processo a ser executado é o treinamento dos algoritmos do sistema. O primeiro estágio deste processo foi realizado por especialistas da saúde. Os textos selecionados por estes foram

segmentados conforme as categorias definidas, além de serem classificados em “positivo”, “negativo” ou “regular”. Seguindo o processo, a próxima etapa executada foi a de pré-processamento. Nesta etapa foram aplicadas as técnicas apresentadas na seção 2.4.3, resultando ao final deste estágio em um conjunto de textos estruturados. Após a etapa de pré-processamento, a indexação dos arquivos estruturados foi realizada. Na etapa de mineração, composta pelo *Ensemble* de algoritmos de aprendizado de máquina, ocorreu o treinamento e aprendizado de cada um dos algoritmos do *Ensemble*.

O processo de avaliação de novos arquivos informados pelo usuário é evidenciado pelas setas vermelhas. Inicialmente o texto a ser avaliado passou pela etapa de pré-processamento e nesta etapa foram aplicadas as mesmas técnicas do processo de treinamento. Após esse estágio o texto passou para a etapa de mineração. Nesta fase o novo texto é avaliado por cada um dos algoritmos treinados. Por fim, a união de todos os resultados deve gerar o resultado final quanto a classificação do conteúdo.

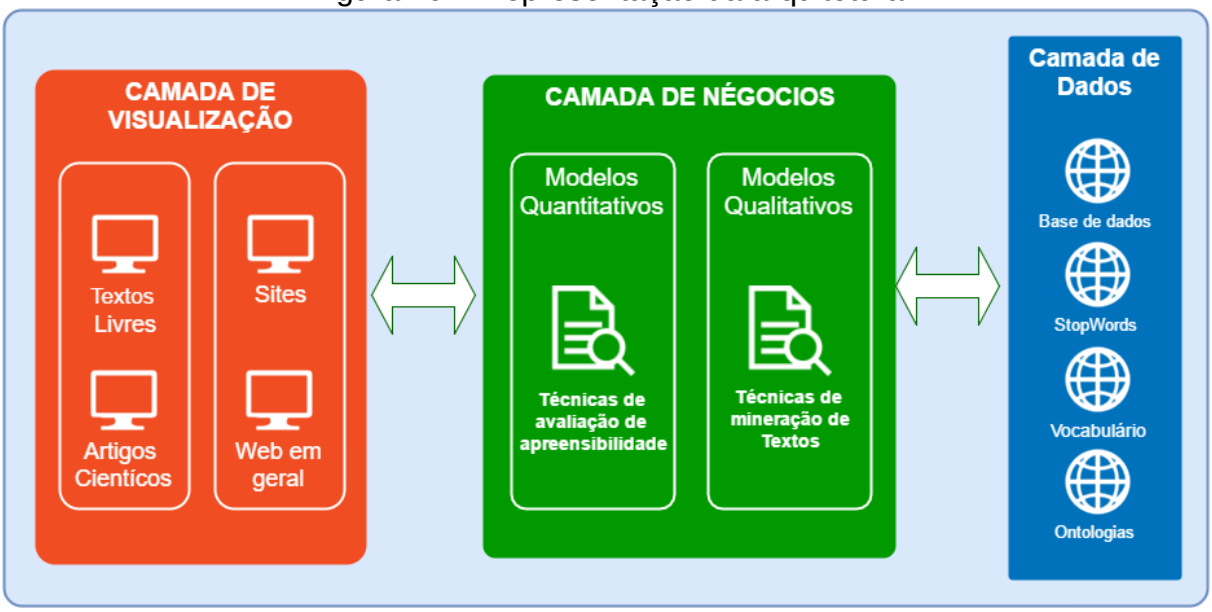
3.4 ARQUITETURA DO SOFTWARE

Em relação a arquitetura do software, optou-se pela manutenção da arquitetura desenvolvida anteriormente por Abel (2016). Dessa maneira, o padrão de arquitetura *3-Tier* (três camadas), será sustentado. O modelo divide-se nas camadas: camada de visualização, camada de negócios e camada de dados. A camada de visualização tem como objetivo armazenar a programação visual do sistema, a interface com a qual o usuário irá interagir. A camada de negócios é responsável pelas regras e validações a serem efetuadas durante uma operação do software, garantindo a integridade das informações. Por fim, a camada de dados é responsável pelo armazenamento das informações em um banco de dados.

As alterações propostas neste trabalho serão aplicadas nas camadas de visualização, com alteração da interface de resultados da avaliação da qualidade e na camada de negócios serão realizadas alterações nos modelos qualitativos.

A modelagem da arquitetura do projeto é apresenta pela Figura 19.

Figura 19 – Representação da arquitetura



Fonte: Próprio Autor.

3.5 MELHORIAS IMPLEMENTADAS

Visando obter resultados semelhantes ao de especialistas humanos para o processo de classificação automática de textos da área da saúde, são apresentadas nas seções seguintes alterações que abrangem a perspectiva “conteúdo”, através de implementações na etapa de mineração de textos que que respeitam as características de abrangência e acurácia, apresentada na seção 2.1. Assim como melhorias e interface, que abrangem a perspectiva “Design”, atendendo as características de usabilidade, rapidez e compatibilidade com diferentes navegados.

3.5.1 Melhorias no processo de mineração e classificação de textos

Participam deste projeto especialistas da área da saúde. Com eles foram discutidos e elencados pontos a serem estudados e alterados, garantindo melhor usabilidade e confiabilidade para a ferramenta.

As primeiras alterações a serem desenvolvidas foram em relação ao processo de mineração e classificação de textos da ferramenta. O sistema desenvolvido utiliza textos selecionados e classificados pelos especialistas da saúde para a realização do treinamento e aprendizado da *ML*. Aspirando agregar maior expertise ao sistema foi implementado um novo o processo de mineração, que conta com uma arquitetura *Ensemble.* Com auxílio dos especialistas da saúde foram

selecionados novos textos para a realização do treinamento. Nestes textos foram destacados e classificados trechos que correspondem às categorias definidas pelos especialistas como primordiais para a definição da classificação da qualidade de um texto da saúde. As categorias selecionadas pelos especialistas são as seguintes:

a) Descrição do tratamento;
b) Benefícios do tratamento;
c) Consequências do tratamento;
d) Influência na qualidade de vida do paciente;
e) Riscos do tratamento;

Desta forma cada categoria estabelecida pelos profissionais da saúde representa um algoritmo ou módulo da arquitetura *Ensemble* desenvolvida na ferramenta. Assim, ao realizar a avaliação de um novo texto todos os algoritmos são executados no respectivo arquivo. A união e soma destes resultados determinará o resultado final da avaliação e classificação do arquivo.

A fim de aperfeiçoar os resultados finais, alguns pontos da etapa de pré-processamento passaram por alterações. Inicialmente foi realizado o incremento de palavras na lista de *StopWords* do sistema. A lista de *StopWords* foi revisada e contou com o incremento de 289 palavras, totalizando 544 palavras. Além das técnicas já implementadas na ferramenta, apresentadas na seção anterior, foi incorporado à etapa de pré-processamento do *SpineFind* uma lista de termos compostos. A lista elaborada, nomeada de "*TermsHandler"* na ferramenta *SpineFind*, pelos especialistas da saúde conta com 35 termos considerados relevantes para a análise textual em textos da saúde. Antes da operação da separação dos *Tokens* é verificado se o texto contém algum dos termos compostos da lista, caso seja encontrado algum termo este é substituído por uma "*Key*" que é gerada pelo sistema e não será separada durante o processo de *Tokenization*. Após a separação dos *Tokens*, todas as "*Keys*" são novamente substituídas pelos termos correspondentes, dando origem a um *Token* com o termo composto.

Uma amostra desta operação é apresentada pelo Algoritmo 1.

Algoritmo 1 – Substituição do termo composto pela *Key*

```
public List<Token> Tokenization(string texto)
        {
            var lista = new List<Token>();

            //Procura por algum dos "termos compostos" no texto
            //caso encontre, substitui pela "key" correspondente
            foreach (var item in TermsHandler.Termos)
            {
                if (texto.Contains(item.Value))
                    texto = texto.Replace(item.Value, item.Key);
            }

            //Quebra o texto pelo delimitadores da língua portuguesa
            string[] textoArray = texto.Split(Delimitadores(),
StringSplitOptions.RemoveEmptyEntries);

            // Percorre palavra por palavra no texto
            foreach (string palavra in textoArray)
            {
                int num;
                bool numerico = int.TryParse(palavra, out num);
                if (!numerico)
                {
                    //Caso o "palavra" seja alguma das "keys" da lista de termos
                    //cria um token com o termo original
                    if (TermsHandler.Termos.ContainsKey(palavra))
                        lista.Add(new Token(TermsHandler.Termos[palavra]));
                    else
                        lista.Add(new Token(palavra));
                }
            }
            return lista;
        }
```

Fonte: Próprio autor.

Visando garantir desempenho ao sistema, os termos existentes na ferramenta passaram pelos seguintes processos da etapa de pré-processamento:

a) Conversão do conteúdo para minúsculas.
b) Remoção dos caracteres numéricos e especiais, como pontuação e outros mais específicos.
c) Substituição dos caracteres com acentuação pelo respectivo caractere sem acentuação, por exemplo “á” é substituído por “a”.

O Quadro 3 apresenta os termos compostos elaborados pelos especialistas humanos e incorporados na ferramenta *SpineFind,* assim como as respectivas “*Keys*” que estão representadas no Algoritmo 1.

Quadro 3 – Termos compostos utilizados na ferramenta

Key	**Termo composto**
tratamentoclinico	tratamento clinico
tratamentocirurgico	tratamento cirurgico
tratamentomedicamentoso	tratamento medicamentoso
herniadedisco	hernia de disco
herniacervical	hernia cervical
dornascostas	dor nas costas
dorlombar	dor lombar
hernialombar	hernia lombar
tumordemedula	tumor de medula
escoliosetoracica	escoliose toracica
escolioselombar	escoliose lombar
dornaperna	dor na perna
dorcervical	dor cervical
dornopescoco	dor no pescoco
tipodecirurgia	tipo de cirurgia
complicacoescirurgicas	complicacoes cirurgicas
riscobeneficio	risco beneficio
placaeparafuso	placa e parafuso
fixacaointerna	fixacao interna
perdadeforca	perda de forca
perdasensitiva	perda sensitiva
perdadesensibilidade	perda de sensibilidade
canalestreitolombar	canal estreito lombar
doencadegenerativa	doenca degenerativa
degeneracaodedisco	degeneracao de disco
tumormedular	tumor medular
compressãomedular	compressão medular
compressaonervosa	compressao nervosa
raiznervosa	raiz nervosa
cirurgiadecoluna	cirurgia de coluna
doencadecoluna	doenca de coluna

discointervertebral	disco intervertebral
vertebrascervicais	vertebras cervicais
vertebrastoracicas	vertebras toracicas
vertebraslombares	vertebras lombares
fraquezamuscular	fraqueza muscular

Fonte: Próprio autor.

Conforme citado na seção 3.1, o sistema utiliza o algoritmo *Naïve Bayes* para realizar a classificação dos textos anexados para avaliação. Com a incorporação da nova arquitetura de mineração de textos, os algoritmos citados na seção 2.4.4 foram investigados e testados para cada uma das categorias elencadas pelos especialistas. O Quadro 4 apresenta as técnicas e algoritmos incorporados ao sistema.

Quadro 4 – Técnicas e algoritmos de aprendizado de máquina para *text mining*

Técnica	**Algoritmo**
Árvores de Decisão	J48
Máquinas de Vetores de Suporte	*Support Vector Machines* – SVM
K-Vizinhos mais próximos	*K-Nearest Neighbor* – KNN
Classificador Bayesiano Ingênuo	*Naïve Bayes*

Fonte: Próprio autor.

Com o propósito de tornar o processo de treinamento dos algoritmos o mais genérico possível, as operações comuns entre todos os algoritmos foram abstraídas para uma classe específica, nomeada “Algoritmo”. Esta classe conta com método designado “RealizaTreinamento”, este é responsável de realizar o treinamento do algoritmo e gerar o classificador a ser utilizado na classificação de um novo texto. Este método tem como parâmetro um enumerador que representa as categorias possíveis, indicando qual categoria está sendo treinada. Outro método presente nesta classe é o “SetClassificador”, este é um método abstrato que deve ser sobescrito por todas as classes que realizam herança da classe “Algoritmo”.

O Algoritmo 2 apresenta os métodos existentes na classe Algoritmo.

Algoritmo 2 – Métodos existentes na classe Algoritmo

```
        public void RealizaTreinamentoWeka(enumCategorias categoria)
        {
            var diretorioDadosServidor = string.Format("{0}\\{1}",
DiretorioDadosServidor, categoria.Key());
            Instances dadosTreinamento =
ImportaArquivosServidor(diretorioDadosServidor);

            // Aplica o filter StringToWordVector
            weka.filters.Filter[] filters = new weka.filters.Filter[2];
            var stringVector = new StringToWordVector();

            filters[0] = new StringToWordVector();
            filters[1] = AttributeSelectionFilter(2);

            weka.filters.MultiFilter filter = new weka.filters.MultiFilter();
            filter.setInputFormat(dadosTreinamento);
            filter.setFilters(filters);

            // Cria o Classificador a partir dos dados de treinamento
            FilteredClassifier classifier = new FilteredClassifier();
            classifier.setFilter(filter);

            //Selecionado o algoritmo de acordo com a classe "filha"
            SetClassificador(classifier);
            classifier.buildClassifier(dadosTreinamento);


            var diretorioClassificadorServidor = string.Format("{0}\\{1}\\",
DiretorioClassificadorServidor, categoria.Key());
            if (!File.Exists(diretorioClassificadorServidor))
                Directory.CreateDirectory(diretorioClassificadorServidor);

            // Salva o Classificador (model) como
/Classificador/Classificador.model
            SerializationHelper.write(string.Format("{0}Classificador.model",
diretorioClassificadorServidor), classifier);

            // Salva os dados de treinamento como
/Classificador/DadosTreinamento.arff
            Utilidades.SalvarArff(dadosTreinamento,
diretorioClassificadorServidor, "DadosTreinamento.arff");
        }
protected abstract void SetClassificador(FilteredClassifier classifier);
```

Fonte: Próprio autor.

Além desta classe abstrata foram desenvolvidas classes representativas para cada uma das categorias. Estas classes realizam a herança da classe “Algoritmo”, com isso a classe específica só precisa sob escrever o método de escolha do algoritmo (“SetClassificador”), escolhendo o algoritmo corresponde a classe. O Algoritmo 3, demonstra um exemplo da classe Algoritmo J48.

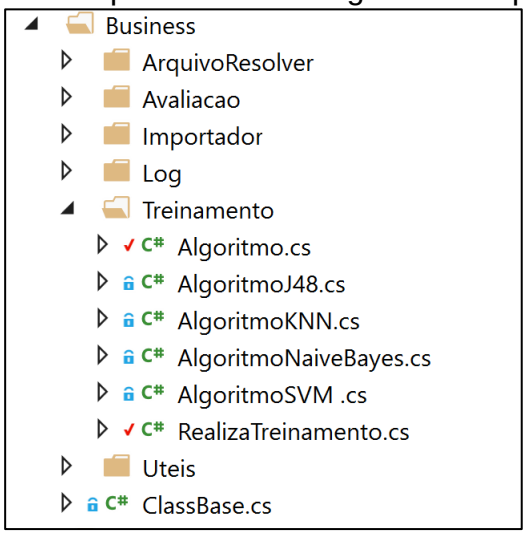
Algoritmo 3 – Exemplo das classes específicas de cada algoritmo

```
public class AlgoritmoJ48 : Algoritmo
  {
      protected override void SetClassificador(FilteredClassifier classifier)
      {
          classifier.setClassifier(new J48());
      }
  }
```

Fonte: Próprio autor.

A Figura 20 apresenta todas classes, representando diferentes algoritmos, desenvolvidas e incorporadas ao sistema, onde cada classe contém a chamada ao algoritmo de aprendizado de máquina existente no Weka.

Figura 20 – Classes representado os algoritmos implementados



Fonte: Próprio autor.

Complementando o processo de treinamento foi implementado a classe responsável por acionar o treinamento do algoritmo. A classe denominada “RealizaTreinamento” conta um método chamado “Treinar” que é encarregado por recuperar os arquivos de treinamento do banco de dados e realizar o treinamento do algoritmo vinculado a categoria. O Algoritmo 4 apresenta a implementação deste método.

Algoritmo 4 – Processo de classificação de uma nova instância

```
public ResultadoTreinamento Treinar()
        {
            //Recupera arquivos para a realização do treinamento
            List<Arquivo> _arquivos = new List<Arquivo>();
            using (var conexao = new SiteDB())
            {
                _arquivos.AddRange(conexao.Arquivos);
```

```
            }

            ResultadoTreinamento resultado = new ResultadoTreinamento();

            //Realiza o treinamento para cada uma das categorias do sistema
            foreach (enumCategorias item in
Enum.GetValues(typeof(enumCategorias)))
            {
                var arquivosCategoria = _arquivos.Where(a => a.Categoria ==
(int)item).ToList();
                if (arquivosCategoria.Count > 0)
                {
                    ExportaArquivosServidor(DiretorioDadosServidor, item.Key(),
arquivosCategoria);

                    //Retorna o algoritmo utilizado para a respectiva categoria
                    Algoritmo algoritmo =
GetAlgotimoUtilizado(item.AlgoritmoUtilizado());

                    //Realiza o treinamento do algoritmo
                    algoritmo.RealizaTreinamentoWeka(item);
                    resultado.CategoriasTreinadas.Add(item);
                }
                else
                    resultado.CategoriasNaoTreinadas.Add(item);
            }

            return resultado;
        }
```

Fonte: Próprio autor.

O vínculo da categoria com o algoritmo utilizado é realizado através do enumerador que representa as categorias no sistema e do enumerador que representa os algoritmos disponíveis pela ferramenta, conforme apresentado pelo Algoritmo 5.

Algoritmo 5 – Vinculo entre categoria e o algoritmo

```
        public static enumAlgoritmo AlgoritmoUtilizado(this enumCategorias value)
        {
            switch (value)
            {
                case enumCategorias.Beneficios:
                    return enumAlgoritmo.NaiveBayes;
                case enumCategorias.Consequencias:
                    return enumAlgoritmo.SVM;
                case enumCategorias.DescricaoTratamento:
                    return enumAlgoritmo.NaiveBayes;
                case enumCategorias.Influencias:
                    return enumAlgoritmo.NaiveBayes;
                case enumCategorias.Riscos:
                    return enumAlgoritmo.NaiveBayes;
                default:
                    return enumAlgoritmo.NaiveBayes;
            }
        }
```

Fonte: Próprio autor.

Com a utilização desta abordagem foi possível eliminar duplicações de códigos e facilitar futuras manutenções ao processo de mineração de textos da ferramenta.

Após a realização do treinamento, os classificadores são mantidos em pastas no servidor para posterior utilização no momento de realizar a classificação de um novo texto. A Figura 21 apresenta a estrutura de armazenamento adotada para indexação destes arquivos classificadores.

Figura 21 – Estrutura de armazenamento dos classificadores por categoria

Fonte: Próprio autor.

O processo de classificação de novas instâncias precisou ser alterado para que atendesse a demanda da arquitetura *Ensemble* implementada, uma vez que todos os algoritmos treinados serão aplicados no novo texto a ser classificado, apresentando os resultados de acordo com o treinamento já realizado.

A mesma abordagem de abstrair e tornar os processos genéricos que foi adotado para a fase de treinamento foi utilizada nesta etapa. Baseado neste cenário foi implementado a classe “RealizaAvaliacao”. Esta classe conta um método público “AvaliarTexto” que recebe os parâmetros “texto” e “nome” e em cima destes parâmetros são aplicados os mesmos métodos de pré-processamento da etapa de treinamento, a fim de normalizar o texto para posterior classificação. Após a aplicação das técnicas de pré-processamento, este método executa um laço de repetição no enumerador que representa as categorias possíveis e para cada categoria é executado o método que retornará a classificação do texto para a categoria. Este método responsável por realizar a classificação da nova instância, recupera o classificador da respectiva categoria e realiza a classificação com base neste arquivo. O algoritmo 6 apresenta o processo descrito anteriormente.

Algoritmo 6 – Processo de classificação de uma nova instância

```
public ResultadoAvaliacao AvaliaTexto(string texto, string nome)
{
  ResultadoAvaliacao resultado = new ResultadoAvaliacao(new Arquivo(texto, nome));

  //Aplicação das funcões de pré-processamento
  resultado.Arquivo.InicializaArquivo();


 //Realiza a classificação para cada uma das categorias
  foreach (enumCategorias item in Enum.GetValues(typeof(enumCategorias)))
  {
    try
    {
     resultado.Resultados.Add(new Classificacao(item,
RealizaClassificacaoWeka(resultado.Arquivo, item)));
    }
    catch (Exception)
    {
     resultado.Resultados.Add(new Classificacao(item, null));
    }

  }
  return resultado;
}
```

Fonte: Próprio autor.

3.5.2 Melhorias de interface

Avaliado junto aos especialistas da saúde foi decidido pela modificação do nome e do logo da aplicação, visto a especialização do sistema na avaliação de problemas referentes a coluna. Assim, optou-se pelo nome "*SpineFind*". A figura 22 ilustra o logo criado.

Figura 22 – Logo do criado para o sistema *SpineFind*

Fonte: Próprio autor.

Devido às alterações realizadas na etapa de classificação de textos a apresentação dos resultados também passou por uma reformulação, com o propósito de apresentar os resultados das classificações de todas as categorias e o

resultado da classificação final do texto avaliado. Visando manter uma interface de fácil entendimento ao usuário optou-se pela incorporação de um “*checklist*” que é responsável por apresentar o resultado da avaliação de cada uma das categorias.

A Figura 23 apresenta o resultado final da interface de resultados da classificação de textos. Já a Quadro 5 visa apresentar os componentes existentes na interface de resultados bem como suas funcionalidades.

Figura 23 – Interface de resultados da avaliação da qualidade

Fonte: Próprio autor.

Quadro 5 – Componentes da tela de resultados da avaliação da qualidade

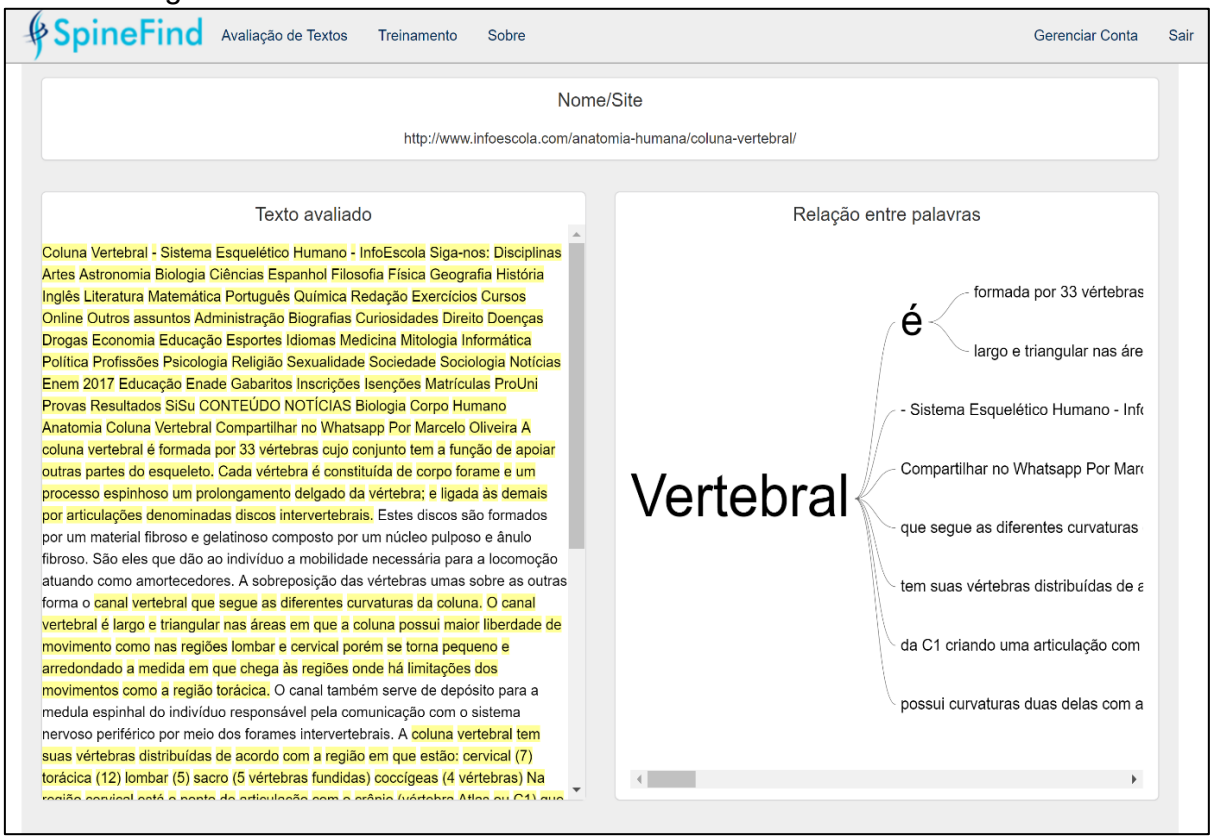
Componente	**Resultados da Qualidade**
Descrição	Exibe os resultados da avaliação textual sobre a qualidade textual, utilizando as técnicas de Mineração de Textos
Funcionalidades	Composto por: a) Qualidade do texto de acordo com classificador, apresentando os resultados para: Positivo, Regular ou Negativo. *b)* *Checklist* apresentando o resultado da avaliação de cada uma das categorias. *c)* Resultado detalhado da classificação, apresentando a

	classificação do texto em cada uma das categorias.

Fonte: Próprio autor.

Outro aperfeiçoamento relacionado a interface do sistema foi a integração do componente *WordTree*. Este componente foi incorporado a aba “Detalhes sobre o texto” ao lado do componente de visualização do texto classificado, com o propósito de apresentar uma visão analítica entre a relação de todas as palavras do texto. Este componente permite a iteração do usuário junto ao texto. Ao clicar em uma palavra no componente “Texto avaliado” são apresentandos através do *WordTree* todas as relações desta palavra no texto. A Figura 24 apresenta o resultado final da incorporação deste componente, exemplificando a relação da palavra “Vertebral” no texto avaliado.

Figura 24 – Interface de “Detalhes sobre o Textos” com *WordTree*



Fonte: Próprio autor.

O Quadro 6 apresenta os componentes existentes na aba “Detalhes sobre o texto” assim como suas funcionalidades.

Quadro 6 - Componentes da tela de detalhes do texto

Componente	**Detalhes do texto**
Descrição	Exibe os resultados da avaliação textual sobre a qualidade textual, utilizando as técnicas de Mineração de Textos
Funcionalidades	Composto por: a) Apresentação do título do conteúdo avaliado. *b)* Apresentação do texto avaliado. *c)* Componente *Word Tree,* apresentando a relação entre as palavras presentes no texto.

Fonte: Próprio autor.

3.5.3 Hospedagem

Para a hospedagem da ferramenta foi utilizado o site *Somee.com*, que possui um plano gratuito de hospedagem para aplicações *ASP.NET*. O plano gratuito conta com 150MB de armazenamento, além de disponibilizar 15MB para armazenamento de dados através do SQL Server. O site oferece acesso FTP a todos os diretórios do servidor, sendo possível desta maneira realizar as publicações necessárias da ferramenta.

O link para acesso ao site é http://www.spinefind.somee.com/.

3.6 TESTES E AVALIAÇÕES DE RESULTADOS

Para a realização do treinamento e avaliação dos algoritmos implementados no *SpineFind* foi indispensável o auxílio dos especialistas da saúde, uma vez que a base de conteúdo textual a sustentar todo o aprendizado do sistema precisou ser revisada. Desta maneira foram disponibilizados novos textos a serem utilizados nesta fase. Estes textos continham trechos destacados representando cada uma das categorias. Com base nestes trechos foi possível montar o *corpus* a ser utilizado posteriormente na avaliação e escolha do algoritmo com melhor resultado para cada uma das categorias.

Para cada categoria foi realizado o treinamento utilizando os algoritmos: *Naïve Bayes, Support Vector Machines* (SVM), *K-Nearest Neighbor* (KNN) e J48.

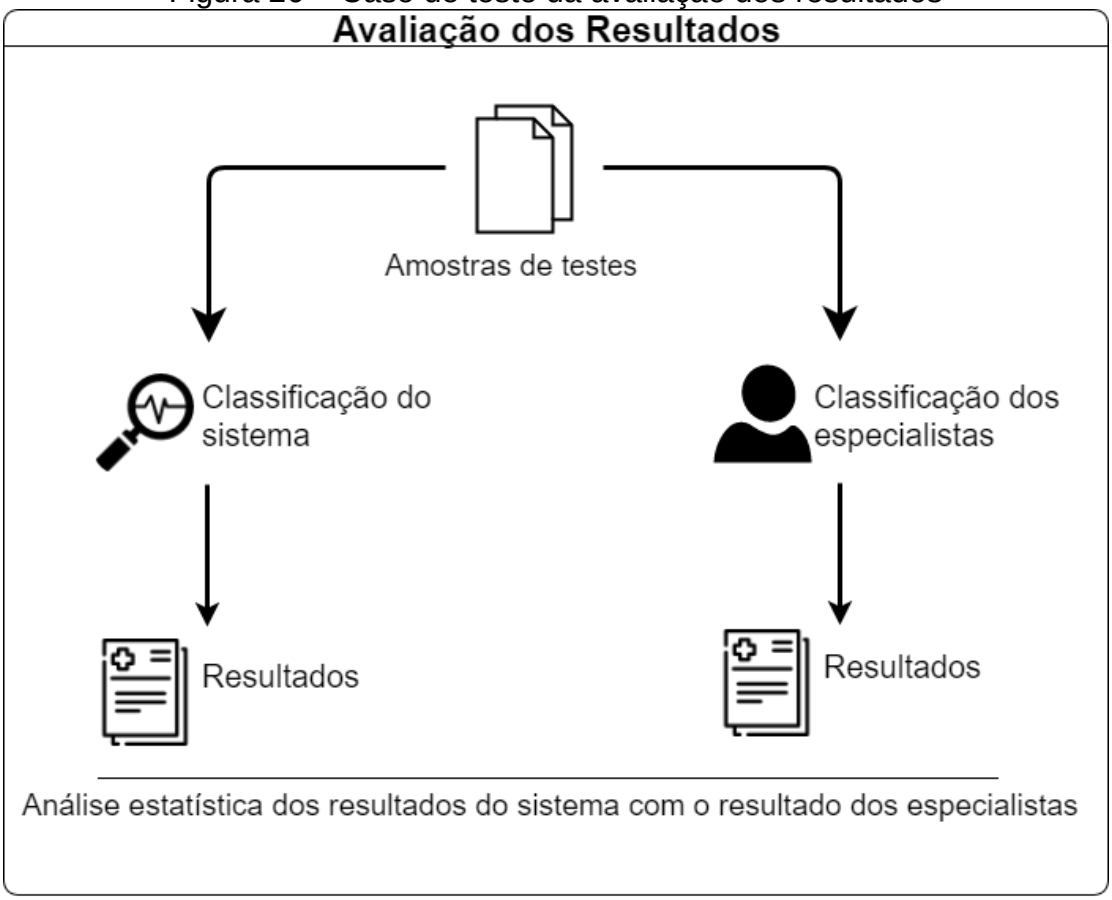
Para a determinação do algoritmo a ser utilizado na versão final da ferramenta foram realizadas comparações entre as classificações dos textos pelos especialistas com as classificações realizadas pelo sistema. Esta abordagem foi dívida em duas etapas. A Figura 25 apresenta a primeira etapa, em que o software passou por um treinamento de acordo com os textos/trechos selecionados e classificados pelos especialistas. Este processo resultou no classificador de cada categoria, que passou a ser utilizado na classificação de um novo texto.

Figura 25 – Treinamento do sistema

Fonte: Próprio autor

Já a Figura 26 evidencia o processo de validação e testes. Neste processo as amostras selecionadas para testes foram classificadas pelo sistema e estes resultados comparados a classificação dos especialistas humanos, garantindo a coesão das respostas geradas pelo sistema. Devido ao pequeno número de textos disponíveis, todos os textos foram utilizados tanto no treinamento, como nos testes. Por fim, para os algoritmos que apresentaram a maior taxa de convergência nestas comparações foram avaliadas as métricas de performance citadas na seção 2.5.

Figura 26 – Caso de teste da avaliação dos resultados



Fonte: Próprio autor.

Abaixo são apresentados os resultados obtidos para cada categoria.

3.6.1 Descrição do tratamento

Esta foi a categoria com o maior número de textos para treinamento da *ML* do *SpineFind*. Com um total de 79 textos selecionados pelos especialistas e que compuseram o *corpus* desta categoria. A Tabela 1 apresenta a quantidade de textos existentes para cada uma das classificações.

Tabela 1 – Quantidade de texto por classificação

Classificação	Quantidade de textos
Negativo	58
Positivo	14
Regular	7

Fonte: Próprio autor.

Visando obter resultados semelhantes aos definidos pelos especialistas foi realizado a comparação entre as classificações realizadas pelos especialistas com a classificação obtida após a *ML* estar devidamente treinada, com cada um dos algoritmos. A Tabela 2 apresenta o resultado final destas comparações.

Tabela 2 – Comparação entre Especialistas x Algoritmos

Algoritmos	Acertos	Erros	% Acerto
Naïve Bayes	75	4	94,94%
SVM	77	2	97,47%
KNN	62	17	78,48%
J48	68	11	86,07%

Fonte: Próprio autor.

Por meio do *Weka* foi possível obter a Matriz de Confusão do algoritmo que resultou na melhor taxa de conversão quando comparado a classificação realizada por um especialista humano, no caso desta categoria foi o algoritmo SVM. A Tabela 3 apresenta esta Matriz de Confusão.

Tabela 3 – Matriz de confusão do algoritmo *Naïve Bayes*

A	B	C	
58	0	0	A = Negativo
0	14	0	B = Positivo
2	0	5	C = Regular

Fonte: Próprio autor.

A partir da análise da matriz de confusão demonstrada na Tabela 3, nota-se que dois textos regulares foram classificados como negativos.

Com base nos valores adquiridos pela Matriz de Confusão foram aplicadas as métricas propostas e obtido a performance e os detalhes da utilização do algoritmo SVM para esta categoria.

A Tabela 4 apresenta os resultados das principais métricas para avaliação do modelo de classificação treinado.

Tabela 4 - Resultado das Métricas para o Algoritmo SVM

Classe	TP-Rate	FP-Rate	Acurácia	Erros	Precisão	Sensitividade	Especificidade
A	1,00	0,10	0,97	0,03	0,97	1,00	0,90
B	1,00	0,00	1,00	0,00	1,00	1,00	1,00
C	0,71	0,00	0,71	0,29	1,00	0,71	1,00

Fonte: Próprio autor.

Com base nos resultados encontrados após a comparação entre as classificações dos especialistas com as classificações realizadas pelo *SpineFind,* com a aplicação das métricas demonstradas acima, escolheu-se o com os melhores resultados para a categoria “Descrição do Tratamento”. O algoritmo selecionado foi o SVM, com uma taxa de convergência entre as classificações de 97,47%.

3.6.2 Benefícios do tratamento

A categoria benefícios foi treinada com 64 textos, sendo a segunda categoria com maior número de textos para treinamento. Através da Tabela 5 é apresentado a quantidade de textos existentes para cada uma das classificações.

Tabela 5 – Quantidade de texto por classificação

Classificação	Quantidade de textos
Negativo	43
Positivo	14
Regular	7

Fonte: Próprio autor.

A Tabela 6 apresenta a comparação com as classificações definidas pelos especialistas da saúde. Nesta tabela é apresentado a taxa de acerto e erro de cada um dos algoritmos.

Tabela 6 – Comparação entre Especialistas x Algoritmos

Algoritmos	Acertos	Erros	% Acerto
Naïve Bayes	55	9	85,94%
SVM	53	11	82,81%
KNN	51	13	79,68%
J48	47	17	73,43%

Fonte: Próprio autor.

Por meio da ferramenta Weka foi possível gerar a Matriz de Confusão do algoritmo que apresentou os melhores resultados, conforme apresentado pela Tabela 7.

Tabela 7 – Matriz de confusão do algoritmo *Naïve Bayes*

A	B	C	
42	1	0	A = Negativo
5	9	0	B = Positivo
3	0	4	C = Regular

Fonte: Próprio autor.

A partir da análise da matriz de confusão demonstrada na Tabela 7, nota-se que:

- Dois textos positivos foram classificados como “Negativo”.
- Três textos regulares foram classificados como “Negativo”.

Através dos valores da Matriz de Confusão foram aplicadas as fórmulas e obtidos os detalhes da aplicação do algoritmo *Naïve Bayes*. A Tabela 8 apresenta os resultados das métricas para avaliação do modelo de classificação treinado.

Tabela 8 - Resultado das Métricas para o Algoritmo *Naïve Bayes*

Classe	TP-Rate	FP-Rate	Acurácia	Erros	Precisão	Sensitividade	Especificidade
A	0,98	0,38	0,79	0,21	0,84	0,98	0,62
B	0,64	0,02	0,57	0,43	0,90	0,64	0,98
C	0,57	0,00	0,57	0,43	1,00	0,57	1,00

Fonte: Próprio autor.

Por meio da visualização dos resultados acima é possível garantir a escolha do algoritmo *Naïve Bayes* como o com melhor desempenho e que mais se assemelha aos resultados de um especialista humano, com uma taxa de convergência entre as classificações de 85,94%.

3.6.3 Consequências do tratamento

Para a categoria “Consequências do tratamento” foram utilizados 35 textos para se treinar cada um dos 4 algoritmos que foram utilizados para os testes, a fim de se determinar aquele com o melhor resultado. Por meio da Tabela 9 é possível observar a quantidade de textos existentes para cada uma das classificações.

Tabela 9 – Quantidade de texto por classificação

Classificação	Quantidade de textos
Negativo	21
Positivo	8
Regular	6

Fonte: Próprio autor.

A Tabela 10 apresenta a comparação entre a classificação realizada pelos especialistas da saúde com a classificação realizada por cada um dos algoritmos avaliados, apresentando o número de acertos e erros de cada um dos algoritmos.

Tabela 10 – Comparação entre Especialistas x Algoritmos

Algoritmos	Acertos	Erros	% Acerto
Naïve Bayes	30	5	85,71%
SVM	31	4	88,57%
KNN	24	11	68,57%
J48	30	5	85,71%

Fonte: Próprio autor.

Com base na Matriz de Confusão do algoritmo que apresentou maior taxa de acerto entre as comparações acima (SVM) foi possível realizar o cálculo das métricas propostas. A Tabela 11 apresenta a Matriz de Confusão.

Tabela 11 – Matriz de confusão do algoritmo SVM

A	B	C	
21	0	0	A = Negativo
3	5	0	B = Positivo
1	0	5	C = Regular

Fonte: Próprio autor.

A partir da análise da matriz de confusão demonstrada na Tabela 11, nota-se que:

- Três textos positivos foram classificados como “Negativo”.
- Um texto regular foi classificado como “Negativo”.

A partir dos valores da Matriz de Confusão para o algoritmo SVM foram aplicadas as métricas propostas e verificado os detalhes da performance da utilização deste algoritmo para a categoria “Consequências do tratamento”.

A Tabela 12 apresenta os resultados das principais métricas para avaliação do modelo de classificação treinado.

Tabela 12 - Resultado das Métricas para o Algoritmo SVM

Classe	TP-Rate	FP-Rate	Acurácia	Erros	Precisão	Sensitividade	Especificidade
A	1,00	0,29	0,81	0,19	0,84	1,00	0,71
B	0,63	0,00	0,63	0,38	1,00	0,63	1,00
C	0,83	0,00	0,83	0,17	1,00	0,83	1,00

Fonte: Próprio autor.

Através dos resultados obtidos entre a comparação da classificação realizada pelos especialistas humanos com a classificação realizada em cada algoritmo, além da visualização e avaliação da performance do algoritmo com maior taxa de convergência desta comparação (88,57%), selecionou-se o algoritmo SVM como aquele que apresenta os melhores resultados para esta categoria.

3.6.4 Influência na qualidade de vida do paciente

A categoria “Influência na qualidade de vida do paciente” foi a categoria com o menor *corpus* entre todas as categorias do sistema *SpineFind.* Para esta categoria foram levantados apenas 11 trechos com informações sobre o assunto. Destes textos, 10 são com classificação negativa e 1 com a classificação regular. Não foram disponibilizados textos para a classificação “positiva”, impossibilitando a classificação positivo para esta categoria.

A Tabela 13 apresenta os resultados da comparação entre as classificações dos especialistas humanos com a classificação realizada por cada um dos 4 algoritmos testados.

Tabela 13 – Comparação entre Especialistas x Algoritmos

Algoritmos	Acertos	Erros	% Acerto
Naïve Bayes	11	0	100%
SVM	10	1	90,91%
KNN	10	1	90,91%
J48	10	1	90,91%

Fonte: Próprio autor.

Devido à pouca diversidade de informações para a etapa de treinamento desta categoria foi obtido uma alta taxa de acerto, visto que os dados são em sua maioria semelhantes e de fácil acerto para o algoritmo.

Devido ao fato de o algoritmo *Naïve Bayes* ser aquele com o maior número de acertos quando comparado a classificação humana foram gerados os dados da Matriz de Confusão da utilização deste algoritmo, a fim de se obter a visualização da performance do mesmo, conforme observado na Tabela 14.

Tabela 14 – Matriz de confusão do algoritmo *Naïve Bayes*

A	B	
11	0	A = Negativo
0	0	B = Regular

Fonte: Próprio autor.

A partir dos valores da Matriz de Confusão para o algoritmo *Naïve Bayes* foram aplicadas as métricas propostas e obtido a performance e os detalhes da utilização deste algoritmo para esta categoria.

A Tabela 15 apresenta os resultados das principais métricas para avaliação do modelo de classificação treinado.

Tabela 15 - Resultado das Métricas para o Algoritmo *Naïve Bayes*

Classe	TP-Rate	FP-Rate	Acurácia	Erros	Precisão	Sensitividade	Especificidade
A	10	0	1,00	0,00	1,00	1,00	1,00
B	1	0	1,00	0,00	1,00	1,00	1,00

Fonte: Próprio autor.

Observando a Matriz de Confusão e o resultado das métricas é possível observar que a taxa de convergência entre classificação dos humanos comparada a classificação da *ML* do sistema alcança o valor de 100%. Porém, como observado anteriormente, para esta categoria foram disponibilizados pouquíssimos dados pelos especialistas da saúde, sendo até mesmo excluído a classificação “Positiva” por não ter dados para treinamento. Devido este fato, os resultados obtidos para esta categoria podem ser considerados não tão conclusivos quanto aqueles obtidos pelas outras categorias, necessitando de novas análises, em trabalhos futuros, quando disponíveis novos dados para treinamento da ferramenta.

3.6.5 Riscos do tratamento

A categoria “Riscos do tratamento” contou com um total de 25 trechos que continham informações sobre o assunto para a etapa de treinamento e avaliação dos algoritmos. Por meio da Tabela 16 é possível observar a quantidade de textos existentes para cada uma das classificações.

Tabela 16 – Quantidade de texto por classificação

Classificação	Quantidade de textos
Negativo	10
Positivo	8
Regular	7

Fonte: Próprio autor.

Buscando avaliar os 4 algoritmos propostos neste projeto, o sistema foi treinado e avaliado com estes textos. Porém, diferentemente dos resultados obtidos nas outras categorias, para a categoria “Riscos” os resultados alcançados não foram considerados satisfatórios. Indiferentemente do algoritmo utilizado, foram obtidas mais classificações incorretas do que classificações corretas. Analisando os trechos utilizados no treinamento foi possível observar que independente da classificação definida pelos especialistas humanos, os trechos eram similares e enxutos, o que os tornavam parecidos e dificultavam a classificação automatizada.

Visando contornar esta situação e tornar a classificação desta categoria mais assertiva foi proposta uma abordagem diferente daquela até então utilizada. Ao invés de utilizar somente o trecho destacado pelos especialistas, extraiu-se e passou a ser utilizado para o treinamento da ferramenta todo o parágrafo onde trecho que continha a informação desta categoria se encontrava. Desta maneira, todo o contexto em volta das informações relacionadas a categoria fez parte do treinamento dos algoritmos.

Com base nesta nova abordagem foi realizado a comparação entre as classificações realizadas pelos especialistas com a classificação obtida após a *ML* estar devidamente treinada com cada um dos algoritmos. A Tabela 17 apresenta o resultado final desta comparação.

Tabela 17 – Comparação entre Especialistas x Algoritmos

Algoritmos	Acertos	Erros	% Acerto
Naïve Bayes	20	5	80%
SVM	19	6	76%
KNN	17	8	68%
J48	19	6	76%

Fonte: Próprio autor.

Com base na Matriz de Confusão, do algoritmo que apresentou maior taxa de acerto entre as comparações acima (*Naïve Bayes*), realizou-se o cálculo das métricas propostas. Assim foi possível observar as classificações realizadas pelo algoritmo. A Tabela 18 apresenta a Matriz de Confusão.

Tabela 18 – Matriz de confusão do algoritmo *Naïve Bayes*

A	B	C	
10	0	0	A = Negativo
2	6	0	B = Positivo
3	0	4	C = Regular

Fonte: Próprio autor.

A partir da análise da matriz de confusão demonstrada na Tabela 18, nota-se que:

- Dois textos positivos foram classificados como “Negativo”.
- Três textos regulares foram classificados como “Negativo”.

Em cima dos valores da Matriz de Confusão foram aplicadas as métricas a fim de se detalhar a performance da utilização deste algoritmo para esta categoria.

A Tabela 19 apresenta os resultados das principais métricas para avaliação do modelo de classificação treinado.

Tabela 19 – Resultado das Métricas para o Algoritmo *Naïve Bayes*

Classe	TP-Rate	FP-Rate	Acurácia	Erros	Precisão	Sensitividade	Especificidade
A	1,00	0,33	0,50	0,50	0,67	1,00	0,67
B	0,75	0,00	0,75	0,25	1,00	0,75	1,00
C	0,57	0,00	0,57	0,43	1,00	0,57	1,00

Fonte: Próprio autor.

Com base nestes resultados foi possível escolher o algoritmo que alcançou os resultados mais similares ao de um especialista humano. O algoritmo selecionado para esta categoria foi o *Naïve Bayes*, com uma taxa de convergência de (80%).

3.6.6 Resultados finais

Com base nos resultados individuais de cada uma das categorias foi possível determinar os algoritmos a constituírem o *Ensemble* implementado no *SpineFind*. Para determinação dos resultados individuais foi realizada a comparação entre a classificação determinada pelos especialistas e as classificações realizadas pelos 4 algoritmos, em cada uma das categorias. Após esta comparação, os algoritmos que tiveram as maiores taxas de acerto passaram por uma análise de performance, através da aplicação das métricas propostas na seção 3.3. A Tabela 20 apresenta o algoritmo selecionado para cada uma das categorias, assim como a taxa de acerto de cada um destes algoritmos. Conforme observado na Tabela 20, para 3 das 5 categorias o algoritmo *Naïve Bayes* foi aquele que obteve uma maior taxa de acerto quando comparado as classificações de um especialista humano, sendo que as outras 2 categorias alcançaram melhores resultados através da utilização do algoritmo *SVM*.

Tabela 20 - Algoritmos utilizados em cada categoria

Categoria	Algoritmo	Acertos	Erros	% Acertos
Descrição do tratamento	SVM	77	2	97,47%
Benefícios do tratamento	*Naïve Bayes*	55	9	85,94%
Consequências do tratamento	SVM	31	4	88,57%
Influência na qualidade de vida do paciente	*Naïve Bayes*	11	0	100%
Riscos do tratamento	*Naïve Bayes*	20	5	80%
		194	20	90,65%

Fonte: Próprio autor.

Ao todo foram disponibilizados 214 textos pelos especialistas da saúde. Estes foram utilizados para realizar o treinamento dos algoritmos e para posterior comparação entre a classificação realizada pelo sistema e a classificação determinada inicialmente pelos especialistas. Ao final 194 textos foram corretamente classificados pela ferramenta, desta forma é possível determinar que 90,65% dos textos avaliados foram corretamente classificados pelo *SpineFind*.

Para determinação da classificação final de um texto foram utilizadas as regras conforme a Tabela 21. As regras representam as classificações das 5 categorias presentes no sistema.

Tabela 21 – Regras para determinar a classificação do texto

Classificação	Regra
Sem classificação	Se “Sem classificação” => 3
Positivo	Se “Positivo” >= 4
Regular	Se “Regular” >= 3 OU “Regular” >= “Negativo”
Negativo	Nenhuma das alternativas acima

Fonte: Próprio autor.

Conforme nos estudos realizados ao início deste projeto, foi possível atender através deste projeto a todas as dimensões de qualidade de informação da área da saúde. A dimensão conteúdo é atendida pelos resultados fornecidos pela etapa de mineração de sistema deste projeto. Já a dimensão design foi abordada nas alterações de interface realizadas neste projeto. Por fim, considera-se que a dimensão técnica foi atendida devido ao fato de a ferramenta respeitar os critérios de segurança, como a autenticação de usuários para a o treinamento da ferramenta, e por deixar claro o seu objetivo através da página “sobre”

4. CONCLUSÕES

4.1 SÍNTESE DO TRABALHO

A grande quantidade de informações, na língua Portuguesa, relacionadas a saúde e condições de vida disponíveis na Internet e o acesso cada vez mais simples e instantâneo a estas informações, se transformou em um cenário de risco e preocupante. A validação da qualidade dos dados existente, somente pode ser garantida por especialistas da saúde, invalidando para grande parte da sociedade a garantia de coesão das informações encontradas. A automação deste processo, tornando-o acessível e disponível a todas as pessoas, é imprescindível e de grande importância, a fim de garantir uma tomada de decisão correta e assertiva de pacientes que buscam informações relacionas à saúde na internet.

Este projeto teve como objetivo principal realizar melhorias e avanços tecnológicos em uma ferramenta *web*, possibilitando a automação do processo de avaliação da qualidade dos dados textuais existentes na internet.

Para a obtenção do objetivo proposto foram estudados conceitos e técnicas de mineração de textos que viabilizem uma avaliação coesa e assertiva. O estudo de trabalhos relacionados, que utilizassem algoritmos de aprendizado de máquina em dados textuais relacionados a saúde, proporcionou escolha dos algoritmos testados durante este projeto. O detalhamento destes algoritmos fez parte do estudo do processo de mineração de texto a ser implementado.

Após estes estudos foram elencadas melhorias e serem implementadas no software objeto deste estudo. Primeiramente, realizou-se uma síntese do cenário no qual se encontrava a ferramenta. Concluída esta síntese foi possível levantar juntamente aos especialistas da saúde, quais melhorias e avanços eram necessários para ferramenta se se torna mais funcional. Como método de pesquisa, adotou-se o processo de mineração de textos proposto por Aranha (2007), assim como a arquitetura que se optou pela manutenção da já implementa no software.

Após estas definições foram elencadas as ferramentas e bibliotecas de auxílio para desenvolvimento e implementação melhorias que foram propostas.

Com auxílio de especialistas da saúde foram selecionados textos a serem utilizados para o treinamento e para as avaliações dos resultados encontrados. Tendo estes dados foi iniciado o desenvolvimento dos itens propostos

anteriormente, a fim de garantir a especialização do sistema nas principais características que levam a classificação da qualidade de um texto da área da saúde, através da implementação de uma arquitetura de *Ensemble* de algoritmos. Com base nestas alterações foram necessárias mudanças a apresentação dos resultados, a fim de evidenciar os resultados individuais e os resultados finais do processo de classificação de textos do sistema.

Com o sistema desenvolvido, considera-se que os resultados são promissores e evidenciam a viabilidade de uso de técnicas de aprendizado automático no tratamento de textos da área da saúde.

4.2 CONTRIBUIÇÕES DO TRABALHO

Ao início deste projeto, a expectativa era de implementar uma ferramenta de classificação automática para textos da área da saúde, escritos na língua Portuguesa, a fim de se atingir resultados semelhantes aos de especialistas da saúde.

Por meio deste contato com os especialistas da saúde foi adquirido durante o desenvolvimento deste trabalho conhecimentos que ultrapassam os estudos presentes no currículo do curso de graduação. Ao final deste projeto é concludente afirmar que o mesmo deixa importantes contribuições, em especial as áreas da Computação e Medicina.

Com a conclusão deste projeto, a área de mineração de textos adquiriu novos conhecimentos acerca da utilização de *Ensembles* de algoritmos para as análises e classificações de textos na língua Portuguesa, conceitos abordados durante o desenvolvimento do trabalho, podendo servir como referência para futuros projetos e estudos.

Além disto, é considerado a que este trabalho pode ser utilizado como base para o desenvolvimento de um framework de aplicação de algoritmos de aprendizado de máquina. Através das configurações e parametrizações adicionadas a ferramenta, é possível realizar a aplicação das técnicas adicionados a ferramenta a diversos casos diferentes, de acordo com a necessidade e o treinamento realizado.

Como contribuições técnicas para a área de desenvolvimento de software são consideradas a organização do código e arquitetura do sistema, que fornece uma alta escalabilidade a este sistema.

Em relação a área da Medicina, este projeto deixa como legado uma ferramenta capaz de auxiliar os pacientes em relação a tomada de decisão quanto a qualidade e confiabilidade das informações disponíveis na Internet.

4.3 TRABALHOS FUTUROS

Como sugestão para trabalhos futuros é proposto a implementação de novas funcionalidades que permitam a integração entre os usuários do sistema, possibilitando a integração destes através da ferramenta. Desta maneira é possível obter uma ferramenta mais atrativa aos especialistas da saúde, permitindo a iteração destes através da criação de chats e fóruns dentro da própria ferramenta, tornando o *SpineFind* um sistema colaborativo.

Conforme apresentado na seção de resultados, alguns dos algoritmos utilizados na arquitetura *Ensemble* do sistema dispuseram de uma pequena quantidade de textos durante a etapa de treinamento. Para estes casos, torna-se interessante a avaliação junto aos especialistas da saúde da possibilidade de realizar novos treinamentos com uma maior quantidade de arquivos, especialmente para a categoria “Influências na qualidade de vida”. Esta categoria que durante a elaboração deste projeto não dispôs de textos com a classificação “Positiva” para a realização do treinamento.

Por fim, seriam interessantes também estudos e implementações acerca de SEO (Otimização para mecanismos de busca), a fim de facilitar a busca e o acesso da ferramenta pelos pacientes através dos mecanismos de buscas disponíveis na Internet.

5. REFERÊNCIAS

ABEL, F. **Análise textual automática: apreensibilidade e qualidade da informação na área da saúde**. Universidade de Caxias do Sul. Caxias do Sul, p. 203. 2016.

ABERNETHY, M. Mineração de dados com WEKA, Parte 1: Introdução e regressão. **IBM developerWorks**, 2010. Disponível em: <https://www.ibm.com/developerworks/br/opensource/library/os-weka1/>. Acesso em: 24 Maio 2017.

ALPAYDIN, Ethem. **Introduction to machine learning.** 3. ed. Massachusetts: Massachusetts Institute Of Technology, 2014. 613 p.

AMARAL, F. **Aprenda Mineração de Dados:** Teoria e prática. São Paulo: Alta Books, 2016.

AMBRÓSIO, A. P. L.; MORAIS, E. A. M. **Mineração de Textos**. Instituto de Informática Universidade Federal de Goiás. Goiás, p. 29. 2007.

ARANHA, C. N. **Uma Abordagem de Pré-Processamento Automático para Mineração de Textos em Português: Sob o Enfoque da Inteligência Computacional**. Pontifícia Universidade Católica do Rio de Janeiro. Rio de Janeiro, p. 144. 2007.

ARANHA, C. N.; PASSOS, E. A Tecnologia de Mineração de Textos. **Revista Elerônica de Sistemas de Informação**, v. V, n. 2, p. 1-8, 2006.

BALLOU, D.; MADNICK, S.; WANG, R. Special Section: Assuring Information Quality. **Journal of Management Information Systems**, p. 9-11, 2004.

BARANAUSKAS, J. A.; MONARD, M. C. Conceitos Sobre Aprendizado de Máquina. **Sistemas Inteligentes Fundamentos e Aplicações**, Barueri, v. I, n. 1, p. 89-114, 2003.

BARBOZA, E. M. F.; NUNES, E. M. D. A. A inteligibilidade dos websites governamentais brasileiros e o acesso para usuários com baixo nível de escolaridade. **Inclusão Social**, Brasilia, v. II, p. 19-33, Abril 2007.

BLEJMAN, M. Como D3.js está mudando a forma de contar histórias com dados (e por que precisamos de uma hackatona). **IJNet**, 2013. Disponivel em: <https://ijnet.org/pt-br/blog/como-d3js-esta-mudando-forma-de-contar-historias-com-dados-e-por-que-precisamos-de-uma-hackaton>. Acesso em: 15 maio 2017.

BORGES, R. D. O.; SILVA, R. A. A. D.; CASTRO, S. S. D. **Utilização da classificação por distância euclidiana no mapeamento dos focos de arenização no setor sul da alta bacia do Rio Araguaia**. Simpósio Sul Brasileiro de Sensoriamento Remoto. Florianópolis: [s.n.]. 21 Abril 2007. p. 21-26.

BOSCH, A. V. D.; BOGERS, T.; KUNDER, M. D. Estimating search engine index size variability: a 9-year longitudinal study. **Scientometrics**, 09 Fevereiro 2016. 839-856.
BOUCKAERT, R. R. et al. **WEKA Manual for Version 3-8-0**. Hamilton, New Zealand: The Univarsity University of Waikato, 2016.

BRANDEWINDER, M. **Machine Learning Projects for.NET Developers**. New York: Apress, 2015.

BUANI, B. E. Z.; HIRAKAWA, A. R.; BUENO, J. F. **Aplicação do algoritmo dos k-vizinhos mais próximos para seleção de características da morfologia de asas de abelhas sem ferrão**. Universidade de São Paulo. Viçosa, p. 5. 2009.

CARVALHO, A. C. P. L. F. D.; LORENA, A. C. **Introdução às Máquinas de Vetores Suporte (Support Vector Machines)**. Universidade de São Paulo. São Paulo, p. 58. 2003.

CHARNOCK, D.; SHEPPERD, S. Learning to DISCERN online: applying an appraisal tool to health websites in a workshop setting. **HEALTH EDUCATION RESEARCH**, Oxford, v. XIX, n. 4, p. 440-446, August 2004.

CISCO. The Zettabyte Era: Trends and Analysis. **Cisco**, 2016. Disponivel em: <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-indexvni/vni-hyperconnectivity-wp.html>.

DA SILVA, L. M. O. **Uma Aplicação de Árvores de Decisão, Redes Neurais e KNN para a Identificação de Modelos ARMA Não-Sazonais e Sazonais**. Pontifícia Universidade Católica do Rio de Janeiro. Rio de Janeiro, p. 145. 2005.

FILHO, V. M. **Um Novo Algoritmo de Agrupamento Semi-Supervisionado Baseado no Fuzzy C-Means**. Universidade Federal de Pernanbuco. Recife, p. 109. 2009.

GIACOMEL, F. D. S. **Um método algorítmico para operações na bolsa de valores baseado em ensembles de redes neurais para modelar e prever os movimentos dos mercados de ações**. Universidade Federal do Rio Grande do Sul. Porto Alegre, p. 92. 2016.

GOMES, R. M. **Desambiguação de Sentido de Palavras Dirigida por Técnicas de Agrupamento sob o Enfoque da Mineração de Textos**. Pontifícia Universidade Católica do Rio de Janeiro. Rio de Janeiro, p. 118. 2009.

HE, W.; ZHA, S.; LI, L. Social media competitive analysis and text mining: A casae study in the pizza industry. **International Journal of Information Management**, p. 464-472, 2013.

HEATON, J. **Programming Neural Networks Programming Neural Networks**. 1. ed. St. Louis: Heaton Research, Inc, v. 1, 2011.

INFORMED Consent. **ETHICS IN MEDICINE**, 2014. Disponível em: <https://depts.washington.edu/bioethx/topics/consent.html>. Acesso em: 20 Abril 2017.

LAW, G. Error in the Fernandez Huerta Readability Formula. **The Linguist List**, 2011. Disponível em: <https://linguistlist.org/issues/22/22-2332.html>. Acesso em: 03 abr. 2017.

LE, James. **The 10 Algorithms Machine Learning Engineers Need to Know.** 2016. Disponível em: <https://www.kdnuggets.com/2016/08/10-algorithms-machine-learning-engineers.html>. Acesso em: 10 set. 2017.

LOPES, M. C. S. **Mineração de Dados Textuais Utilizando**. Universidade Federal do Rio de Janeiro. Rio de Janeiro, p. 162. 2004.

MENDONÇA, A. P. B.; NETO, A. P. Critérios de avaliação da qualidade da informação em sistes de saúde: uma proposta. **Revista Eletrônica de Comunicação, Informação & Inovação em Saúde**, p. 1-15, 2015.

MICROSOFT. Referência técnica do algoritmo Árvores de Decisão da Microsoft. **Microsoft Developer Network**, 2016. Disponível em: <https://msdn.microsoft.com/pt-br/library/cc645868.aspx>. Acesso em: 05 maio 2017.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre Aprendizado de Máquina. In: REZENDE, S. O. **Sistemas inteligentes:** fundamentos e aplicações. [S.l.]: Editora Manole Ltda, 2003. Cap. 4, p. 39-56.

PACHECO, A. K vizinhos mais próximos – KNN. **Computação Inteligente**, 2017. Disponível em: <http://www.computacaointeligente.com.br/algoritmos/knn-k-vizinhos-mais-proximos/>. Acesso em: 01 maio 2017.

SANCHES, Marcelo Kaminski. **Aprendizado de Máquina Semi-supervisionado**: Proposta de um Algoritmo para Rotular Exemplos a Partir de Poucos Exemplos Rotulados. Dissertação (Mestrado) -- Universidade de São Paulo, 2003.

SHALEV-SHWARTZ, S.; BEN-DAVID, S. **Understanding Machine Learning:** From Theory to Algorithms. Cambridge: Cambridge University Press, 2014.

SHARKEY, A. **Combining artificial neural nets**. Sheffield: Springer-Verlag, 1998.

SILVA, E. R. C. C. **TÉCNICAS DE DATA E TEXT MINING PARA ANOTAÇÃO DE UM ARQUIVO DIGITAL**. Universidade de Aveiro. Aveiro, p. 89. 2010.

SMOLA, A.; VISHWANATHAN, S. V. N. **Introduction to Machine Learning**. Cambridge: Cambridge University Press, 2008.

SOARES, F. D. A. **Mineração de Textos na Coleta Inteligente de Dados na Web**. Pontifícia Universidade Católica do Rio de Janeiro. Rio de Janeiro, p. 120. 2008.

SUPERVISED and Unsupervised Machine Learning Algorithms. **Machine Learning Mastery**, 2016. Disponível em: <http://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/>. Acesso em: 2 Abril 2017.

SZLOSEK, D. A.; JONATHAN M, F. Using Machine Learning and Natural Language Processing Algorithms to Automate the Evaluation of Clinical Decision Support in Electronic Medical Record Systems. **eGEMs (Generating Evidence & Methods to improve patient outcomes)**, p. 1-11, 2016.

WEKA. Stemmers. **Weka**, 2016. Disponível em: <https://weka.wikispaces.com/Stemmers>. Acesso em: 11 abr. 2017.

WILKES, K. Avoiding the Dangers of Online Health Information. **Whole Health Insider**, 2015. Disponível em: <http://www.wholehealthinsider.com/newsletter/avoiding-dangers-online-health-information/>. Acesso em: 02 abr. 2017.

WITTEN, I. H.; FRANK, E. **Data Mining:** Practical Machine Learning Tools and Techniques. 2ª. ed. San Francisco: Elsevier, 2005.

YAMADA, H. et al. A Development of Automatic Audit System for Written Informed Consent using Machine Learning. **Studies in Health Technology and Informatics**, v. 216, p. 926-926, 2015.

ZHANG, Y. et al. **A Machine Learning Approach for Rating the Quality of Depression Treatment Web**. iConference 2014 Proceedings. Berlin: Berlin School of Library and Information Science, Humboldt-Universität zu Berlin. 2014. p. 192-212.

APÊNDICE A

LISTA DE STOPWORDS

À	ao	cima	Depois	duas	esteve	fomos	houvera
às	aonde	cinco	Desde	durante	estivéramos	for	houveram
área	aos	coisa	desligado	e	estivéssemos	fora	houverei
é	após	com	Dessa	ela	estive	foram	houverem
éramos	apenas	como	dessas	elas	estivemos	forem	houveremos
és	apoio	comprido	Desse	ele	estiver	forma	houveria
último	apontar	conhecido	desses	eles	estivera	formos	houveriam
era	aquela	conselho	Desta	em	estiveram	fosse	houvermos
haja	aquelas	contra	destas	embora	estiverem	fossem	houvesse
são	aquele	contudo	Deste	enquanto	estivermos	foste	houvessem
sou	aqueles	corrente	destes	então	estivesse	fostes	iniciar
tenha	aqui	cuja	Deve	entre	estivessem	fui	inicio
terá	aquilo	cujas	devem	eram	estiveste	geral	irá
terei	as	cujo	deverá	essa	estivestes	grande	ir
teria	assim	cujos	Dez	essas	estou	grandes	isso
tivéramos	até	custa	dezanove	esse	eu	grupo	isto
tive	atrás	dá	dezesseis	esses	exemplo	há	já
tiver	através	dão	dezessete	está	fôramos	hão	lá
aí	baixo	dúvida	dezoito	estás	fôssemos	hajam	lado
a	bastante	da	Dia	estávamos	faço	hajamos	lhe
acerca	bem	daquela	diante	estão	falta	havemos	lhes
adeus	boa	daquelas	direita	esta	fará	havia	ligado
agora	boas	daquele	dispõe	estado	favor	hei	local
ainda	bom	daqueles	dispõem	estamos	faz	hoje	logo
além	bons	dar	diversa	estará	fazeis	hora	longe
algo	breve	das	diversas	estar	fazem	horas	lugar
algumas	cá	de	diversos	estas	fazemos	houvéramos	máximo
alguns	cada	debaixo	Diz	estava	fazer	houvéssemos	mês
ali	caminho	dela	dizem	estavam	fazes	houve	maior
ambas	catorze	delas	dizer	este	fazia	houvemos	maioria
ambos	cedo	dele	Do	esteja	fez	houverá	maiorias
ano	cento	deles	Dois	estejam	fim	houverão	mais
anos	certamente	demais	Dos	estejamos	final	houveríamos	mal
antes	certeza	dentro	doze	estes	foi	houver	mas

me	nesses	ou	Possivelmente	queres	seu	tente	tua
mediante	nesta	outra	posso	quero	seus	tentei	tuas
meio	nestas	outras	pouca	questão	sexta	terão	tudo
menor	neste	outro	poucas	quieto	sexto	teríamos	um
menos	nestes	outros	pouco	quinta	sim	ter	uma
meses	no	pôde	poucos	quinto	sistema	terceira	umas
mesma	noite	põe	povo	quinze	sob	terceiro	uns
mesmas	nome	põem	própria	relação	sobre	teremos	usa
mesmo	nos	para	próprias	sétima	sois	teriam	usar
mesmos	nossa	parece	próprio	sétimo	somente	teu	vários
meu	nossas	parte	próprios	só	somos	teus	vão
meus	nosso	partir	próxima	sabe	sua	teve	vêm
mil	nossos	pegar	próximas	sabem	suas	tinha	vós
minha	nova	pela	próximo	saber	tão	tinham	vai
minhas	novas	pelas	próximos	se	têm	tipo	vais
momento	nove	pelo	primeira	segunda	tínhamos	tivéssemos	valor
muito	novo	pelos	primeiras	segundo	tal	tivemos	veja
muitos	novos	perante	primeiro	sei	talvez	tivera	vem
não	num	perto	primeiros	seis	também	tiveram	vens
nível	numa	pessoas	puderam	seja	tanta	tiverem	ver
nós	numas	pode	Quê	sejam	tantas	tivermos	verdade
número	nunca	podem	quais	sejamos	tanto	tivesse	vez
na	nuns	poderá	qual	sem	tarde	tivessem	vezes
nada	o	poder	qualquer	sempre	te	tiveste	viagem
naquela	obra	podia	quando	sendo	tem	tivestes	vindo
naquelas	obrigada	pois	quanto	será	temos	toda	vinte
naquele	obrigado	ponto	quarta	serão	tempo	todas	você
naqueles	oitava	pontos	quarto	seríamos	tendes	todo	vocês
nas	oitavo	por	quatro	ser	tenham	todos	vos
nem	oito	porquê	Que	serei	tenhamos	três	vossa
nenhuma	onde	porque	quem	seremos	tenho	trabalhar	vossas
nessa	ontem	portanto	quer	seria	tens	trabalho	vosso
nessas	onze	posição	quereis	seriam	tentar	treze	vossos
nesse	os	possível	querem	sete	tentaram	tu	zero

APÊNDICE B

GUIA DO USUÁRIO

Neste apêndice são apresentadas as instruções de uso do sistema para o usuário. Também são descritos como devem ser disponibilizados os textos para o treinamento da ferramenta e para a avaliação.

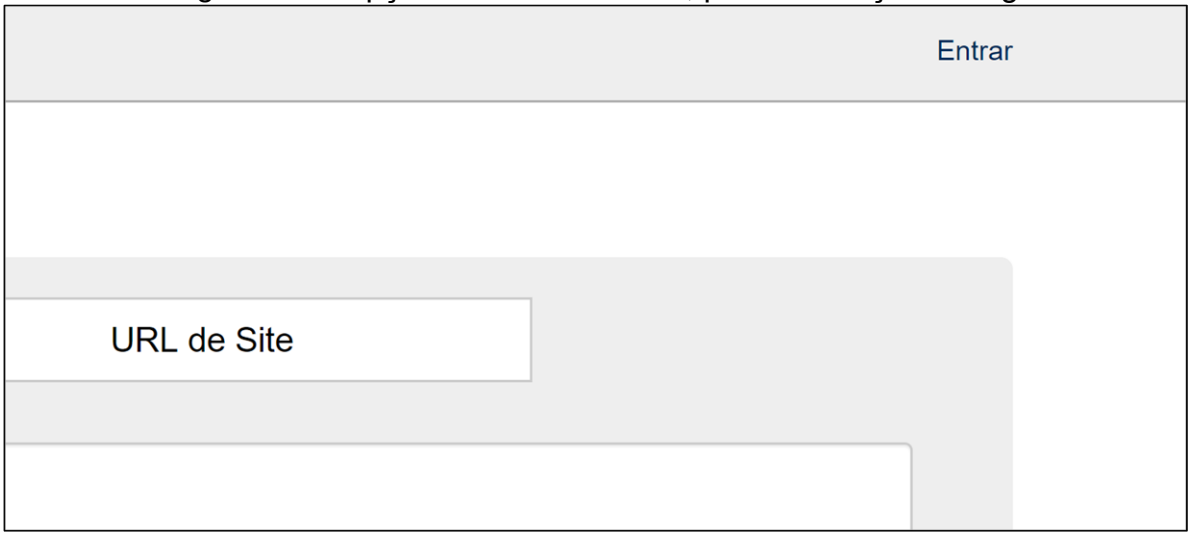
REALIZAR TREINAMENTO

A etapa de treinamento está disponível somente para os usuários administradores do sistema. Abaixo, é apresentado o passo-a-passo para realização do treinamento da aplicação.

Etapa 1 – Acesso ao sistema e alteração de senha

Através de qualquer página da aplicação, é possível realizar o *login* de um usuário. A aplicação possui apenas um usuário chamado “*admin*” que possui uma senha padrão “*admin*”, que foram configurados para o primeiro uso. Recomenda-se, que a senha seja alterada após o primeiro acesso ao sistema. Para acessar a tela de acesso ao sistema, é necessário clicar em “Entrar” no canto direito superior no menu, conforme exibido na Figura 27.

Figura 27 – Opção “Entrar” no menu, para realização do login



Fonte: Próprio autor.

Para efetivar o acesso ao sistema, deve ser informado o usuário e a senha nos respectivos campos demonstrados na Figura 28, além de realizar a confirmação do acesso.

Figura 28 – Confirmação de *Login* ao sistema

Fonte: Próprio autor.

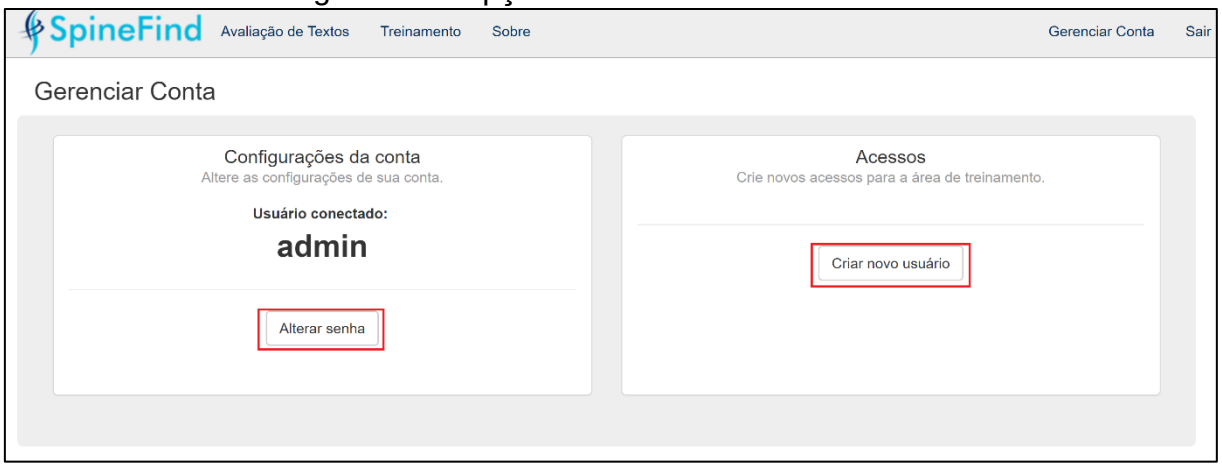
Após ter acesso ao sistema, novas opções estarão disponíveis no menu superior do sistema. Para realizar a troca da senha, é necessário clicar em “Gerenciar Conta”, e para finalizar o acesso ao sistema é necessário clicar em “Sair”. Estas opções são demonstradas na Figura 29.

Figura 29 – Opção de gerenciar conta e sair do sistema

Fonte: Próprio autor.

Conforme exibido na Figura 30, através do acesso da opção “Gerenciar Conta” é possível realizar a troca de senha do usuário que está acessando o sistema, assim como realizar a criação de novos usuários.

Figura 30 – Opções da tela “Gerencia Conta”



Fonte: Próprio autor.

Etapa 2 – Adicionando arquivos de treinamento

Estando logado ao sistema, a partir de qualquer página é possível clicar em “Treinamento” no menu superior do site. Nessa página, é possível incluir um único arquivo de treinamento de forma manual, através do botão “Novo Arquivo”. Também é possível realizar o *upload* de vários arquivos, através do botão “Carregar arquivos”.

Na inserção unitária de um novo texto, é necessário informar os dados referentes a este novo arquivo, conforme demonstrado pela Figura 31.

Figura 31 - inserção unitária de um novo arquivo para treinamento

Fonte: Próprio autor.

Já através da opção “Carregar Arquivos” é possível realizar a inserção de uma lista de itens, para isso os arquivos devem estar no formatado “*.txt*”. Além disto a página de oferece algumas dicas que facilitam e agilizam a inserção destes arquivos, conforme demonstrado pela Figura 32.

Figura 32 – Inserção de vários arquivos

Fonte: Próprio autor.

Etapa 3 – Realizando o treinamento

O treinamento é realizado através do clique no botão “Realizar Treinamento” exibido em azul no canto superior direito da tabela de arquivos, conforme a Figura 33.

Figura 33 – Lista de arquivos e opção de “Realizar Treinamento”

Fonte: Próprio autor.

Ao final do treinamento, uma página informando da conclusão do treinamento é exibida, esta conta com algumas informações sobre o treinamento de cada uma das categorias existentes no sistema, detalhando quais treinamentos foram realizados. Conforme é apresentado pela Figura 34.

Figura 34 – Treinamento concluído

Fonte: Próprio autor.

AVALIAR UM NOVO TEXTO

A avaliação de um novo texto apresenta os resultados acerca de sua apreensibilidade e qualidade. Enquanto que os resultados acerca da qualidade do texto possuem como pré-requisito a realização da etapa de treinamento da aplicação, os resultados acerca da apreensibilidade independem da mesma.

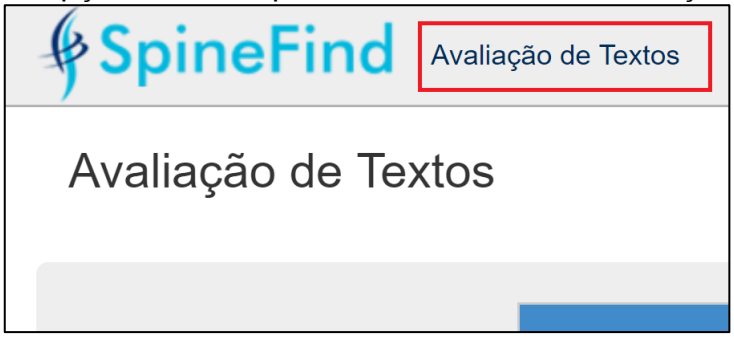
A avaliação de um novo texto não necessita que o usuário se identifique através do seu *login*, sendo disponível para qualquer pessoa com acesso à internet.

A seguir, é descrito o passo-a-passo para avaliar uma nova amostra textual através da aplicação.

Etapa 1 – Inserindo um texto ou *URL*

A página inicial da aplicação é responsável pela avaliação de textos, e também pode ser acessada através da opção “Avaliação de Textos” no menu superior do sistema, conforme a Figura 35.

Figura 35 – Opção do menu para acessar a tela de avaliação de textos



Fonte: Próprio autor.

Na tela de avaliação de textos, estão disponíveis duas opções para o usuário, as abas “Texto” e “*URL* de site”. A primeira aba permite que um novo texto seja informado manualmente, já a segunda disponibiliza a inserção de uma *URL*, que contenha o conteúdo a ser avaliado.

Após informar o texto manual ou a *URL* de um site, para que o texto seja avaliado, o botão “Realizar Avaliação” deve ser acionado, conforme destacado na Figura 36.

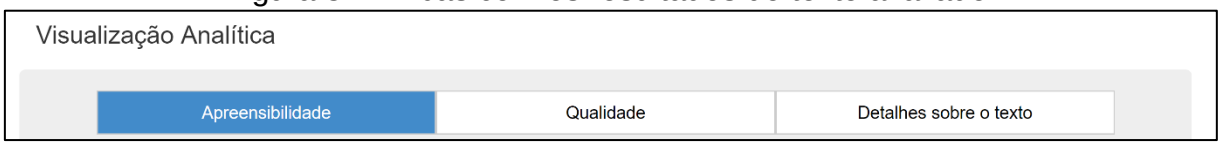
Figura 36 – Opções para realizar a avalição de um novo conteúdo textual

Fonte: Próprio autor.

Etapa 2 – Resultados da avaliação

Após a realização do processo anterior, o usuário será redirecionado para página contendo os resultados do *SpineFind*. Nesta página o usuário tem disponíveis três novas abas, conforme a Figura 37.

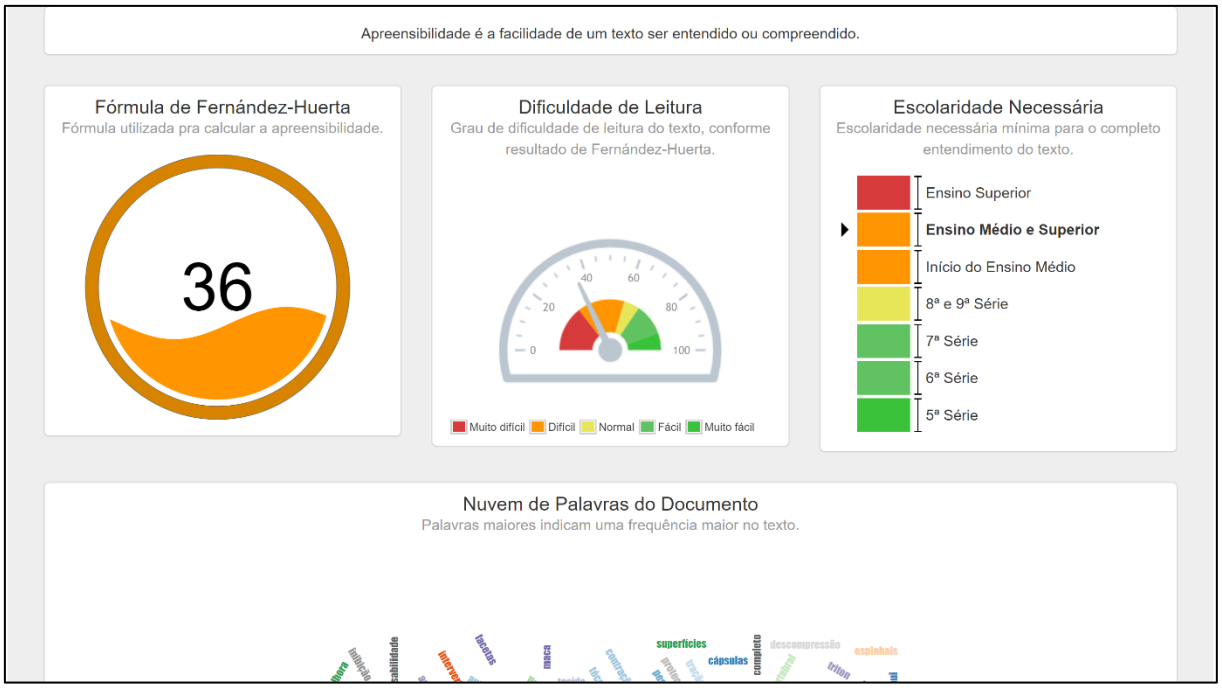
Figura 37 – Abas com os resultados do texto avaliado



Fonte: Próprio autor.

A primeira aba apresenta os resultados referentes a apreensibilidade, conforme a Figura 38.

Figura 38 – Resultados referentes a apreensibilidade do texto



Fonte: Próprio autor.

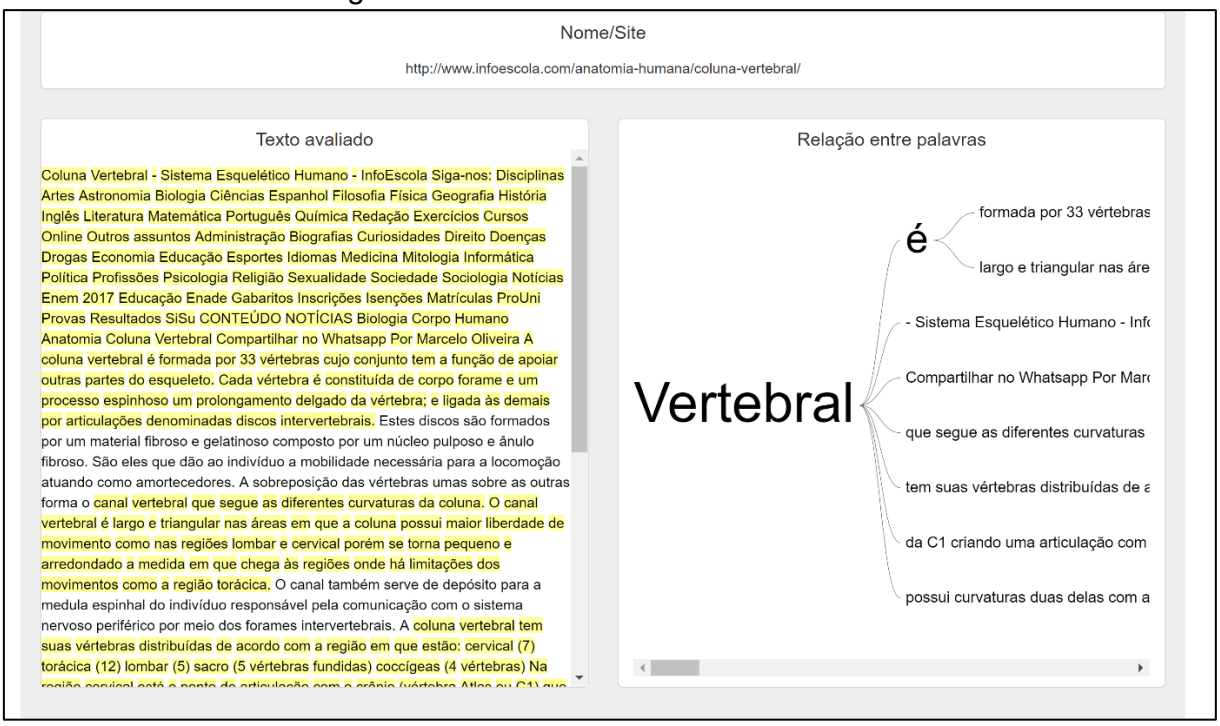
Já a segunda aba, “Qualidade”, apresenta os resultados referentes a qualidade do arquivo analisado. Conforme a Figura 39.

Figura 39 – Resultados referentes a qualidade do texto

Fonte: Próprio autor.

Por fim, na aba “Detalhes sobre o texto”, é possível observar o texto avaliado, bem como interagir com o texto através do “clique” sob as palavras do mesmo. Conforme a Figura 40.

Figura 40 – Aba “Detalhes sobre o texto”



Fonte: Próprio autor.