

Uma Ferramenta para Classificação Manual de Dados

Adyson Lucas Bernardo Queiroga, Yuri de Almeida Malheiros Barbosa

Departamento de Ciências Exatas (DCX) – Universidade Federal da Paraíba (UFPB)

CEP 58297-000 – Av. Santa Elisabete, 160, Rio Tinto – PB – Brazil

{adyson.lucas,yuri}@dcx.ufpb.edu.br

Abstract. *The large amount of text available on the web such as comments, news, social networks and chats has created in companies and institutions an interest in opinions and feelings expressed by users. However, analyzing this data manually is a difficult and time-consuming task. An alternative to automatic data labeling is supervised machine learning, which aims to create classifiers that learn from experiences and patterns. To obtain a supervised model with high accuracy, is necessary to train it with a sample of previously labeled data. This work aims to create a tool to assist manual data labeling for the training of supervised learning classifiers.*

Resumo. *A grande quantidade de texto disponível na web em forma de comentários, notícias, redes sociais e bate-papos têm gerado em empresas e instituições um interesse por opiniões e sentimento expressos por usuários. Porém, analisar esses dados manualmente é uma tarefa difícil e que demanda muito tempo. Uma alternativa para classificação automática dos dados é a aprendizagem de máquina supervisionada, que visa criar classificadores que aprendem a partir de experiências e padrões. Para obter um modelo supervisionado com alta precisão de acerto é necessário treiná-lo com uma amostra de dados previamente rotulados. Este trabalho tem como objetivo a criação de uma ferramenta para auxiliar na classificação manual de dados para o treinamento de classificadores de aprendizagem de máquina supervisionada.*

1. Introdução

A Internet têm sido um lugar de destaque para extração de informação, devido a sua grande quantidade de usuários. De forma frequente os usuários utilizam este meio para acesso à notícias, redes sociais, bate-papos, entre outros. As informações geradas pelos usuários despertam nas empresas e instituições interesse pelas opiniões, pensamentos e sentimentos relatados na internet. Porém, analisar esses dados manualmente, em certa medida, se torna inviável, por ser uma tarefa difícil e que demanda muito tempo.

A análise de sentimentos têm sido uma das áreas de pesquisa com mais rápido crescimento, com cerca de 7.000 artigos, sendo 99% deles publicados entre 2004 e 2016 [Mika Mäntylä 2016]. Entre as principais técnicas para análise de sentimento, está a

¹ Trabalho de conclusão de curso, sob orientação do professor Yuri de Almeida Malheiros Barbosa, submetido ao Curso de Bacharelado em Sistemas de Informação do Centro de Ciências Aplicadas e Educação (CCAEE) da Universidade Federal da Paraíba, como parte dos requisitos necessários para obtenção do grau de BACHAREL EM SISTEMAS DE INFORMAÇÃO.

aprendizagem de máquina supervisionada, em que dados rotulados são utilizados para que um classificador possa aprender padrões e classificar futuras entradas.

Para treinar um classificador são necessários muitos dados rotulados, tornando essa atividade uma das grandes dificuldades, uma vez que para realizar a análise de sentimento problemas de alta subjetividade é visto durante esse processo, bem como a necessidade de tempo para que as pessoas determinem a polaridade de muitas sentenças [Araújo 2015]. A atribuição dos rótulos nos dados de treinamento pode ser automatizado, mas os dados rotulados manualmente tendem a ter mais qualidade, pois há maior garantia de que os dados foram rotulados corretamente, ocasionando em menos falhas dos classificadores. Dessa forma, ferramentas para facilitar e agilizar a atribuição de rótulos manualmente aos dados de treinamento podem ser cruciais para a viabilização de soluções usando aprendizagem de máquina.

Este trabalho tem como objetivo desenvolver uma ferramenta para facilitar a classificação manual de dados textuais para treinamento de classificadores de aprendizagem de máquina. Para isso foi desenvolvida uma aplicação *web*, na qual pesquisadores possam inserir seu *dataset* e exibi-los para diversos usuários para classificação dos dados. A aplicação *web* foi feita utilizando Angular 8, os dados serão processados em uma API feita com Node.JS e persistidos no MongoDB.

O trabalho apresenta-se organizado em seções conforme descrito a seguir. A fundamentação teórica encontra-se na Seção 2. Na Seção 3 é apresentada a metodologia contendo os requisitos da ferramenta desenvolvida e ferramentas disponíveis relacionadas ao tema em estudo. As funcionalidades, arquitetura e interface da ferramenta são apresentadas na Seção 4. Na Seção 5 serão apresentados os resultados. Por fim, a Seção 6 apresenta as conclusões finais e perspectivas de trabalhos futuros.

2. Fundamentação Teórica

2.1 Análise de Sentimentos

Pesquisas vêm sendo desenvolvidas nas áreas de tratamento de opiniões, sentimentos e emoções em textos, em vista da grande demanda de texto disponível na *web* em forma de comentários, notícias, redes sociais e bate-papos. A partir destes ambientes os usuários terminam sendo induzidos a dar posicionamentos sobre diferentes temas em destaque, sejam esses sobre eventos, produtos, serviços, situações do dia a dia e pessoas [Liu 2012]. Conforme Liu (2012), abordagens baseadas em técnicas, algoritmos e modelos responsáveis pelo tratamento de opiniões são definidas pelas áreas de Mineração de Opiniões e Análise de Sentimentos. Essas áreas vêm obtendo um grande destaque em pesquisas por principalmente visar ajudar os usuários a encontrar dados opinativos e detectar a polaridade da opinião, isto é, detectar se um dado texto possui conotação positiva, negativa ou de classe neutra (ausência de sentimento).

Existem três categorias definidas por Liu (2012) para análise de sentimentos com relação à granularidade da análise textual.

- Granularidade de documento: equivale em classificar se a opinião geral no documento expõe um sentimento positivo, negativo ou neutro [Pang *et al.* 2002].

- Granularidade de sentença: relacionada a subdivisão do texto em sentenças específicas, em que ocorre uma análise de tais sentenças para determinar se expressam sentimentos positivos, negativos ou neutros.
- Granularidade de entidade e de aspecto: denominada como mineração de opiniões baseadas em características, ao invés de apresentar o foco maior em documentos, parágrafos e sentenças.

2.2 Aprendizagem de Máquina Supervisionada

Aprendizagem de Máquina é um subcampo da Inteligência Artificial, que visa a construção máquinas ou classificadores que conseguem aprender a partir da experiência, observando padrões de dados. A aprendizagem de máquina é dividida em duas categorias [Norvig e Russell 2014]: aprendizagem supervisionada e não-supervisionada. Na aprendizagem supervisionada, um classificador é treinado a partir de um *dataset* composto por vetores de características, também chamados de dados de entrada, com um rótulo associado, ou dados de saída, para que posteriormente seja possível a classificação de novas entradas [Araújo 2015]. Enquanto na aprendizagem não-supervisionada, o algoritmo necessita apenas de dados de entrada, dessa forma a classificação ocorre por meio de observações automáticas de padrões e similaridades entre as entradas fornecidas [Ferneda 2006].

É importante entender que na aprendizagem supervisionada, foco do presente trabalho, um conjunto de dados rotulados são anteriormente definidos com objetivo de treinar um classificador, para que este seja capaz de indicar/prever rótulos para novas entradas. Dessa maneira, para alcançar um bom modelo supervisionado que tome decisões precisas é indispensável que exista uma amostra do domínio do problema disponível com dados rotulados conhecidos para que seja possível obter um conjunto de dados finitos para treinamento e testes. Considerando como exemplo que um classificador, treinado com um *dataset* de *tweets* rotulados como positivos ou negativos, pode prever se um determinado *tweet*, ainda não rotulado, é ou não positivo.

2.2 Ferramentas existentes

Antes de iniciar o desenvolvimento da ferramenta, foram realizadas buscas com o intuito de descobrir outras ferramentas semelhantes que contemplassem as funcionalidades almejadas, definidas na Subseção 3.1.

Grande parte das ferramentas para classificação textual encontradas oferecem, principalmente, a funcionalidade de *text annotation*. Esta funcionalidade, ilustrada na Figura 1, consiste na atribuição de rótulos a uma palavra ou um conjunto delas que compõem um texto ou sentença. Alguns exemplos dessas ferramentas são: doccano², tagtog³, Datururks⁴ e LightTag⁵.

² <https://doccano.herokuapp.com>

³ <https://www.tagtog.net/>

⁴ <https://datururks.com/>

⁵ <https://www.lighttag.io/>

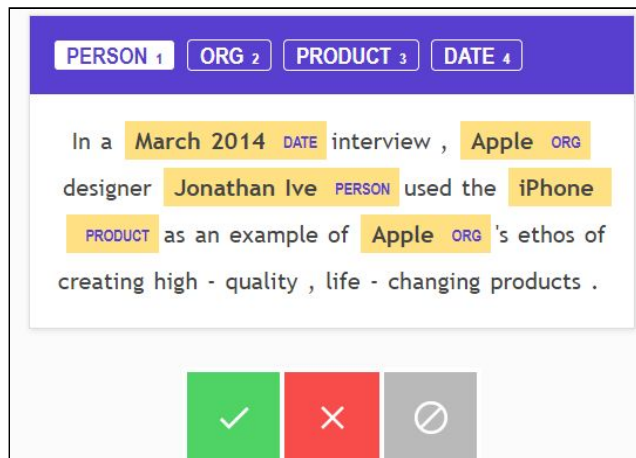


Figura 1: Tela de classificação da ferramenta *prodi.gy*⁶.

A ferramenta com maior proximidade em relação às funcionalidades é a *Prodi.gy*, que possibilita além da anotação textual, a classificação textual e de imagens. Essa ferramenta permite ainda a inserção de dados através de arquivos CSV, porém os dados devem ser estruturados de acordo com um modelo disponibilizado, não viabilizando a exibição de mais de uma característica de um vetor para classificar.

Por fim, nenhuma das ferramentas é adaptável às funcionalidades ou completamente gratuita, algumas chegando a custar US\$ 1.200 (mil e duzentos dólares) por mês como é o caso da *LightTag* e a licença mais barata da *prodi.gy* custa \$390 (trezentos e noventa dólares).

3. Metodologia

Nesta seção serão apresentadas as funcionalidades almejadas para a aplicação que será desenvolvida.

3.1 Funcionalidades

No intuito de atingir o objetivo do presente trabalho foram definidos requisitos para ferramenta de classificação textual:

1. Possibilitar o carregamento e exportação de dados através de arquivos CSV – *Comma-Separated Values* – em português, valores separados por vírgula;
2. Permitir o agrupamento de dados para classificação (sendo possível exibir mais de uma coluna para cada linha do CSV);
3. Não limitar a quantidade de rótulos;
4. Permitir que diferentes usuários classifiquem os mesmos dados adicionados;
5. Conceder flexibilidade em relação aos modelos de dados (cada projeto pode ter o objeto representado no CSV de um modo diferente).
6. Permitir portabilidade e distribuição gratuita para o público.

Na Seção 5 serão apresentadas as soluções utilizadas para desenvolver cada uma das funcionalidades.

⁶ <https://prodi.gy/>

4. Implementação

Para alcançar o objetivo deste trabalho foi desenvolvida uma aplicação web, na qual usuários possam inserir seu *dataset* e exibi-los para que diferentes usuários possam fazer a classificação manual destes dados.

Para o desenvolvimento da aplicação, foi escolhida uma arquitetura cliente-servidor, na qual os dados são processados pela camada de servidor e consumidos pela camada de cliente. A camada do servidor é composta por uma API – *Application Programming Interface* – desenvolvida com o *framework Express.js*, utilizando *Typescript* como linguagem e os dados são armazenados em um banco de dados *MongoDB*. Já a camada de cliente é composta por uma aplicação desenvolvida utilizando o *framework Angular 8*, esta camada é responsável por captar e exibir os dados buscados na API através de requisições HTTP – *Hypertext Transfer Protocol*.

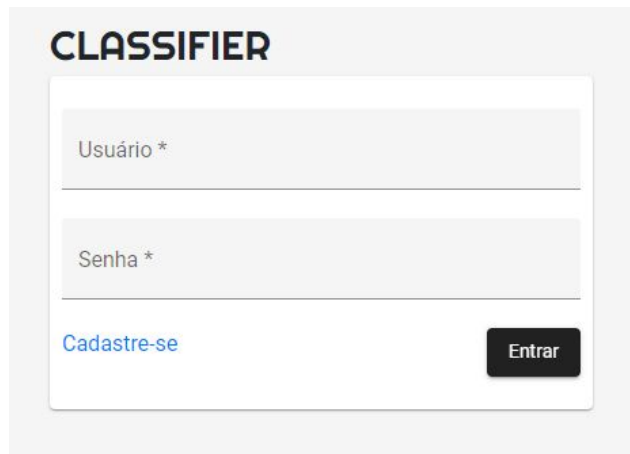
4.1 Rotas da API

Nesta subseção são apresentadas as principais rotas da API, que são os pontos de acesso disponibilizados para que o cliente possa acessar os recursos desenvolvidos no *backend*. A tabela a seguir contém a lista as rotas, mostrando qual endereço deve ser acessado, o método HTTP e o objetivo esperado ao acessá-la.

Endereço	Método	Objetivo
auth/register	POST	Cadastrar usuário.
auth/login	POST	Logar o usuário no sistema.
/project	POST	Cadastrar um novo projeto.
/project	GET	Buscar a lista de projetos cadastrados.
/project/:id	GET	Buscar um projeto específico pelo seu identificador (id).
/file/upload/csv	POST	Fazer o upload do arquivo CSV.
/project/:id/data	GET	Buscar os dados de um projeto para exibição e classificação.
/project/:id/labeling	POST	Enviar os dados classificados para a API.

4.2 Interface

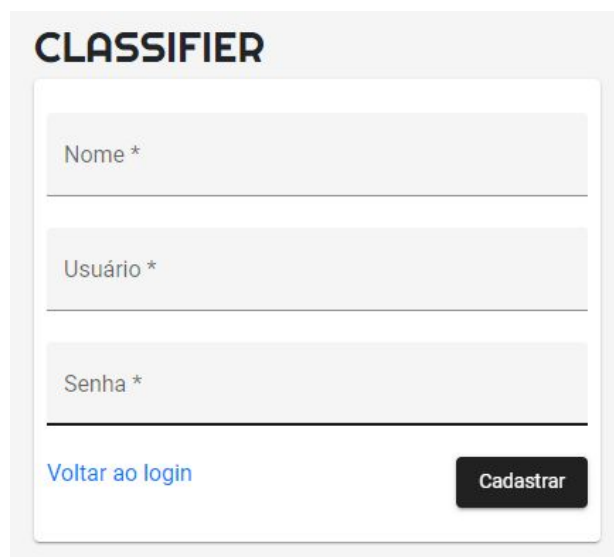
A primeira tela exibida ao abrir o sistema é a tela de login (Figura 3), caso o usuário já esteja registrado, basta preencher os nome de usuário e senha para logar no sistema. Senão, pode-se clicar em “Cadastre-se” para se registrar.



The image shows a login form titled "CLASSIFIER". It contains two input fields: "Usuário *" and "Senha *". Below the "Usuário *" field is a blue link "Cadastre-se". To the right of the "Senha *" field is a dark button labeled "Entrar".

Figura 3: Tela de login.

Na tela de cadastro de usuário (Figura 4), o usuário deverá preencher seu nome, o nome de usuário e uma senha para se cadastrar no sistema. Após finalizar o cadastro, o usuário deve clicar em “Voltar ao login” para logar. Após o login o sistema irá redirecionar para a tela de listagem de projetos (Figura 5).



The image shows a user registration form titled "CLASSIFIER". It contains three input fields: "Nome *", "Usuário *", and "Senha *". Below the "Nome *" field is a blue link "Voltar ao login". To the right of the "Senha *" field is a dark button labeled "Cadastrar".

Figura 4: Tela de cadastro de usuário.

Na tela de listagem de projetos, mostrada na Figura 5, são exibidos para o usuário o título e a descrição de projetos cadastrados e as opções de classificar ou fazer o download (em formato CSV) dos dados do projeto.

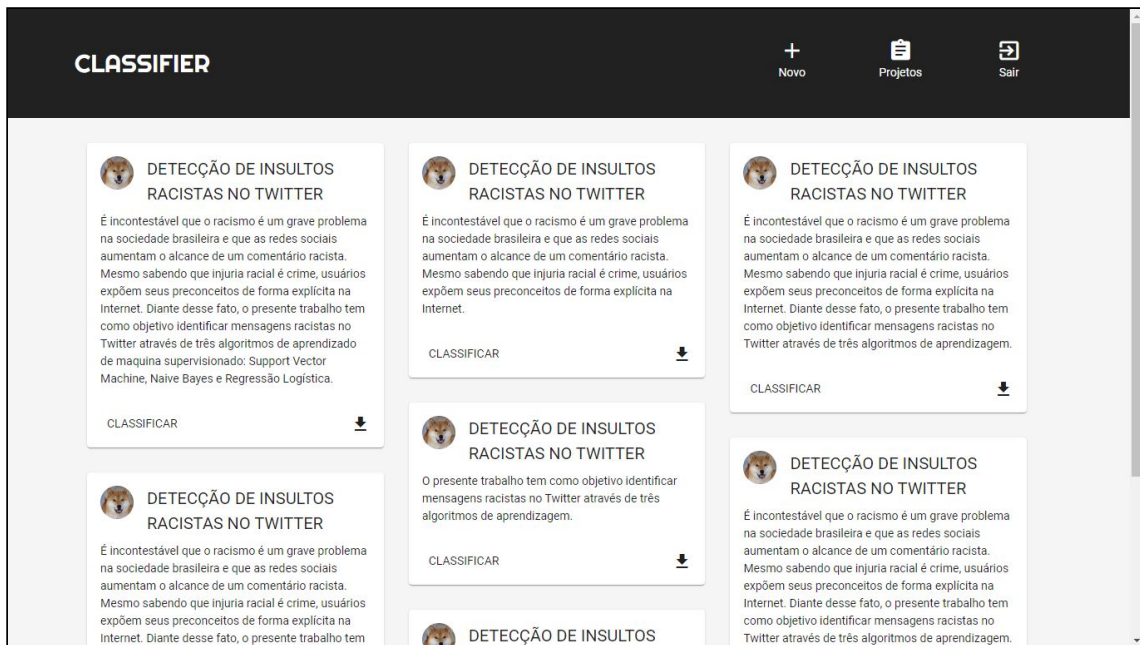


Figura 5: Tela de listagem de projetos.

Para classificar os dados de um projeto, o usuário deverá clicar em “Classificar” no *card*⁷ que representa este projeto. O *download* dos dados classificados pode ser feito clicando no ícone de seta, localizado no canto inferior direito do *card*.

Caso o usuário deseje criar um novo projeto, deverá clicar em “+ Novo” na barra de navegação. Então será direcionado para a tela de criação de projetos, onde é exibido um formulário para o cadastro de novos projetos na ferramenta. Este é dividido em três etapas.

Na primeira etapa (Figura 6) o usuário deverá inserir os dados de informação do projeto: título e descrição.

⁷ Elemento interativo contendo informações de um assunto apresentados quase sempre num formato retangular.

Figura 6: Primeira etapa do cadastro de projeto.

Na segunda etapa (Figura 7) é necessário fazer o *upload* do arquivo CSV contendo os dados que deverão ser classificados. Após o *upload* do arquivo, deverão ser selecionadas quais das colunas do arquivo serão exibidas para classificação. A coluna é composta por nome e propriedade. O nome será exibido junto ao dado na tela de classificação (Figura 9) e a propriedade se refere ao nome da coluna, coletado na primeira linha do arquivo CSV.

Figura 7: Segunda etapa do cadastro de projeto.

Por fim, na terceira etapa (Figura 8) o usuário deverá listar quais os rótulos estarão disponíveis para que seja feita a classificação dos dados. Para a criação do rótulo é necessário atribuir um nome, que será exibido no botão na tela de classificação, uma chave que será utilizada para criar uma nova coluna no arquivo CSV ao exportar. Esta coluna será preenchida pelo valor referente ao rótulo, atribuído pelo usuário em sua criação.

Figura 8: Terceira etapa do cadastro de projeto.

Por fim a tela de classificação é responsável por exibir os dados e rótulos para a classificação. Como mostrado na Figura 9, cada dado a ser rotulado será apresentado à

esquerda, estruturado de acordo com as colunas selecionadas no cadastro do projeto e à direita estarão botões representando os rótulos.



Figura 9: Tela de classificação.

5. Resultados

Nesta seção serão pontuadas, novamente, as funcionalidades desejadas, descritas na subseção 3.1, e quais soluções foram adotadas para cumprir cada requisito.

1. Importação e exportação de dados via CSV.

Para a importação dos arquivos, foi criado um componente (*FileUploadComponent*⁸) no frontend, que utiliza o *ng2-file-upload*⁹ para enviar arquivos à API da ferramenta. No *backend* foi criado um serviço (*CsvService*¹⁰) que utiliza a biblioteca *csv-parser*¹¹ para conversão do arquivo para objeto.

2. Agrupamento de dados para classificação;

Durante a criação do projeto, ao fazer *upload* do arquivo com os dados, a *API* da ferramenta irá retornar quais colunas de dados o arquivo possui. Essas colunas serão exibidas em um *select*¹² no formulário para que o usuário selecione quais delas deverão ser exibidas durante a classificação, nomeando-as caso ache necessário. Na tela de classificação serão exibidas as colunas selecionadas, com seu nome em negrito seguido do dado representado por ela.

3. Não limitar a quantidade de rótulos;

As ferramentas já existentes, pagas, cobram mais de acordo com a quantidade de dados que o usuário deseja classificar ou rótulos usados para na classificação. Para a

⁸ Disponível em <<http://tiny.cc/v2kjmz>>. Acesso em março de 2020.

⁹ Disponível em <<https://www.npmjs.com/package/ng2-file-upload>>. Acesso em março de 2020.

¹⁰ Disponível em <<http://tiny.cc/h3kjmz>>. Acesso em março de 2020.

¹¹ Disponível em <<https://www.npmjs.com/package/csv-parser>>. Acesso em março de 2020.

¹² Elemento HTML que apresenta um menu de opções.

contemplação deste requisito, na criação do projeto, o usuário pode criar quantos rótulos achar necessário.

4. Classificação por diferentes usuários.

Para que a seja feita a classificação dos dados, o usuário classificador deverá estar logado na ferramenta. Para isso foram criados, no *frontend* e *backend*, serviços para cadastro e login do usuário. Ao classificar um dados, a classificação será composta pelos identificadores do rótulo, do dado e do usuário que o classificou.

5. Conceder flexibilidade em relação aos modelos de dados.

Cada projeto pode ter uma diferente estrutura de dado, como por exemplo, um projeto com um *dataset* de *tweets* possui características diferentes de um *dataset* que representa um conjunto de filmes. Para que os dados de ambos os projetos possam ser armazenados e classificados, foi utilizado um banco não relacional, o *MongoDB*. Este tipo de banco permite o armazenamento de dados sem uma definição prévia de sua estrutura.

6. Portabilidade e distribuição gratuita.

Para a portabilidade, por ser um sistema *web*, a ferramenta poderá ser acessada através de um navegador, em qualquer dispositivo. Quanto a distribuição gratuita, os repositório,s com os códigos da ferramenta, podem ser encontrados no *BitBucket*¹³ nos seguintes endereços:

- *Frontend*: <https://bitbucket.org/adysonl/classifier-client>
- *Backend*: <https://bitbucket.org/adysonl/classifier-api>

Para o teste e validação das funcionalidades desenvolvidas, foi manuseado o *dataset* de *tweets* possivelmente racistas coletados no trabalho “Detecção de Insultos Racistas no Twitter” (2019). Este trabalho teve como objetivo o treinamento de três algoritmos de aprendizagem de máquina supervisionada para identificação de mensagens racistas.

Das colunas disponíveis (name, text, image, followers, location, entre outras) nos dados da coleta, foram selecionadas para exibição o nome do usuário (username) e o conteúdo textual do tweet (*text*) que poderiam ser rotulados como racista, não racista e neutro.

6. Considerações finais

Neste trabalho foi apresentada uma ferramenta desenvolvida com o intuito de facilitar a atribuição de rótulos para dados que serão utilizados em treinamento de classificadores de aprendizagem de máquina supervisionada. Nela, usuários criam projetos adicionando dados através de arquivos CSV, selecionam as colunas do arquivo que devem ser exibidas e os rótulos que podem ser atribuídos aos dados na classificação. É possível também fazer o download dos dados com seus respectivos rótulos após serem classificados.

Para trabalhos futuros, é necessária a evolução e avaliação da ferramenta nas perspectivas de usabilidade e segurança. Tratar e exibir melhor erros tantos de interface quanto vindos da API, utilizar o padrão JWT – *JSON Web Token* – para autenticação de

¹³ Serviço de hospedagem de projetos, similar ao Github.

usuários e proteção de rotas. Além disso, podem ser acrescentadas novas funcionalidades, como exibição e análise dos dados relacionados à classificação, permitir a anotação textual e classificação de outros tipos de dados (imagens, áudios, etc).

Referências

Yuri Malheiros. “Emotte: Uma Ferramenta De Análise de Sentimentos para o Twitter”, Em: “Anais Estendidos do XX Simpósio Brasileiro de Sistemas Multimídia e Web”, p. 62-65.

Araújo, Kevin Martins. Utilização Do Algoritmo De Máquina De Vetores De Suporte (Svm) Para Predição De Dados Climáticos. 2015. 96 f. Monografia (Especialização) - Curso de Ciência da Computação, Centro Universitário Luterano de Palmas, Palmas, 2015.

Ewerton Paulo da Silva e Yuri Malheiros. “Um Conjunto de Dados Extraído do Twitter para Análise de Sentimentos na Língua Portuguesa”, <https://si.dcx.ufpb.br/wp-content/uploads/2015/12/EWERTON-PAULO-DA-SILVA.pdf>, Abril, 2019.

Gabriela Vitória Ferraro de Andrade Sudo. “Detecção de Insultos Racistas no Twitter.”, <https://si.dcx.ufpb.br/wp-content/uploads/2015/12/GABRIELA-SUDO.pdf>. Acesso em março de 2020.

Liu, B. Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers, 2012.

Mika V. Mäntylä, Daniel Graziotin, Miikka Kuuttila. “The Evolution of Sentiment Analysis - A Review of Research Topics, Venues, and Top Cited Papers”. <https://arxiv.org/ftp/arxiv/papers/1612/1612.01556.pdf>, Abril, 2019.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? sentiment classification using machine learning techniques. In Proceedings of EMNLP, pages 79–86.

Peter Norvig, Stuart Russell. Inteligência Artificial: Tradução da 3ª Edição, volume 1. Elsevier Brasil, 2014.

Mohammad, S., Turney, P. D. (2013). Crowdsourcing a word-emotion association lexicon. Computational Intelligence, 29.