UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO”
Faculdade de Ciências
Campus de Bauru – SP

PEDRO HENRIQUE MOTA SILVA
VINÍCIUS CAMARGO DA SILVA

SENTTRACK: APLICAÇÃO PARA VISUALIZAÇÃO E ANÁLISE DE SENTIMENTOS SOBRE TÓPICOS DO TWITTER

Bauru – SP
2019

PEDRO HENRIQUE MOTA SILVA
VINÍCIUS CAMARGO DA SILVA

SENTTRACK: APLICAÇÃO PARA VISUALIZAÇÃO E ANÁLISE DE SENTIMENTOS SOBRE TÓPICOS DO TWITTER

Trabalho de Conclusão de Curso apresentado à Universidade Estadual Paulista "Júlio de Mesquita Filho" como parte das exigências para obtenção do título de bacharel em Sistemas de Informação.

Orientador: Prof. Dr. João Paulo Papa

Bauru – SP
2019

RESUMO

A utilização da internet para o compartilhamento de conteúdos variados, em especial no contexto das mídias sociais, tem escalado rapidamente. Com o crescente número de usuários e, consequentemente, de publicações, uma massiva intensificação no volume de dados gerados e disponibilizados na internet se formou, contribuindo muito para o avanço de diversas áreas. Graças às vastas fontes de texto disponíveis e o aprimoramento de técnicas de inteligência artificial, o Processamento de Linguagem Natural (PLN) foi muito beneficiado. Em paralelo, termos como visualização de dados e *analytics* se tornaram extremamente importantes no mercado, culminando, inclusive, na ascensão de novas carreiras, como Ciência de Dados e *Business Intelligence*. A opinião do indivíduo disseminada nas redes se tornou um indicador valioso para o sucesso de negócios, corroborando para o aumento do interesse em tarefas como a mineração de opinião (ou análise de sentimentos). Dentro desse contexto, o SentTrack se propõe a possibilitar uma forma fácil e rápida para coleta, visualização e análise de publicações veiculadas na rede social Twitter. A ideia é que o usuário da aplicação possa interagir com um *dashboard* que, com a colaboração de modelos de Aprendizado de Máquina, possibilite uma análise concisa e clara dos sentimentos vinculados a tópicos em pauta na rede social.

Palavras-chave: análise de sentimentos, visualização de dados, redes sociais.

ABSTRACT

The use of the internet for sharing varied content, especially in the context of social media, has escalated rapidly. With the growing number of users and, consequently, of publications, a massive intensification in the volume of data generated and made available on the internet has formed, contributing a lot to the advancement of several areas. Thanks to the many text sources available and the enhancement of artificial intelligence techniques, Natural Language Processing (NLP) has greatly benefited. In parallel, terms such as data visualization and analytics have become extremely important in the marketplace, culminating even in the emergence of new careers such as Data Science and Business Intelligence. Individual opinion spread across networks has become a valuable indicator of business success, supporting the increased interest in tasks such as opinion mining (or sentiment analysis). Within this context, SentTrack proposes to provide an easy and fast way to collect, view and analyze publications posted on the Twitter social network. The idea is that the user of the application may interact with a dashboard that, with the collaboration of Machine Learning models, enables a concise and clear analysis of sentiment linked to topics on the social network.

Keywords: sentiment analysis, data visualization, social media.

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

LISTA DE SIGLAS

PLN	Processamento de Linguagem Natural
RNN	Rede Neural Recorrente
GRU	*Gated Recurrent Unit*
PCA	*Principal component analysis*
t-SNE	*t-Distributed Stochastic Neighbor Embedding*

SUMÁRIO

1. INTRODUÇÃO

Hoje, dados de mídias sociais são muito importantes para companhias e organizações em geral (KURNIAWATI; SHANKS; BENKMAMEDOVA, 2013). Esses dados descrevem usuários, o que, na prática, significa descrever o comportamento humano (RUTHS; PFEFFER, 2014). Desse modo, esses dados aparecem como um componente interessante para a manutenção e o desenvolvimento de negócios, já que tem potencial para caracterizar o consumidor, o cliente e até mesmo a concorrência.

Isso tem impulsionado novos mercados e contribuiu para criação de carreiras que sequer existiam pouco tempo atrás, como, por exemplo, a de Cientista de Dados (PRESS, 2013).

Como apontado por diversos autores (MCAFEE et al., 2012; WU et al., 2014; LOHR, 2012), a sociedade encontra-se em uma era onde os dados são vastos e complexos. Nesse contexto, extrair informação útil através dessa densa camada de dados é imprescindível para empresas, governos e organizações em geral, onde informação é fundamental no processo de tomada de decisão.

Considerando a importância e o impacto dos dados no presente e a sua indiscutível crescente para o futuro, o trabalho apresenta uma ferramenta que pretende unir o estudo da opinião em mídias sociais, inteligência artificial e visualização de dados em uma aplicação baseada em web que, amigável, facilite o estudo e a exploração dentro desse contexto.

2. DETALHAMENTO DO PROBLEMA

Mídias sociais, cada vez mais, têm se tornado importantes fontes de dados. Nesse tipo de plataforma, o conteúdo gerado por seus usuários apresenta grande potencial para as organizações, já que, em geral, tem a capacidade de refletir seus padrões de comportamento e de interação, que auxiliam no melhor entendimento desses indivíduos e suas relações: "Podemos agora integrar teorias sociais com métodos computacionais para estudar como os indivíduos (também conhecidos como átomos sociais) interagem e como as comunidades (ou seja, as moléculas sociais) se formam." (ZAFARANI; ABBASI; LIU, 2014, tradução nossa).

Um tópico importante cujo estudo vem se popularizando com o crescimento das mídias sociais é a **opinião** do indivíduo e como quantificá-la. Nossas opiniões influenciam nossa percepção, interesses e decisões (LIU, 2012), portanto, conseguir analisar e quantificar esse conteúdo certamente seria uma mais-valia para qualquer organização ou governo.

A análise de sentimentos (ou mineração de opinião) – tópico do Processamento de Linguagem Natural que visa, na sua forma mais simples, a capacidade de discriminar documentos, sentenças ou textos em geral por sua polaridade através de análise automática – tem muito a contribuir nesse sentido. Nasukawa e Yi (2003) destacaram a importância de uma técnica como essa para as organizações, argumentando que teria a habilidade de prover funcionalidades poderosas, como reforçar análise competitiva, análise de marketing e a detecção de rumores desfavoráveis para gerenciamento de risco.

Hoje, o potencial desse estudo é enorme, considerando o volume e a qualidade dos dados que são gerados – em comparação aos de uma ou duas décadas atrás – e o grande avanço que o Processamento de Linguagem Natural (PLN) obteve recentemente, muito em função da ascensão dos modelos de Aprendizado de Máquina, especialmente os de Aprendizado Profundo (do inglês, *Deep Learning*) (LECUN; BENGIO; HINTON, 2015).

A maioria dos trabalhos acadêmicos dessa área estão focados na obtenção de modelos que consigam resolver o problema da maneira mais eficiente e acurada. No entanto, no contexto das mídias sociais, seria muito mais útil para um usuário qualquer, ao invés de ter somente um modelo com máxima acurácia, poder utilizar as polaridades que este é capaz de inferir para agregar valor junto a outras informações, como, por exemplo, tempo e assunto.

Uma forma comum de ajudar no entendimento de informações unindo diferentes variáveis, é usando gráficos e visualizações, em virtude de que, quando elaborados de forma

correta, nos permitem encontrar padrões e processar mentalmente grandes volumes de informação com naturalidade (WATE, 2012).

É nesse contexto que se insere o presente trabalho, cuja intenção é o introdução de uma ferramenta que permita entender melhor a opinião de usuários nas mídias sociais, fazendo uso de técnicas de Processamento de Linguagem Natural e análise de sentimentos, além de dispor de mecanismos gráficos e visualizações para estimular a compreensão. A ideia é que, através da ferramenta, o usuário consiga explorar e interagir com o conteúdo produzido por usuários do Twitter, somado a predições de sentimento geradas pelo sistema.

3. SOLUÇÕES EXISTENTES E OBJETIVOS ESPECÍFICOS DA PROPOSTA

Nesta seção serão apresentadas duas soluções já existentes no contexto do projeto, bem como os objetivos específicos do mesmo. O Quadro 1 compara a solução proposta pelo trabalho em relação às principais características das soluções já existentes, cujos detalhes são exibidos no sub-tópicos 3.1 e 3.2.

Quadro 1 - Comparação entre soluções existentes e a solução desenvolvida

Característica	Sentdex	Brand24	SentTrack
Visualização por assunto	Sim	Sim	Sim
Visualização por tempo	Sim	Sim	Sim
Lista de principais termos	Sim	Não	Sim
Visualização por localização	Sim	Não	Sim
Notificação por menção	Não	Sim	Não
Buscas personalizadas	Não	Sim[1]	Sim

Fonte: elaborado pelos autores

3.1. Sentdex

Sentdex[2] é uma plataforma online que armazena informação sobre como as pessoas, em escala global, se sentem com relação a vários tópicos, como finanças, política e assuntos gerais.

A plataforma apresenta os resultados de um algoritmo de Análise de Sentimento aplicado sobre diversas fontes, a depender do tópico em questão. São dezenas de fontes que contam com inúmeros portais de notícias, dentre outras.

Através de dados do Twitter, o website apresenta um infográfico baseado em geolocalização e no sentimento inferido pelo algoritmo, sendo esta a única seção da plataforma que faz uso direto de mídia social.

[1] Brand24 oferece mecanismos de busca por termos dentro de um número limitado de publicações, como publicações próprias ou menções do perfil do usuário.
[2] http://sentdex.com/

3.2. Brand24

Brand24[3] oferece um serviço robusto de monitoramento de mídias sociais. Através dele é possível acompanhar desde menções dos perfis do usuário a *hashtags* específicas, podendo-se observar o alcance das publicações, criar alertas e interagir com o conteúdo.

Através dessa aplicação também é possível gerar relatórios automáticos com estatísticas simples a respeito dos dados, além de monitorar um gráfico de volume de menções no tempo.

O serviço oferece filtros para a exploração das menções aos perfis do usuário, dentre os quais, o de polaridade de sentimento. No entanto, as visualizações relacionadas a sentimento são escassas.

3.2. Objetivos específicos

Embora hajam ferramentas interessantes centradas em mídias sociais e opinião pública (como Brand24 e Sentdex), não foram encontradas soluções com enfoque específico na visualização e na análise de sentimentos de mídias sociais.

Dessa forma, propõe-se um sistema baseado em serviço web capaz de coletar, armazenar e extrair informações sobre a polaridade da opinião pública a respeito de qualquer assunto disponível na rede social Twitter.

A ideia é que o sentimento seja explorado junto com informações inerentes aos tweets para a confecção de visualizações.

Para efeito de comparação, propomos uma ferramenta semelhante ao Brand24 com foco primário em *tweets* e análise de sentimentos, além de dispor de mais apelo visual e exploratório que o Sentdex.

[3] https://brand24.com/

4. FUNDAMENTOS E TÉCNICAS

Nesta seção são enumeradas e descritos os fundamentos teóricos que serviram de embasamento para implementação do trabalho, bem como as técnicas aplicadas na obtenção dos resultados.

4.1. Análise de sentimento de tweets

Diversos autores já abordaram a problemática da aplicação de análise de sentimentos no contexto das publicações de Twitter (AGARWAL et al. 2011; GO et al. 2009; KOULOUMPIS et al. 2011; PAK; PAROUBEK, 2010; SAIF et al. 2012), fornecendo desde metodologias de coleta e anotação dos dados a treinamento e otimização de modelos.

Embora não seja uma questão trivial, uma abordagem possível para a análise de sentimentos pode ser interpretá-la como um problema de classificação de texto, ou seja, um problema onde deseja-se classificar pedaços de texto em categorias pré-determinadas de acordo com um padrão de interesse. Uma outra abordagem seria a regressão, onde, para cada exemplar, ao invés de uma categoria, deseja-se inferir um escalar (por exemplo, qualquer valor entre -1 e 1, onde -1 significaria extremamente negativo e 1 extremamente positivo).

No caso da análise de sentimentos, normalmente opta-se pela primeira abordagem, uma vez que é menos subjetivo determinar, por exemplo, se um texto transmite um sentimento positivo, negativo ou neutro do que delimitar “quão” positivo (ou negativo), na forma de valor numérico.

As redes neurais ganharam muita atenção nesse contexto, especialmente com o advento do Aprendizado Profundo (LECUN; BENGIO; HINTON, 2015). Sendo assim, optou-se por usá-las no presente trabalho como parte do algoritmo de classificação.

4.2. Arquitetura neural

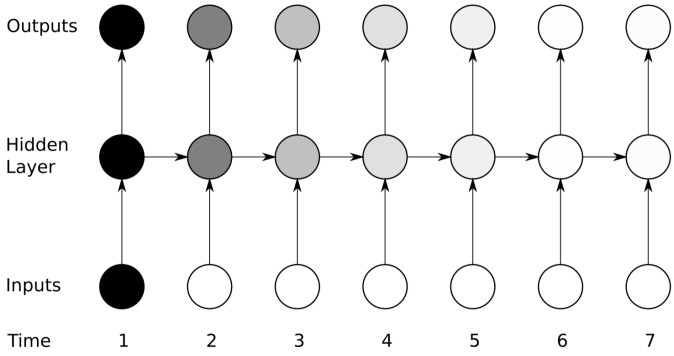
Usualmente, as arquiteturas mais tradicionais de redes neurais assumem independência entre diferentes pares de entrada e saída. Outra característica é que elas assumem que a dimensionalidade das entradas e saídas seja conhecida e fixa. Embora essa abordagem seja aplicável a muitos problemas, em diversas situações, especialmente onde há dependência

sequencial entre diferentes entradas (como quadros de um vídeo ou palavras de uma frase), essas arquiteturas não abrangem completamente o problema ou simplesmente não oferecem resultados notáveis. Dentro dessa problemática, surgem as Redes Neurais Recorrentes.

As Redes Neurais Recorrentes (do inglês, *Recurrent Neural Networks* ou RNNs) propõem uma metodologia que aplica uma mesma tarefa em cada um dos elementos de uma sequência, resultando em saídas que dependem de estados prévios da rede (HAYKIN, 1998).

A ideia intuitiva de uma RNN é que esta aprenda guardar as informações mais relevantes no decorrer da sequência (como num processo de memorização), usando-as para modelar a saída, como apresentado na Figura 1. Isso é muito interessante, já que é sensato imaginar que os diferentes instantes de uma sequência estão relacionados internamente e apresentam alguma dependência. Dentro dessa noção intuitiva, o estado da rede representam sua "memória".

Figura 1 - Modelo padrão de RNN



Fonte: Graves (2012)

O modelo utilizado no projeto é uma versão adaptada da utilizada por Yang et al. (2016), uma RNN com mecanismo de atenção para classificação de documentos, com a diferença de que não foi necessário mais de um nível de hierarquia. A metodologia faz uso de "duas RNNs" distintas, cada uma abordando a sequência de entrada em uma direção (arquitetura comumente referida como RNN bidirecional), sendo as unidades de um tipo especial, as *Gated Recurrent Units* (GRUs) (CHO et al., 2014).

4.2.1 GRU

O objetivo de uma GRU é permitir uma memória longa porém seletiva, que inibe a transição de informação desnecessária pelos estados da rede, sem que hajam grandes custos computacionais (se comparada a outros tipos de redes recorrentes).

Seja t um instante no tempo (neste caso, o da t-ésima palavra) e x_t seu *embedding*. Os parâmetros treináveis da GRU são denotados por W_* e b_* ($*$ denota a multiplicação elemento a elemento, $[A, B]$ a concatenação dos vetores A e B e σ a função de ativação *sigmoid*):

$$z_t = \sigma(W_z[h_{t-1}, x_t] + b_z) \tag{4.1}$$

$$r_t = \sigma(W_r[h_{t-1}, x_t] + b_r) \tag{4.2}$$

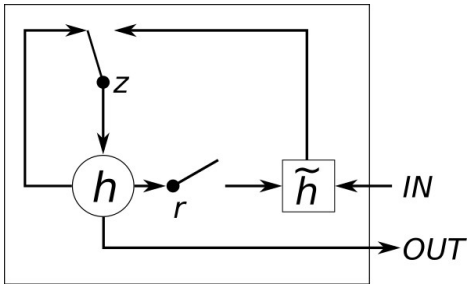
$$\tilde{h}_t = \tanh(W[r_t * h_{t-1}, x_t] + b) \tag{4.3}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{4.4}$$

onde h_t são os estados da rede.

A Figura 2 ilustra funcionamento interno de uma célula do tipo GRU.

Figura 2 - Célula GRU



Fonte: Chung et al. (2014)

4.2.2 Mecanismo de atenção

Como apontado por Yang et al. (2016), “nem todas as palavras contribuem igualmente para representar o sentido de uma sentença” (tradução nossa), assim, a proposta do mecanismo de atenção é auxiliar na performance de RNNs, especialmente ao lidar com sentenças mais longas, oferecendo comprovados incrementos de performance. Inspirada por

Bahdanau et al. (2014), a formulação utilizada por Yang et al (2016) e aplicada neste trabalho é:

$$u_t = V_a^T \tanh(W_a h_{t-1} + b_a) \tag{4.5}$$

$$a_t = \frac{e^{u_t}}{\sum_i e^{u_i}} \tag{4.6}$$

$$v_s = \sum_t a_t h_t \tag{4.7}$$

A ideia é que os alinhamentos a_t auxiliem a rede a decidir em quais estados no tempo, e concomitantemente, quais elementos da sentença de entrada, concentrar-se durante as ativações.

Depois de obtido o vetor v_s os *logits* são obtidos através dos parâmetros W_s e b_s:

$$l = W_s v_s + b_s \tag{4.8}$$

Por fim, a probabilidade de que a sentença esteja associada à categoria i, $i \in K$, é calculada:

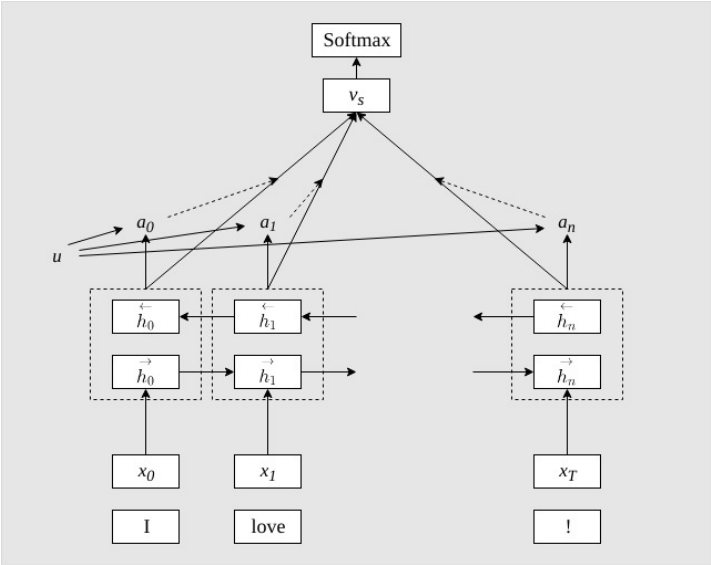
$$p_i = \frac{e^{l_i}}{\sum_j e^{l_j}} \tag{4.9}$$

Note que, na etapa de treinamento, os parâmetros da rede são treinados para minimizar a função de custo J, conhecida como entropia:

$$J(\theta) = \sum_{i=1}^{|K|} p_i \log(p_i) \tag{4.10}$$

onde $|K|$ é o número de categorias.

Figura 3 - Rede neural recorrente com mecanismo de atenção



Fonte: elaborada pelos autores.

A Figura 3 representa a arquitetura utilizada no trabalho. Embora hajam arquiteturas possivelmente mais acuradas disponíveis para a solução do problema (DEVLIN et al., 2018; LAN et al., 2019; YANG et al., 2019), a ideia foi contar um classificador que não requeresse um alto poder computacional para os padrões da aplicação.

4.2. GloVe

Os vetores GloVe surgem com a proposta de capturar, através de representações vetoriais, características semânticas e sintáticas de palavras ou termos (PENNINGTON; SOCHER; MANNING, 2014). Uma de suas maiores contribuições é a forma simples como eles são computados.

Uma das características do GloVe é que a distância (euclidiana ou cossenoidal) entre dois vetores, normalmente é uma medida efetiva de similaridade semântica ou linguística das palavras correspondentes. Outra característica interessante é a operacionalização do relacionamento entre as palavras. Por exemplo, o vetor resultante da operação entre vetores GloVe (rei - rainha) é aproximadamente o mesmo de (homem - mulher).

Os vetores são obtidos através de treinamento usando o algoritmo do gradiente descendente, minimizando a equação 4.11:

$$J = \sum_{i=1}^{V} \sum_{j=1}^{V} f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij})^2 \tag{4.11}$$

onde X é matriz de coocorrência entre os termos do vocabulário de tamanho V.

Depois do treinamento, os autores utilizam a soma $w_i + \tilde{w}_i$ como a representação vetorial do termo i.

4.3. Redução de dimensionalidade

Nesta seção, encontram-se os detalhes sobre duas técnicas de redução de dimensionalidade para aplicações em problemas de visualização da informação: o PCA (*Principal Component Analysis*) e o t-SNE (*t-Distributed Stochastic Neighbor Embedding*).

4.3.1 PCA

O PCA (Principal Component Analysis) é um método clássico de redução de dimensionalidade através do qual é efetuada a transformação de um conjunto original de dados em um novo conjunto reduzido de variáveis que sintetiza as principais características extraídas dos dados (YEUNG; RUZZO, 2001).

Os principais objetivos do PCA são a extração das informações mais relevantes dentro de um conjunto de dados, a redução no tamanho de tal conjunto e a análise das informações extraídas nas novas variáveis. Para desempenhar tal função, o PCA encontra, através de combinações lineares das variáveis originais, o valor de novas variáveis chamadas de "componentes principais" (ABDI; WILLIAMS, 2010).

4.3.2 t-SNE

Diferente do PCA, o t-SNE (*t-Distributed Stochastic Neighbor Embedding*) consegue capturar dependências não-lineares. Ele usa os relacionamentos locais entre pontos para criar um mapeamento de baixa dimensão, o que permite essa captura (MAATEN; HINTON, 2008).

A ideia central do t-SNE é minimizar a divergência entre a distribuição que mede semelhanças em pares dos objetos na dimensão original e uma distribuição que mede

semelhanças em pares dos pontos de dimensão resultante correspondentes. A tentativa é de que pontos localmente próximos no espaço vetorial de alta dimensão continuem proporcionalmente distantes em baixa-dimensão.

Em geral, visualizações produzidas usando t-SNE são consideravelmente melhores do que as produzidas usando outras técnicas (MAATEN; HINTON, 2008).

4.4. Clusterização e K-Means

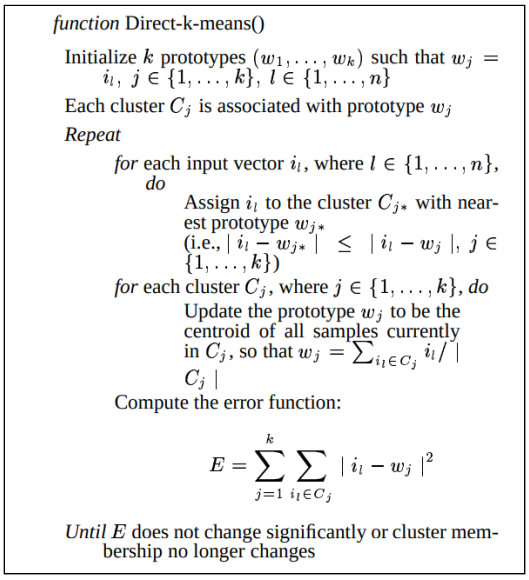
A clusterização é uma técnica muito empregada em aplicações de análise e visualização de informação. Por meio dela, é possível distinguir a existência de diferentes agrupamentos dentro de um *dataset*, o que proporciona uma visualização de elementos correlacionados por algum atributo.

Segundo Alsabti et al. (1997), o processo de clusterização é definido como uma forma de particionar padrões de dados em conjuntos ou *clusters*. De tal maneira, os padrões de dados dentro de um cluster são iguais, enquanto que os padrões entre dois clusters são diferentes.

O K-means é um dos métodos mais utilizados no processo de clusterização e consiste em particionar um conjunto de dados em k grupos. Seguindo seu algoritmo, inicialmente define-se quais serão os k centros de clusters. Tais centros são, então, refinados iterativamente conforme a seguinte lógica: cada valor atribuído para o cluster mais próximo com centro mais próximo; cada centro de cluster é atualizado de forma que seja a média dos componentes do cluster. A conversão do algoritmo ocorre quando não há mais mudanças possíveis a serem realizadas na atribuição de componentes a clusters (WAGSTAFF et al., 2001).

A Figura 4 apresenta o pseudocódigo do algoritmo de refinamento do K-means, bem como sua função de cálculo de erro.

Figura 4 - Algoritmo K-means



Fonte: Alsabti (1997)

O principal problema do algoritmo K-means é a seleção do número *k* de clusters que irão compor o problema. Múltiplas abordagens foram elaboradas com o intuito de se chegar a uma valor ótimo para *k*. As duas mais conhecidas são: *Elbow* e *Silhouette*.

O método *Elbow* é uma abordagem visual e consiste de iniciar o valor de *k* em 2 e incrementá-lo iterativamente, encontrando os clusters e calculando o custo do treinamento. A partir de um certo valor, o resultado da função de custo tem uma queda drástica e seu gráfico começa a apresentar uma forma de planalto, o que indica o valor mais próximo do ideal para *k* (KODINARIYA; MAKWANA, 2013).

O método *Silhouette*, consiste na diferença entre a o grau de proximidade dos elementos em um cluster a o grau de separação em relação aos elementos de outros clusters, isto é, sob uma perspectiva gráfica, consiste na formação de “silhuetas” de grupos. O método é definido pela seguinte fórmula:

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))} \tag{4.12}$$

Na qual i é cada elemento do conjunto de dados sendo analisado; $a(i)$ representa a distância média entre i e cada elemento pertencente ao mesmo cluster de i, $b(i)$ representa o valor mínimo das distâncias entre o elemento i e cada um dos elementos dos outros clusters. O valor de *s* pode variar entre -1 e 1. Quanto mais próximo de -1, maiores as probabilidades

de o elemento estar no cluster errado. Quando mais de próximo de 1, maiores as chances de a clusterização estar correta. Um resultado próximo de 0 indica que o elemento poderia ser alocado tanto para um cluster quanto para outro (KODINARIYA; MAKWANA, 2013).

4.5. Protocolo WebSocket

O protocolo WebSocket é muito utilizado na comunicação cliente-servidor de aplicações com atualização em tempo-real. Sua estrutura permite que, enquanto uma conexão estiver estabelecida entre cliente e servidor, mensagens possam ser trocadas entre estes a qualquer momento e forma assíncrona, suprimindo a necessidade de reconexão a cada envio de dados (WESSELS et al., 2010). Por se tratar de uma comunicação assíncrona, as trocas de mensagens não são bloqueantes para o processamento de ambos os lados, proporcionando melhor performance em aplicações *real-time*.

Alternativamente, outra técnica de coleta de dados e comunicação em tempo real entre cliente e servidor é a chamada *HTTP Polling*, na qual um cliente dispara, periodicamente, mensagens de requisição para um servidor. Este, por sua vez, responde com uma mensagem portando os dados solicitados. Entretanto, a necessidade de definir, a cada nova requisição, quais serão os *headers* da mensagem enviada, pode ocasionar sobrecarga na comunicação (PIMENTEL; NICKERSON, 2012).

4.6. Banco de dados

Nesta seção apresentam-se os conceitos do modelo de banco de dados aplicado no módulo de armazenamento do sistema apresentado: o NoSQL.

Os bancos de dados NoSQL (*Not only SQL*) são bases não relacionais - ou seja, não se baseiam no conceito de relações (tabelas) - projetadas para o armazenamento de grande quantidade de informações, bem como processamento paralelo massivo de dados. Tais bases de dados oferecem suporte a atividades de caráter analítico, exploratório e preditivo . Como o nome do modelo já indica, a maioria de suas bases de dados não faz utilização da linguagem SQL (*Structured Query Language*) - comum no modelo relacional - para atualização ou leitura dos dados (MONIRUZZAMAN; HOSSAIN, 2013).

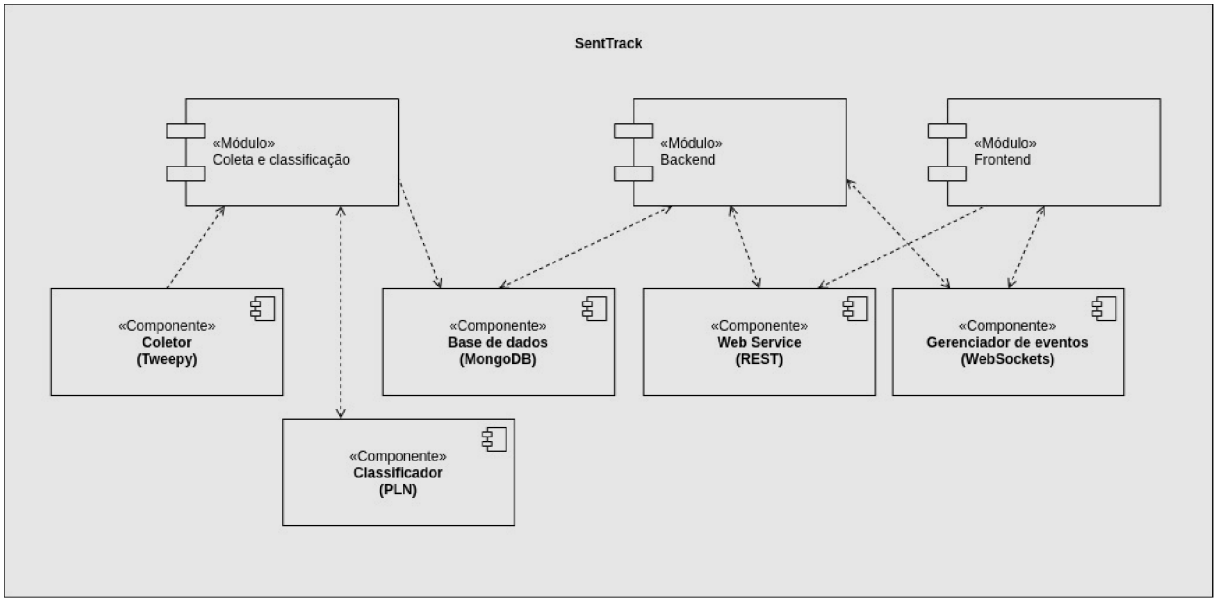
Os bancos de dados NoSQL podem ser classificados em 5 tipos, de acordo com suas respectivas metodologias de armazenamento e recuperação de dados: chave-valor, orientado a colunas, documentos, grafos, orientado a objetos (NAYAK; PORIYA; POOJARY, 2013).

5. ESPECIFICAÇÃO DO SISTEMA

Nesta seção, apresentam-se os detalhes de especificação do sistema desenvolvido, bem como diagramas que melhor demonstram quais são os módulos que compõem a plataforma e como tais módulos interagem entre si para que o objetivo do software seja atingido.

Na Figura 5 apresenta-se, em um diagrama, a disposição e comunicação dos módulos que fazem parte da plataforma juntamente com seus componentes.

Figura 5 - Diagrama de componentes



Fonte: elaborada pelos autores

Como pode-se verificar no diagrama de componentes (Figura 5), o sistema é composto por três módulos principais: Coleta e Classificação, Backend, Frontend.

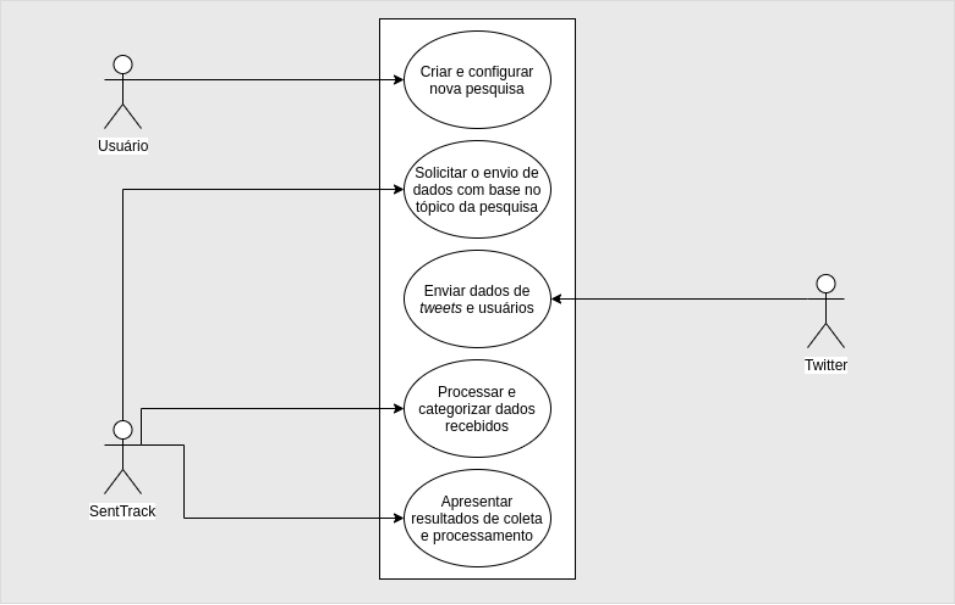
O módulo de Coleta e Classificação é o responsável por interceptar os dados enviados pelo Twitter, aplicar os procedimentos de classificação, redução de dimensionalidade e clusterização e armazenar tanto os resultados destes processamentos quanto os dados originais recebidos.

O módulo de *Frontend* é fica responsável por receber os comandos e interações do usuário e disparar as respectivas ações no sistema. Este módulo dispara requisições para o módulo de *Backend* e recebe os dados processados e organizados para apresentação.

O módulo de Backend fica encarregado por atuar como intermediador entre os demais módulos. Por um lado, recebe requisições disparadas pelo *Frontend*, faz consultas no sistema de armazenamento e responde a tais requisições com os dados processados. Por outro lado, comunica-se com o módulo de Coleta e Classificação emitindo requisições para iniciar e/ou finalizar a coleta de informações. Este módulo também é responsável por interceptar modificações realizadas no banco de dados e notificar tais mudanças ao *Frontend* de maneira que as informações apresentadas estejam sempre atualizadas.

Na Figura 6 a seguir, pode-se verificar as interações entre os atores que compõem o funcionamento do sistema.

Figura 6 - Diagrama de casos de uso

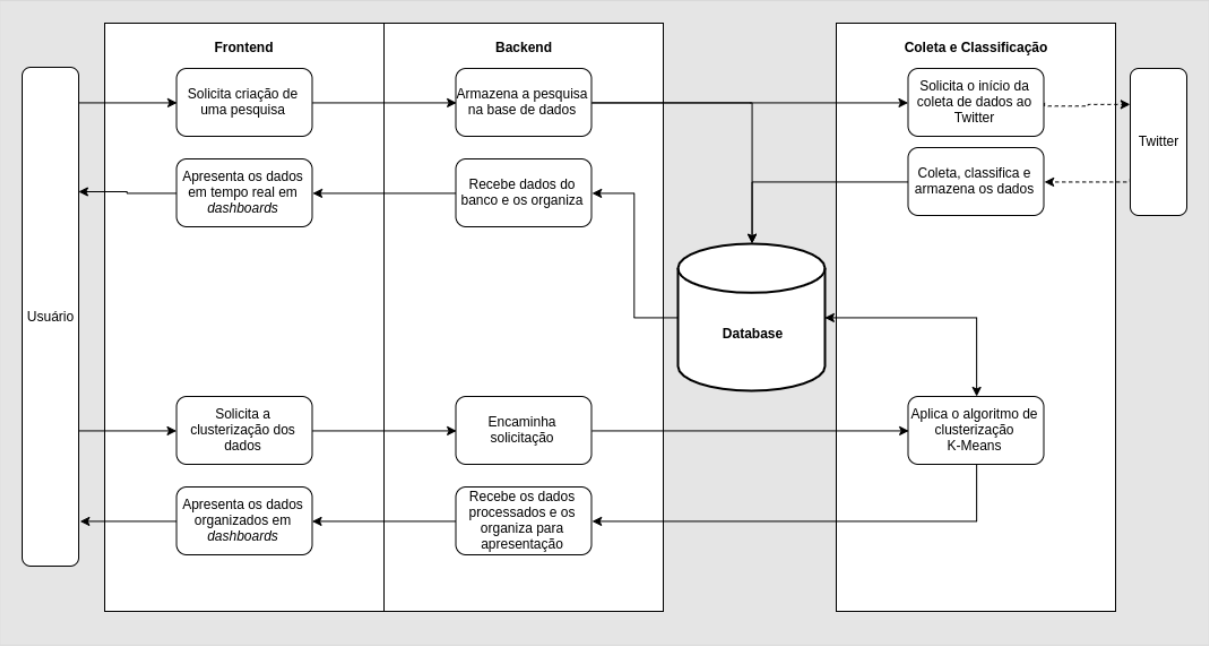


Fonte: elaborado pelos autores

Como é possível verificar na Figura 6, o sistema interage com dois atores: o usuário final e o Twitter, atuando com agente intermediário entre os demais.

A Figura 7 a seguir apresenta o fluxo de funcionamento, exibindo o fluxo de troca de mensagens entre os três módulos do sistema.

Figura 7 - Fluxo de funcionamento entre os módulos do sistema



Fonte: elaborado pelos autores

6. IMPLEMENTAÇÃO DO SISTEMA

Nesta seção discute-se a elaboração e organização dos módulos e componentes do sistema, aprofundando-se em um detalhamento mais técnico.

6.1. Coleta de dados

O Twitter disponibiliza a utilização de seus serviços para consulta de informações em dois formatos: REST *web service* e *streaming*.

No formato de REST *web service*, o cliente (SentTrack) deveria fazer requisições - seguindo o protocolo HTTP - para os servidores da rede social, os quais responderiam com as informações solicitadas no formato JSON (*JavaScript Object Notation*)[4].

Com o formato de *streaming*, o SentTrack estabeleceria uma conexão com com os serviços do Twitter através de uma requisição HTTP inicial. A partir deste momento, o Twitter iniciaria o envio automático de informações para o SentTrack a cada atualização, isto é, a cada novo dado gerado.

Seguindo a proposta inicial de desenvolver uma plataforma com *dashboard* atualizado em tempo real, optou-se pela seleção do formato *streaming*.

Para facilitar a conexão, criação de sessão e interceptação de mensagens do Twitter, utilizou-se a biblioteca Tweepy[5], com implementação na linguagem de programação Python. Dentro do SentTrack, o Tweepy fica encarregado de notificar a rotina de recebimento e processamento de dados a cada novo evento recebido.

6.2. Pré-processamento

Toda vez que um novo *tweet* chega ao sistema – e antes de ser inserido no classificador descrito na seção 6.3 – é submetido ao pré-processamento dos dados. Embora o sistema mantenha uma cópia do texto original na base de dados, este precisa passar pela fase de engenharia de características. Esta fase consiste na extração de características relevantes do texto a fim de serem utilizadas pelo modelo de análise de sentimento.

[4] https://developer.twitter.com/en/docs/api-reference-index
[5] https://www.tweepy.org/

Nesta etapa as URLs e nomes de usuários são removidos do texto, o texto é minificado (todos os caracteres são convertidos para caixa-baixa), números são retirados e o texto é normalizado (as palavras “wooooooow” e “woooow”, por exemplo, são tratadas como a mesma). Tudo isso foi implementado através de expressões regulares.

Depois disso, as sentenças passam por tokenização, onde palavras e pontuação são individualizados, desse modo, cada sentença é transformada em uma lista de *tokens*. A partir dessa lista, extrai-se uma representação para o texto na forma de um vetores numéricos, que, esperançosamente, contém suas informações sintáticas e, principalmente, semânticas. Os vetores são os *Global Vectors* (PENNINGTON; SOCHER; MANNING, 2014) dos *tokens* presentes no texto. Cabe ressaltar que os vetores GloVe utilizados no projeto foram pré-treinados em bases compostas por 2 bilhões de tweets[6].

Além de armazenados, os vetores GloVe obtidos são aproveitados pelas técnicas de redução de dimensionalidade t-SNE (*t-Distributed Stochastic Neighbor Embedding*) e PCA (*Principal Component Analysis*), resultando em novos vetores de duas dimensões que serão utilizados em etapas posteriores para confecção de visualizações.

A etapa de pré-processamento dos *tweets* foi desempenhada com o auxílio da biblioteca NLTK, muito popular nesse contexto.

6.3. Classificação

Após os vetores GloVe serem obtidos, o classificador treinado os usa para fazer uma predição de sentimento, que é salva no banco de dados.

O classificador trata-se de uma rede neural construída segundo a arquitetura descrita na seção 4.2 e cujos os detalhes de metodologia são descritos da seção 6.3.1 à seção 6.3.4.

6.3.1 Base de dados

Quando se fala em redes neurais e classificação uma questão muito importante a se considerar é a base de dados. No aprendizado supervisionado, a base de dados é a componente mais importante da etapa de treinamento. Ela é composta pelos exemplos sobre os quais o modelo desenvolverá suas habilidades de classificação.

[6] https://nlp.stanford.edu/projects/glove/

Durante o desenvolvimento do projeto, poucas bases públicas (no contexto dos tweets e análise de sentimentos) foram encontradas, especialmente bases de contexto aberto – onde se insere a problemática do trabalho, já que a ferramenta não é limitada a um único nicho de assuntos.

Culminou-se por utilizar uma base[7] que contém 1.6 milhão de tweets, dividida em base de treino (com tweets positivos e negativos, utilizando a metodologia descrita por GO et al. 2009) e base de teste (com 498 tweets entre positivos, negativos e neutros, manualmente anotados).

Uma dificuldade encontrada foi que, a base de treino mencionada, além de ruidosa (pela forma como é anotada), não representa completamente a realidade, já que não dispõe de tweets neutros. Na tentativa de contornar o problema, duas metodologias foram utilizadas, resultando em dois modelos ligeiramente distintos que foram comparados antes que uma decisão fosse tomada.

6.3.2 Modelo Fronteira

A primeira dessas metodologias, que será denotada como modelo **Fronteira**, valeu-se do uso de "certeza" nas predições do classificador.

Uma rede neural calcula a probabilidade de que uma amostra corresponda a cada uma das categorias pré-estabelecidas, e define sua predição com base na maior delas, já que a equação 4.9 garante que a soma das das probabilidades será sempre 100%. Assim, uma das formas de interpretar o grau de "certeza" de uma predição é olhar justamente para esse valor. Se uma rede neural prediz que uma sentença tem probabilidades iguais a 70% de ser positiva e 30% de ser negativa, pode-se dizer que a rede neural predisse "positivo" com 70% de certeza.

O modelo fronteira, basicamente, estabelece um limite de certeza aceitável para que uma predição seja considerada, de fato, positiva ou negativa. A suposição é de que, caso o classificador não esteja convicto entre positivo ou negativo, provavelmente a predição mais adequada seja o neutro. Dessa forma, a necessidade por tweets neutros na base de treino é aliviada e passa a ser possível compreender as três categorias.

O valor de certeza que melhor se adequou ao problema durante os testes foi 60%.

[7] http://www.sentiment140.com/

6.3.3 Modelo Balanceado

A outra metodologia aplicada foi treinar um modelo com base em um arranjo entre a base de treino original e novos tweets coletados, assumidamente, neutros**.** Essa metodologia será chamada modelo **Balanceado** no decorrer do trabalho.

O processo mais seguro de anotação seria o processo manual, no entanto, esse é um processo bastante demorado, principalmente se for levado em conta a quantidade de tweets necessária para a convergência de um modelo neural. Assim, a saída encontrada foi coletar tweets de hashtags específicas, apontadas por Kouloumpis, Wilson e Moore (2010) e assumi-los como neutros. Neste caso, a metodologia pode ser considerada bastante ruidosa, talvez, ainda mais que a dos autores, uma vez que o levantamento feito por eles aconteceu orientado a uma base de tweets que já não está mais disponível, e, neste trabalho, as hashtags foram usadas para a coleta de novos tweets.

Foi coletado um total de 10000 tweets nessa condição, que foram somados a outros 20000 tweets da base original (metade positivos e medade negativos). Priorizou-se compor uma base de treino balanceada entre as categorias, considerando que, em geral, os resultados são superiores.

6.3.4 Treinamento e escolha do modelo final

Todos modelos foram treinados por 5 épocas com *batches* de tamanho 128 e camada de atenção de tamanho 32, com otimização Adam (KINGMA; BA, 2014).

Como medidas principais de desempenho, utilizamos a acurácia e medida F1 (média harmônica entre a precisão e revocação) dos modelos.

O melhor modelo treinado foi do tipo Balanceado, com estado de tamanho 64, que apresentou 0,5850 para medida f1 e 0,5783 para acurácia. Demais detalhes sobre a performance dos modelos encontram-se no Apêndice A.

6.5. Clusterização

Quando uma coleta é encerrada, os dados são submetidos à clusterização. O algoritmo selecionado para esta etapa foi o K-means - descrito na seção 4.4 -, baseando-se como métrica na distância euclidiana dos pontos.

Os dados de entrada do algoritmo são os vetores GloVe de cada *tweet*. De forma a garantir uma boa assertividade no processo, optou-se pela utilização de 3 números diferentes de clusters. Para isso, os clusters são calculados considerando-se um valor de *k* (número de clusters) dentro de um intervalo de 2 a 15. Para cada cluster encontrado, é calculado também o *score* do método *Silhouette*.

Depois de calculados todos os clusters, os *scores* encontrados são elencados. São, então, selecionados os 3 melhores *scores* e seus respectivos clusters. Cada *tweet* recebe, então, os valores dos clusters a que pertencem de acordo com cada uma das 3 clusterizações selecionadas.

6.6. Armazenamento

Para a elaboração do subsistema de armazenamento de dados, cogitou-se a possibilidade de utilização de dois modelos distintos: relacional e NoSQL orientado a documentos.

O modelo relacional apresenta uma arquitetura e esquema de armazenamento propícios a análises complexas com boa performance para consultas de grande número de dados. Entretanto, observou-se que alguns bancos de dados NoSQL apresentam melhor performance em operações de escrita e atualização de dados (LI; MANOHARAN, 2013). Como existia a intenção de construir-se uma plataforma com atualizações em tempo real, atribuiu-se muito peso ao fator performance, o que resultou na escolha de um banco de dados NoSQL.

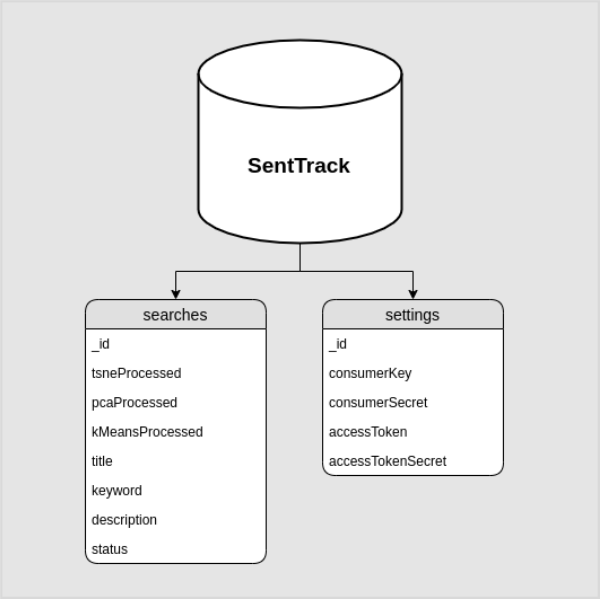
A base de dados selecionada foi o MongoDB[8], um banco de dados orientado a documentos que oferece a vantagem de se trabalhar com variadas estruturas de dados - e não apenas tabelas, como no modelo relacional - o que garante a flexibilidade necessária para armazenar os dados no formato JSON fornecidos pelo Twitter. A escolha desta base de dados

[8] https://www.mongodb.com/

foi embasada pela necessidade de um modelo robusto para transações massivas com boa performance para leitura e atualização, dado o fato de que o período entre um envio de dados e outro por parte do Twitter é muito pequeno e exige baixo tempo de resposta. Além disso, o MongoDB oferece o recurso de *Change Streams*, através do qual base de dados emite um evento para notificar alterações feitas nos dados, ideal para sistemas em tempo real.

Dentro do sistema de armazenamento, foi criada uma base de dados principal com o nome de SentTrack. Esta base de dados armazena duas coleções de documentos, sendo uma coleção para os metadados das pesquisas criadas e, outra para manter os tokens de autenticação para conexão com os serviços do Twitter. A Figura 8 a seguir apresenta a estrutura da base de dados SentTrack.

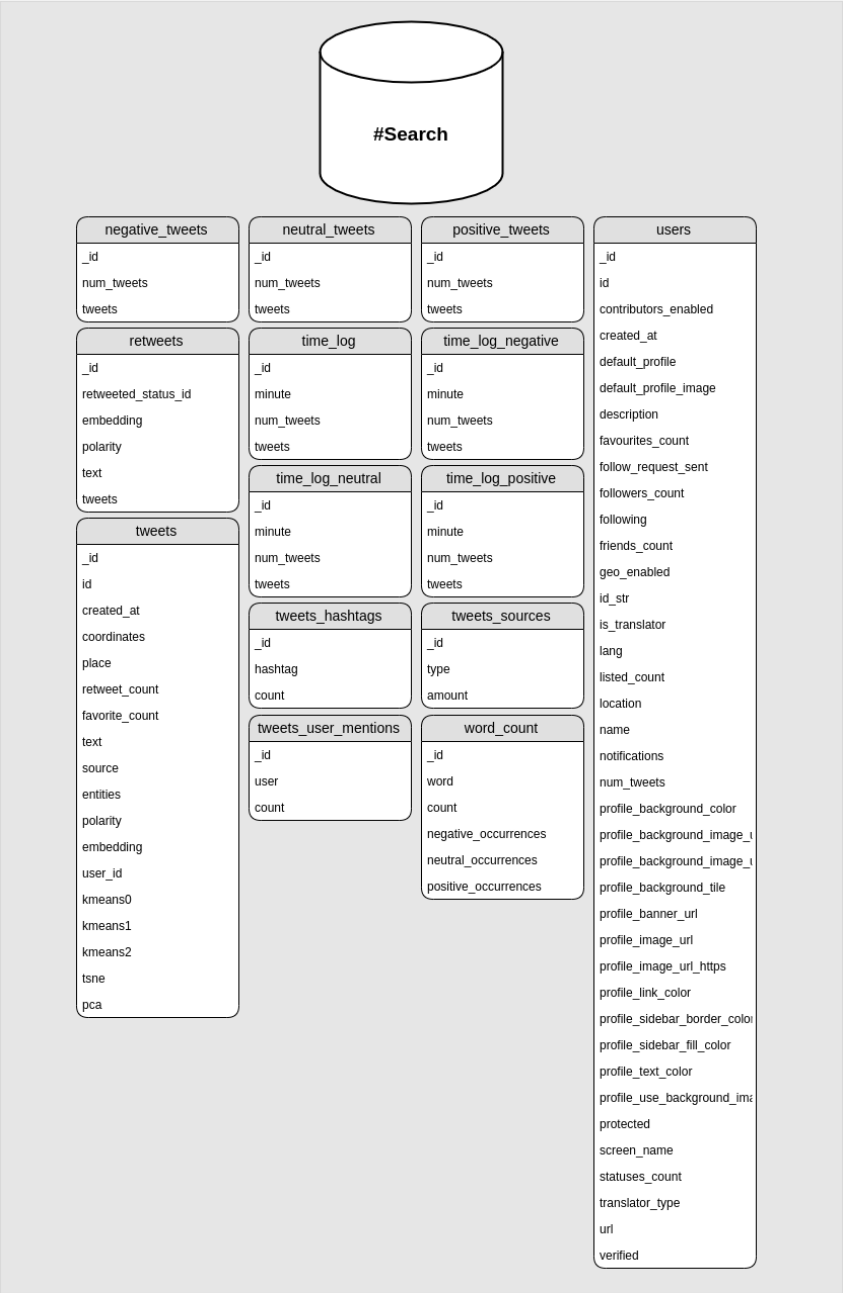
Figura 8 - Coleções de documentos da base de dados SentTrack



Fonte: elaborado pelos autores

Para cada pesquisa configurada pelo usuário, o sistema cria uma base de dados para armazenar informações pertinentes a pesquisa, de forma a proporcionar uma maior organização para os dados e melhor performance em operações de consulta. A Figura 9 a seguir apresenta a organização das coleções de documentos de cada base de dados.

Figura 9 - Coleções de documentos de uma pesquisa



Fonte: elaborado pelos autores

Os dados recebidos pelo sistema se resumem a um grande objeto aninhado contendo informações tanto a respeito do *tweet* como a respeito do usuário que o publicou. A análise dos dados seguindo esse formato padrão seria custosa e poderia ocasionar problemas de latência nas operações de consulta, tendo em vista que seriam necessárias *queries* complexas. Por exemplo, para obter-se a quantidade de *tweets* classificados como positivos, o sistema deveria aplicar um *query* sobre a coleção de dados inteira para extrair essa contabilização,

algo pouco performático. Com base nisso, optou-se por extrair, dos dados recebidos, as principais características a serem analisadas. Para cada característica foi criada uma coleção de dados, encarregada por armazenar os dados compatíveis com tal característica. No caso de exemplo de contabilização de *tweets* classificados como positivos, foi criada uma coleção apenas para a contabilização e armazenamento de tais *tweets*. Dessa forma, o resultado da contagem já fica armazenado em um documento dentro da coleção. Para o recuperar tal resultado do banco, basta uma simples *query* de consulta a um único documento da coleção e não a ela toda.

6.7. Web service

O módulo de *Backend* foi desenvolvido utilizando-se Node.js, um interpretador para a linguagem de programação JavaScript que permite o desenvolvimento de serviços assíncronos e escaláveis[9] e, consequentemente, favorece a estruturação de sistemas em tempo-real.

Foi desenvolvida uma API REST (*Representational State Transfer*), a qual ficou encarregada de receber as requisições HTTP disparadas pelo módulo de *Frontend* e retornar os dados solicitados por este, já processados e prontos para visualização.

O serviço foi construído seguindo-se os padrões da arquitetura MVC (*Model View Controller*). Portanto, para cada coleção de dados, foi estabelecido um *Model*, responsável por desempenhar as interações com a base de dados. Foram implementados *Controllers* para a interceptação das mensagens enviadas pelo cliente e execução da respectiva rotina de regra de negócio. Esta arquitetura foi implementada com o auxílio do framework Express[10], o qual apresenta recursos e uma estrutura minimalista para a configuração de API's em Node, e com o auxílio da biblioteca Mongoose[11] para modelagem de objetos referentes a cada coleção de dados. A camada de visualização (*View*) do MVC ficou a cargo do módulo de *Frontend*, descrito na seção 6.3.

A funcionalidade de atualização em tempo real também foi implementada dentro do Web Service. Através de um recurso chamado de *Change Streams* oferecido pela base de dados MongoDB com integração para a biblioteca Mongoose, quando a coleta de dados de uma pesquisa é iniciada, o serviço em Node.js começa a "observar" mudanças efetuadas no

[9] https://nodejs.org/en/about/
[10] https://expressjs.com/pt-br/
[11] https://mongoosejs.com/

armazenamento. De tal maneira, cada inserção ou atualização de dados feita pelo módulo de Coleta e Classificação, reflete em um evento de notificação que é enviado para o Web Service contendo as modificações realizadas.

O envio de dados atualizados do Web Service para o cliente foi possibilitado pela implementação do protocolo WebSocket - descrito na seção 4.5 - através da utilização da Socket.io, uma biblioteca desenvolvida para comunicação bidirecional, baseada em eventos e em tempo real entre cliente e servidor[12]. Deste modo, a base de dados notifica as alterações para o Web Service, o qual formata os dados atualizados e os repassa para o cliente em uma sequência de trocas de eventos.

6.8. Cliente

O módulo *Frontend* - ou cliente - da aplicação foi desenvolvido seguindo os conceitos de SPA (*Single Page Application*). De acordo com Jadhav et al. (2015), uma SPA apresenta componentes implementados individualmente que podem ser substituídos ou atualizados sem a necessidade de atualização da página inteira.

Seguindo esta lógica, optou-se pela seleção da biblioteca React.js, que facilita o desenvolvimento de componentes que podem ser reutilizados em múltiplas partes da aplicação e que são atualizados de forma eficiente à medida em que recebem modificações nos dados[13]. Optou-se pela escolha da biblioteca React.js justamente devido à sua capacidade de trabalhar com componentes e à sua reatividade, isto é, atualização eficiente dos componentes.

Os gráficos que compõem os *dashboards* foram desenvolvidos com a utilização da biblioteca Plotly. Optou-se pelo uso desta biblioteca pelo vasto portfólio de gráficos oferecidos por tal, além da possibilidade de integração com o React.js - também utilizado no sistema. Outro fator de vantagem da biblioteca Plotly é sua função de *refresh*, ou seja, atualização dos gráficos na medida em que recebem novos dados, obedecendo o conceito de SPA.

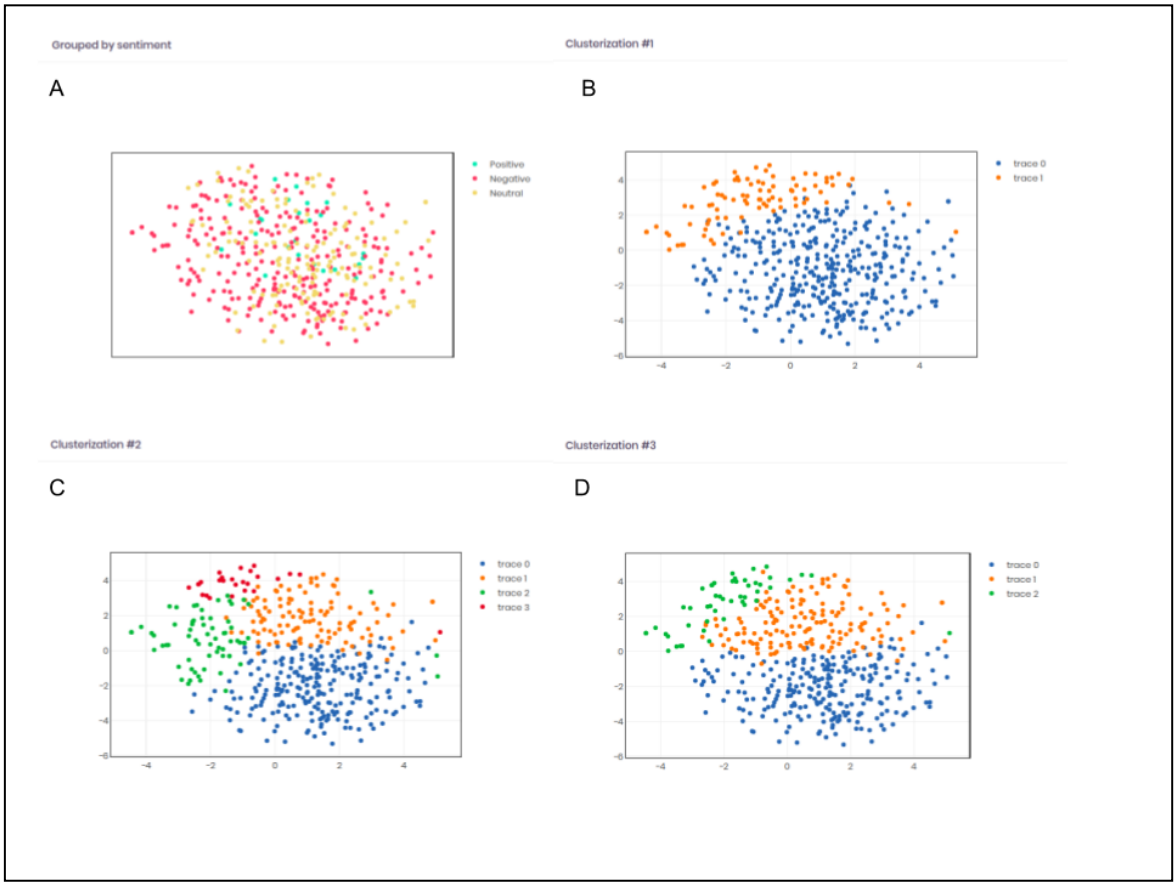
[12] https://socket.io/
[13] https://pt-br.reactjs.org/

7. SENTTRACK

Nesta seção apresentam-se os resultados obtidos através da implementação e utilização da plataforma proposta e desenvolvida, SentTrack, sobretudo no que diz respeito às técnicas de visualização da informação utilizadas para demonstração dos dados coletados da rede social Twitter.

Na Figura 10 são apresentados 4 gráficos do tipo *scatter plot* contendo dados extraídos dos *tweets* coletados. Nos gráficos, cada ponto representa um *tweet* e cada cor representa uma categoria ou cluster encontrado. As coordenadas foram encontradas submetendo-se o vetor GloVe de cada *tweet* ao método de redução de dimensionalidade t-SNE.

Figura 10 - Gráficos *scatter plot* com dimensionalidade reduzida por t-SNE



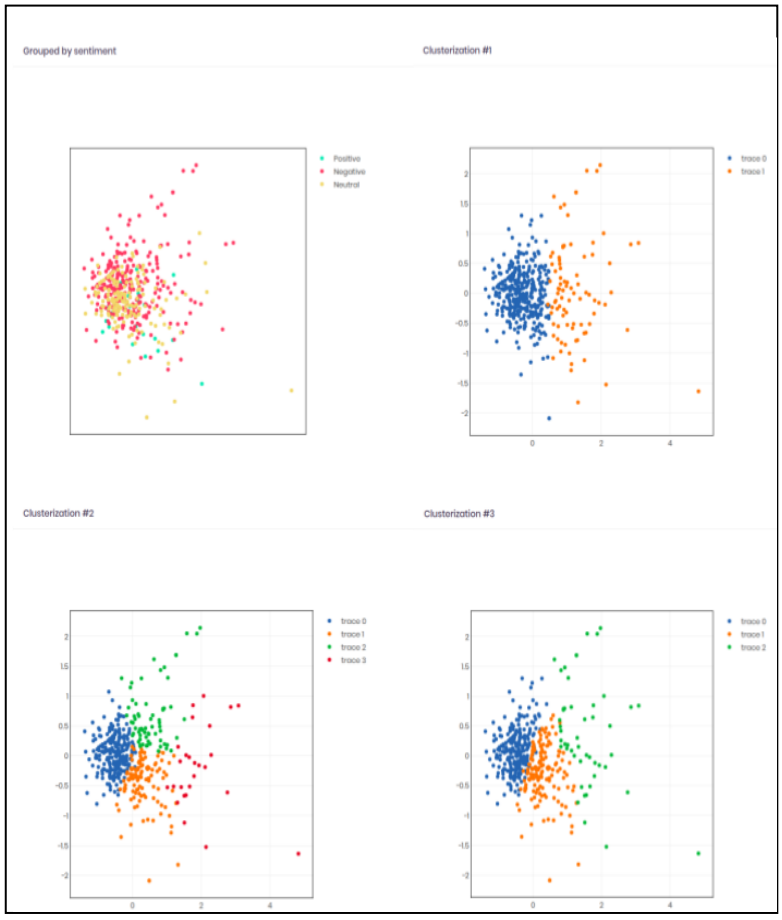
Fonte: elaborado pelos autores

Na Figura 10 (a) pode-se verificar um gráfico no qual os clusters foram determinados a partir da polaridade encontrada pelo classificador. Nas Figuras 10 (b), 10 (c) e 10 (d), os

clusters foram obtidos a partir da técnica K-means, cada qual com um valor distinto para *k* (número de clusters).

Na Figura 11, a seguir, os gráficos seguem a mesma lógica apresentados na Figura 10, entretanto, o método de redução de dimensionalidade aplicado foi o PCA.

Figura 11 - Gráficos *scatter plot* com dimensionalidade reduzida por PCA

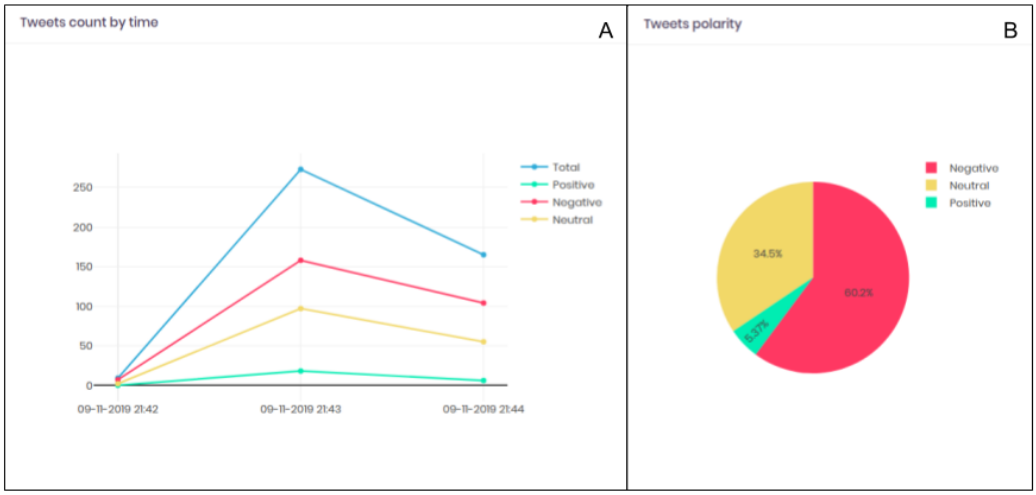


Fonte: elaborado pelos autores

O usuário pode interagir com esses gráficos, escolhendo tweets individualmente para análise e verificação de detalhes.

Foram desenvolvidos, também, gráficos quantitativos relacionados à polaridade dos *tweets* classificados. Na Figura 12 (a), encontra-se um gráfico de linha que demonstra o montante de *tweets* para cada polaridade (positivo, negativo e neutro) - além de um totalizador - ao longo do tempo. Já na Figura 12 (b), encontra-se um gráfico de setores com a contabilização total de *tweets* para cada polaridade, sendo que cada setor representa uma polaridade.

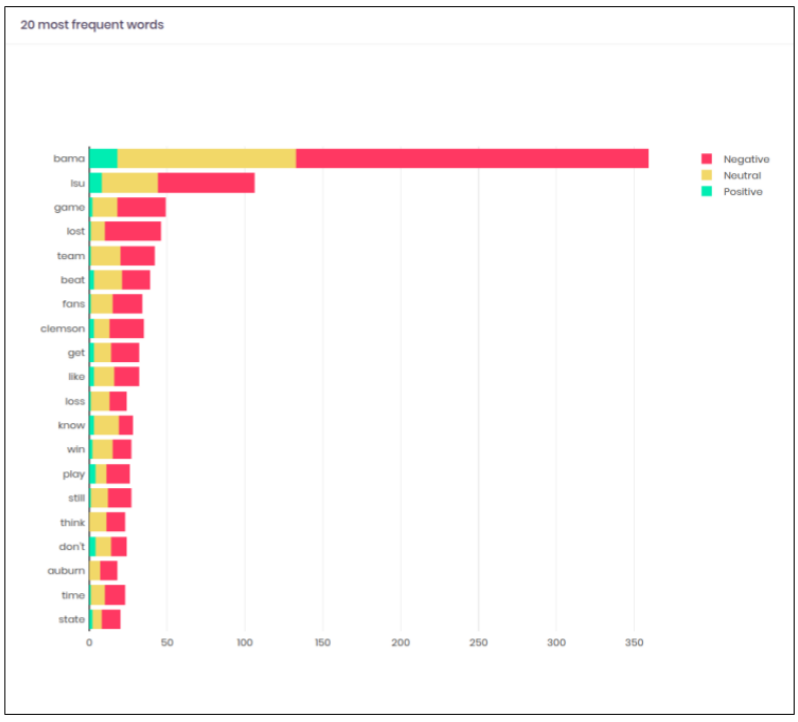
Figura 12 - Gráficos quantitativos de polaridade



Fonte: elaborado pelos autores

Na Figura 13 a seguir, apresenta-se um gráfico de barras responsável por apresentar as 20 palavras mais frequentes extraídas dos *tweets* coletados, além da contabilização de ocorrências para cada polaridade (positivo, negativo e neutro).

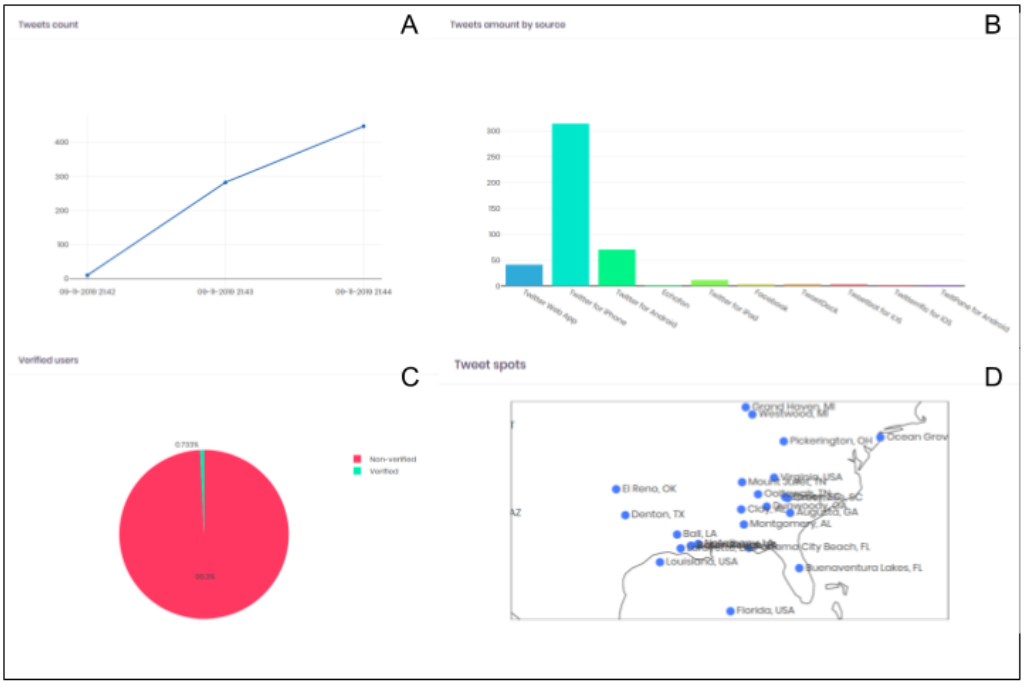
Figura 13 - Palavras mais frequentes



Fonte: elaborado pelos autores

Foram implementados, também, gráficos para visualização e análise de dados gerais extraídos dos serviços do Twitter, não relacionados à classificação de polaridades sobre os *tweets*, como apresentado na Figura 14.

Figura 14 - Gráficos com dados gerais extraídos dos serviços do Twitter



Fonte: elaborado pelos autores

A Figura 14 (a) apresenta um gráfico de linha que representa a contagem total de *tweets* coletados por minuto desde o início da coleta. A Figura 14 (b) apresenta um gráfico de barras no qual é possível verificar as diferentes fontes de origem dos *tweets* coletados. Na Figura 14 (c), apresenta-se um gráfico de setores responsável por contabilizar todos os usuários verificados e não-verificados do Twitter que realizaram publicações durante o período de coleta de dados. Já na Figura 14 (d), apresenta-se um mapa que indica as localizações de publicação dos *tweets* coletados.

Vale ressaltar que, tanto os gráficos da Figura 12 como os da Figura 14 possuem o recurso de atualização em tempo real. Dessa forma, a cada novo dado coletado é realizada a atualização da renderização dos gráficos.

Para a elaboração dos gráficos que compõem o *dashboard* desenvolvido foram considerados contextos e informações relevantes para um potencial processo de tomada de decisão, visando a garantia de conclusões mais precisas no que diz respeito à opinião pública.

8. CONCLUSÕES

O crescente volume de dados de publicações em mídias sociais aliado ao avanço de técnicas de análise de sentimento proporciona um cenário ideal para o desempenho de análises e observações interessantes, uma vez que é possível abstrair os sentimentos e opiniões vinculadas. Uma ferramenta analítica, como o SentTrack, explora tal cenário, automatizando o processo de organização e classificação de dados e, consequentemente, simplificando sua análise.

O SentTrack colabora com o estudo de padrões de comportamento de usuários de redes sociais e contribui com o surgimento de *insights*. A inteligência artificial associada a dispositivos gráficos e o estímulo da capacidade visual-cognitiva permite uma compreensão mais clara e precisa de informações relacionadas à opinião pública.

Através do SentTrack o usuário final não necessita ter conhecimentos muito específicos de ciência de dados para usufruir dos benefícios da ferramenta, que de forma simples, incorpora de maneira automatizada uma gama de conhecimentos e técnicas complexas que um leigo (por exemplo, um analista de marketing) provavelmente não terá.

Por fim, dentre as dificuldades enfrentadas, destaca-se a não existência de modelos e bases de dados públicos que abranjam completamente a problemática de análise de sentimento em contexto de redes sociais. A questão das bases, em especial, é bastante razoável, já que, uma vez que se tenha bases rotuladas de qualidade, o problema de treinar um classificador eficiente se torna muito menor.

9. CONSIDERAÇÕES FINAIS

O presente trabalho demonstra uma das múltiplas formas de aplicação de modelos de Aprendizado de Máquina - sobretudo no que diz respeito à Análise de Sentimento e Processamento de Linguagem Natural - em conjunto com técnicas de visualização de informação.

Para futuros trabalhos, deixa-se como sugestão, a elaboração de outros formatos de visualização de dados que podem agregar valor na análise, tais como tabelas pivô, ou seja, tabelas dinâmicas que permitem a estudo multidimensional das variáveis.

Sugere-se também, para futuros trabalhos, a experimentação de outros modelos classificatórios, tais como a CNN (*Convolutional Neural Network*). Uma outra possibilidade a ser explorada, seria, de acordo com as interações do usuário, refinar as capacidades do classificador, possivelmente usando técnicas tais como *Active Learning*.

A aplicação de um *layout* responsivo, tornando possível a visualização de dados de forma eficiente em diferentes tamanhos de telas, também seria uma possibilidade interessante.

REFERÊNCIAS

ABDI, H.; WILLIAMS, L. J. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, v. 2, n. 4, p. 403-459, 2010.

AGARWAL, A.; XIE, B.; VOVSHA, I.; RAMBOW, O.; PASSONNEAU, R. Sentiment analysis of twitter data. *Proceedings of the Workshop on Language in Social Media*, p. 30-38, 2011.

ALSABTI, K.; RANKA, S.; VINEET, S. An efficient k-means clustering algorithm. *Eletrical Engineering and Computer Science*. 43. 1997. Disponível em: <https://surface.syr.edu/eecs/43>. Acesso em: 13 de novembro de 2019.

BAHDANAU, D.; CHO, K; BENGIO, Y. Neural machine translation by jointly learning to align and translate. *arXiv:1409.0473*. 2014.

BRAND24. Disponível em: <https://brand24.com/>. Acesso em: 17 de junho de 2019.

CHO, K.; MERRIËNBOER, B.V.; GULCEHRE, C.; BAHDANAU, D.; BOUGARES, F.; SCHWENK, H; BENGIO, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv:1406.1078*. 2014.

CHUNG, J.; GULCEHRE, C.; CHO, K; BENGIO, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv:1412.3555*. 2014.

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

GO, A.; BHAYANI, R; HUANG, L. Twitter sentiment classification using distant supervision. *CS224N Project Report*, Stanford, v. 1, n. 2, 2009.

GRAVES, A. Supervised Sequence Labelling with Recurrent Neural Networks. 2012. Disponível em: <https://www.cs.toronto.edu/~graves/preprint.pdf>. Acesso em: 13 de novembro de 2019.

HAYKIN, S. S. Neural Networks: A Comprehensive Foundation. 2. ed. *Prentice-Hall*, 1998.

JADHAV, M.; SAWANT, B.; DESHMUKH, A. Single Page Application using AngularJS. International Journal of Computer Science and Information Technologies, v. 6, n. 3, 2015.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *arXiv:1412.6980*. 2014

KODINARIYA, T.; MAKWANA, P. Review on determining number of Cluster in K-Means Clustering. *International Journal of Advance Research in Computer Science and Management Studies*. v. 1, n. 6, p. 90-95, 2013.

KOULOUMPIS, E.; WILSON, T.; MOORE, J. Twitter sentiment analysis: The good the bad and the omg!. *Fifth International AAAI conference on weblogs and social media*, 2011.

KURNIAWATI, K.; SHANKS, G. G.; BENKMAMEDOVA, N. The Business Impact Of Social Media Analytics. *ECIS*, v. 13, p. 13, 2013.

LAN, Z.; CHEN, M.; GOODMAN, S.; GIMPEL, K.; SHARMA, P.; SORICUT, R.Albert: A lite bert for self-supervised learning of language representations. *arXiv:1909.11942*. 2019.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *nature*, v. 521, n. 7553, p. 436, 2015.

LI, X; MANOHARAN, S.; A performance comparison of SQL and NoSQL databases. 2013.

LIU, B. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, v. 5, n. 1, p. 1-167, 2012.

LOHR, S. The age of big data. *New York Times*, v. 11, 2012.

MAATEN, L. V. D.; HINTON, G. Visualizing data using t-SNE. *Journal of machine learning research*,v. 9, p. 2579-2605, 2008.

MCAFEE, A.; BRYNJOLFSSON, E.; DAVENPORT, T. H.; PATIL, D.J.; BARTON, D. Big data: the management revolution. *Harvard business review*, v. *90*, n. 10, p. 60-68, 2012.

MONIRUZZAMAN, A.; HOSSAIN, S. NoSQL Database: New Era of Databases for Big data Analytics - Classification, Characteristics and Comparison. *arXiv:1307.0191*. 2013.

NASUKAWA, T.; YI, J. Sentiment analysis: Capturing favorability using natural language processing. *In Proceedings of the 2nd international conference on Knowledge capture*, p. 70-77, ACM, 2003.

NAYAK, A.; PORIYA, A.; POOJARY, D. Type of NOSQL Databases and its Comparison with Relational Databases. *International Journal of Applied Information Systems*, v.5, n. 4, p. 16-19, 2013.

PAK, A.; PAROUBEK, P. Twitter as a corpus for sentiment analysis and opinion mining. *LREc*, v. 10, n. 2010, p. 1320-1326, 2010.

PENNINGTON, J.; SOCHER, R.; MANNING, C. GloVe: Global vectors for word representation. *In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, p. 1532-1543, 2014.

PIMENTEL, V.; NICKERSON, B. Communicating and Displaying Real-Time Data with WebSocket. *IEEE Internet Comput.*, vol. 16, n. 4, p. 45-53, 2012.

PRESS, G. A Very Short History Of Data Science . *Forbes*, 2013. Disponível em: <https://www.forbes.com/sites/gilpress/2013/05/28/a-very-short-history-of-data-science/#45027e5555cf>. Acesso em: 24 de junho de 2019.

RUTHS, D.; PFEFFER, J. Social media for large studies of behavior. Science, v. 346, n. 6213, p. 1063-1064, 2014.

SAIF, H., HE, Y.; ALANI, H. Semantic sentiment analysis of twitter. *International semantic web conference*, p. 508-524, Springer, Berlin, Heidelberg, 2012.

SENTDEX. Disponível em: <http://sentdex.com/>. Acesso em: 17 de junho de 2019.

WAGSTAFF, K.; CARDIE, C.; ROGERS, S. SCHROEDL, S. Constrained K-means Clustering with Background Knowledge. 2001. Disponível em: <https://web.cse.msu.edu/~cse802/notes/ConstrainedKmeans.pdf>. Acesso em: 13 de novembro de 2019.

WATE, C. Information visualization: perception for design. *Elsevier*, 2012.

WESSELS A.; PURVIS, M.; JACKSON, J.; RAHMAN S. Remote Data Visualization through WebSockets. *Information Technology: New Generations (ITNG)*, p. 1050-1051, 2011 Eighth International Conference on. IEEE, 2011.

WU, X.; ZHU, X.; WU, G. Q.; DING, W. Data mining with big data. *IEEE transactions on knowledge and data engineering*, v. 26, n. 1, p. 97-107, 2014.

YANG, Z.; YANG, D.; DYER, C.; HE, X.; SMOLA, A; HOVY, E. Hierarchical attention networks for document classification. *North American chapter of the association for computational linguistics: human language technologies*, p. 1480-1489, 2016.

YANG, Z.; DAI, Z.; YANG, Y.; CARDBONELL, J.. SALAKHUTDINOV, R; LE, Q. V. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *arXiv:1906.08237*. 2019.

YEUNG, K.; RUZZO, W. Principal component analysis for clustering gene expression data. *Bioinformatics*, v. 17, n. 9, p. 763-774, 2001.

ZAFARANI, R.; ABBASI, M. A.; LIU, H. Social media mining: an introduction. *Cambridge University Press*, 2014.

APÊNDICE A – Detalhes de performance dos modelos

A Tabela A-1 apresenta um comparativo entre o desempenho médio dos dois tipos de modelo, levando em conta bases de treino com 30000 tweets tanto para o Balanceado quanto para o Fronteira. Como apresentado, modelo Balanceado supera o modelo fronteira.

Tabela A-1 - Desempenho médio utilizando bases de treino com tamanhos iguais

Modelo	Precisão	Revocação	**Medida F1**	**Acurácia**
Balanceado	0,5809 ± 0,0395	0,5425 ± 0,0039	**0,5606 ± 0,0191**	**0,5783 ± 0,0037**
Fronteira	0,5228 ± 0,0221	0,5435 ± 0,0139	0,5329 ± 0,0179	0,5747 ± 0,0154

Fonte: elaborada pelos autores

Na Tabela A-2 temos o desempenho para diferentes tamanhos de base de dados, utilizando somente o modelo Fronteira. A ideia foi verificar se o desempenho acompanharia o crescimento da base de treino. Depois de 200000 tweets o modelo não apresentou melhorias, mesmo aumentado o tamanho do estado da RNN, como apontado na tabela.

Tabela A-2 - Desempenho do modelo Fronteira utilizando bases de treino com tamanhos distintos

Conjunto de treino	Tamanho do estado	Precisão	Revocação	**Medida F1**	**Acurácia**
30k	64	0,5376	0,5551	0,5462	0,5884
200k	64	0,5446	0,5586	**0,5515**	**0,5924**
400k	64	0,5122	0,5429	0,5271	0,5783
400k	128	0,5096	0,5357	0,5223	0,5702

Fonte: elaborada pelos autores