UNIVERSIDADE FEDERAL DO CEARÁ

CAMPUS QUIXADÁ

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

RAIMUNDO DE ACACIO LEONEL JUNIOR

UM FRAMEWORK PARA MINERAÇÃO DE TEXTOS DE REDES SOCIAIS

QUIXADÁ – CEARÁ

2016

RAIMUNDO DE ACACIO LEONEL JUNIOR

UM FRAMEWORK PARA MINERAÇÃO DE TEXTOS DE REDES SOCIAIS

Monografia apresentada no curso de Sistemas de Informação da Universidade Federal do Ceará, como requisito parcial à obtenção do título de bacharel em Sistemas de Informação. Área de concentração: Computação.

Orientadora: Profa. Ticiana Linhares Coelho da Silva

QUIXADÁ – CEARÁ

2016

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

L599f Leonel Junior, Raimundo de Acacio.
Um framework para mineração de textos de redes sociais. / Raimundo de Acacio Leonel Junior. – 2016.
51 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Sistemas de Informação, Quixadá, 2016.
Orientação: Profa. Dra. Ticiana Linhares Coelho da Silva.

1. Frameworks. 2. Mineração de Dados. 3. Redes Sociais. I. Título.

CDD 005

RAIMUNDO DE ACACIO LEONEL JUNIOR

UM FRAMEWORK PARA MINERAÇÃO DE TEXTOS DE REDES SOCIAIS

Monografia apresentada no curso de Sistemas de Informação da Universidade Federal do Ceará, como requisito parcial à obtenção do título de bacharel em Sistemas de Informação. Área de concentração: Computação.

Aprovada em:

BANCA EXAMINADORA

Profa. Ticiana Linhares Coelho da Silva (Orientadora)
Campus Quixadá
Universidade Federal do Ceará – UFC

Davi Romero de Vasconcelos
Campus Quixadá
Universidade Federal do Ceará - UFC

Leonardo Sampaio Rocha
Universidade Estadual do Ceará - UECE

Regis Pires Magalhães
Campus Quixadá
Universidade Federal do Ceará - UFC

A Deus.

Aos meus pais, minha familia, minha namorada e meus amigos.

AGRADECIMENTOS

Agradeço primeiramente a Deus por ter me dado o dom da vida, saúde e força para superar todas as todas minhas dificuldades.

Agradeço a toda minha família, em especial aos meus pais Galego e Diunizia, e aos meus irmãos Cristina e Marcos, que com muito carinho e apoio, não mediram esforços para que eu chegasse até essa etapa de minha vida.

A minha namorada Jussara Josué, pessoa com quem amo partilhar a vida. Obrigado pelo carinho, paciência e por sua capacidade de me trazer paz na correria de cada semestre.

Agradeço muito a minha orientadora, Ticiana Linhares Coelho da Silva, que sempre acreditou na minha capacidade e me deu o estímulo e apoio necessários para continuar.

Agradeço a todos os meus amigos de Dep. Irapuan Pinheiro pelo companheirismo. Em especial, Das Chagas, Jandson, Wendel, Jordana, Geslany, Pituca, Gewerton e Anderson.

Aos meus amigos do CSF João, Rapha, JLo, Lorrayne, Victor, Vini, Giovana, Bruno. E aos meus irmãos do CSF, Edson, Gustavo, Carlos, Tárek, Bruno Lima, Tiago, André, Thalles pela amizade e por todos os momentos em que vivemos juntos.

Agradeço aos grandes amigos que fiz nessa Universidade, com os quais vivi muitos momentos importantes e inesquecíveis da minha vida. Em especial, Sávio, Ricardo, Rafael, Wanrly, Araujo, Anderson, Yago, Alex, Alexsandro, Erick, Holanda, Adeilson, Kayro, Caio, William, Claudio, Cintia, Tercio, Klyssia, Bruno, Anderson, Danrley, Guilherme, Macilio, Kerley e Cainã.

Agradeço em especial aos amigos Alan Martins, Daniel Filho, Araujo filho, Tercio Jorge e Salomão da Silva por terem me dado muito suporte e incentivo para a realização desse trabalho.

Agradeço a todo o grupo PET-SI pelas experiências compartilhadas e pelo conhecimento adquiri fazendo parte desse incrível grupo. Ao professor tutor Davi Romero por seus importantes conselhos e ensinamentos durante a maior parte da minha graduação. Ao co-tutor Lucas Ismaily e aos meus queridos amigos petianos. Em especial, Mardson, Leonara, Wolney, Paulo Felipe, Lina, Wellington, Marcelo, Alysson, Matheus, Flavio, Lucas, Rafael e Allan.

Agradeço a todos os professores do campus UFC Quixadá pelo excelente trabalho que exercem, nos dando ensinamentos impagáveis. Em especial aos professores Davi Romero, Ticiana Linhares, Régis Pires, Paulo de Tarso, Carla Ilane, Tânia, Sammy, Ricardo, Wladmir, Ênio, Jefferson Carvalho e Jefferson Kenedy.

"A verdadeira motivação vem de realização, desenvolvimento pessoal, satisfação no trabalho e reconhecimento."

(Frederick Herzberg)

RESUMO

Grandes avanços em tecnologias de armazenamento e a popularização da internet têm possibilitado que organizações gerem e armazenem grandes quantidades de dados. Os dados produzidos por redes sociais crescem a cada dia. Os usuários dessas redes sociais compartilham informações, sentimentos e opiniões sobre os mais diversos assuntos que acontecem em todo mundo. Essa produção crescente de dados gera a oportunidade e a necessidade de realizar análises para extrair conhecimentos úteis desses dados. Assim sendo, muitos trabalhos já foram realizados com o propósito de extrair conhecimento de dados em redes sociais. A maioria desses trabalhos compartilham de uma metodologia muito semelhante. As fases de coleta de dados, pré-processamento, análise e visualização, são etapas comuns em trabalhos de Mineração de Texto em redes sociais. Contudo, tais trabalhos necessitaram realizar implementações próprias para a realização de cada uma das fases. Consequentemente, tais trabalhos necessitaram despender muito tempo nas fases intermediárias do processo, restando menos tempo para a real análise dos dados. Motivado por esse problema, esse trabalho teve o objetivo de criar um framework capaz de auxiliar e agilizar estudos com ênfase em Mineração de textos em redes sociais. O framework foi criado englobando as fases de coleta de dados, pré-processamento, análise e visualização através de um sistema web. Neste trabalho também foi realizado um estudo de caso com ênfase em dados coletados no Twitter referentes ao *impeachment* da ex-presidente Dilma Rousseff. Tal estudo foi realizado com a intensão de validar o framework criado e de analisar a repercussão do *impeachment* na rede social Twitter.

Palavras-chave: Frameworks. Mineração de dados. Redes sociais.

ABSTRACT

Big advances in storage technology and the popularization of the internet have enabled organizations to generate and store big amounts of data. The data produced by social networks grows each day. Its users share information, feelings and opinions about a wide range of matters that happen over the world. This increasing production of data gives the opportunity and necessity of performing analyses to extract useful knowledge from it. Therefore, many researches were already performed for the purposes of extracting knowledge from social network data. Most of these researches have shared a really similar methodology. The stages of data collect, preprocessing, analyses and visualization are common stages in text mining and social network researches. However, these researches need to make their own implementations to perform each one of its stages. Consequently, these researches need to spend a lot of time in the intermediate stages of the process, leaving less time for the real data analyses. Motivated by this problem, this research had as the objective the creation of a framework capable of helping and accelerating studies with emphasis on text mining in social network. The framework was created including the stages of data collect, preprocessing, analyses and visualization through a web system. For this research it was also performed a case study with emphasis on data collected from twitter related to the impeachment of the former president of Brazil, Dilma Rousseff. This research was performed with the intention to validate the created framework and analyze the repercussion of the impeachment in the social network Twitter.

Keywords: Frameworks. Data mining. Social networks.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

PLN	Processamento de Linguagem Natural
NLTK	Natural Language Toolkit
JSON	JavaScript Object Notation

SUMÁRIO

1 INTRODUÇÃO

Grandes avanços em tecnologias de armazenamento e a popularização da internet e dos dispositivos móveis têm possibilitado que organizações gerem e armazenem grandes quantidades de dados. A cada dia, são criados cerca de 2,5 quintilhões de bytes de dados e a quantidade de dados existente no mundo dobra a cada dois anos (WU et al., 2014).

Várias são as fontes desse grande volume de dados gerado nos dias atuais. Leonel Junior et al. (2014) destaca algumas destas fontes, que produzem juntas, dados que alcançam a escala de petabytes diários. São elas: redes sociais, sistemas corporativos, serviços e sistemas web, transações financeiras, e-commerce entre outros. Dentre tantas fontes de dados existentes, esse trabalho possui ênfase em dados provenientes de redes sociais.

Aggarwal e Zhai (2012) destacam que redes sociais são uma fonte muito comum de texto na web, pois elas permitem que atores humanos se expressem e se comuniquem de forma rápida e livre sobre os mais diversos tipos de assuntos. Tais características das redes sociais propiciam uma grande geração de dados. Segundo Simos (2015), o Facebook[1] possui mais de 1.4 milhões de usuários ativos, e gera a maior quantidade de dados dentre as redes sociais. São cerca de 250 milhões de curtidas em posts no facebook por hora. Simos (2015) também destaca que no twitter, segunda maior rede social do mundo, são gerados 347,222 mil *tweets* por segundo, totalizando cerca de 21 milhões de *tweets* por hora.

Contudo, apenas gerar grandes quantidades de dados não é o bastante. Tais dados em seu formato bruto e/ou vistos de forma individual são de certa forma desperdiçados, pois existe um grande potencial de descoberta de conhecimentos. Dessa forma, é necessário realizar análises para extrair informações importantes que podem ser transformadas em conhecimento útil, auxiliar no processo de tomada de decisão, entre outros usos. Essas análises podem ser feitas por meio de técnicas de mineração de dados.

O processo de minerar dados é composto por um conjunto de técnicas baseadas em modelos capazes de encontrar padrões, sumarizar dados, extrair novos conhecimentos ou realizar predições com o objetivo de descobrir informações com base em grandes volumes de dados (SILVA et al., 2013).

É evidente a relevância de estudos com ênfase em análise de dados de redes sociais. Atualmente, já existem vários trabalhos realizados nessa área, como o trabalho de Rodrigues (2016) que aplicou técnicas de evolução de *clusters* para acompanhar a dinamicidade dos assuntos

[1] http://facebook.com

em redes sociais, ou ainda como o de Leite (2016) que usou um algoritmo de classificação para categorizar *tweets* em notícias referentes a religião, esporte, política, entre outros. E por fim, os trabalhos de Viana (2014) e Filho (2014) fizeram uma análise de sentimentos em mensagens da rede social Twitter, classificando os *tweets* em positivo, negativo, ambíguo e neutro.

Devido a esses trabalhos acima apresentados seguirem o mesmo processo metodológico e a apresentarem bons resultados, este trabalho identificou a oportunidade de propor um *framework* capaz de expressar tal processo de análise em dados de rede sociais. Dessa forma, este trabalho é um guia para novos trabalhos nessa mesma temática e evita "retrabalhos" de implementação. Todos os trabalhos citados no parágrafo anterior compartilham de um processo metodológico extremamente semelhante. Eles realizaram uma coleta de dados utilizando como fonte uma rede social (Twitter), fizeram o pré-processamento de dados usando PLN (Processamento de Linguagem Natural), aplicaram uma técnica de mineração de dados e por fim, analisaram os resultados.

A existência de retrabalho de implementação por si só já gera a necessidade de criar uma solução capaz de minimizar esse problema para trabalhos futuros. Além disso, é válido salientar que o retrabalho possui um impacto muito grande no resultado final de cada um desses trabalhos, pois os autores gastam muito tempo e esforço em fases intermediárias do processo de mineração de texto, sobrando assim menos tempo para a fase de análise e validação dos resultados.

Com esse problema em ênfase, este trabalho propõe uma solução em forma de *framework* genérico capaz de minimizar o problema de retrabalho em projetos de mineração de textos em dados de redes sociais. Tal *framework* está focado nos passos de coleta dos dados em redes sociais, pré-processamento dos dados, análise e geração de visualização dos textos minerados. Acrescenta-se também, como um passo adicional, a utilização de um sistema web visando a disponibilização e o compartilhamento dos resultados de forma mais ampla.

O *framework* foi criado utilizando um conjunto de aplicações já existentes, como também algumas aplicações criadas pelos autores deste trabalho. É válido destacar que as aplicações existentes que foram utilizadas pelo *framework* foram revisadas e documentadas, visando facilitar a utilização por diferentes pessoas em trabalhos futuros.

Além disso, foi criado em paralelo um estudo de caso com dados do Twitter sobre o *impeachment* da ex-presidente Dilma Rousseff, com a finalidade de validar o *framework* proposto. Para cada etapa do processo do estudo de caso, foi criada uma parte do *framework* capaz de

atender as necessidades para a finalização da mesma.

Um ponto de extrema importância é o fato de o *framework* possuir um sistema web capaz de disponibilizar de forma ampla os resultados obtidos pelos trabalhos. É necessário destacar a importância de expôr tais resultados a sociedade e não apenas para a comunidade acadêmica.

O trabalho está estruturado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados. Na Seção 3 são expostos os conceitos básicos que são necessários para o entendimento do trabalho. Na Seção 4 é definido o objetivo geral do trabalho, bem como os objetivos específicos. A Seção 5 apresenta a metodologia que será utilizada. Na Seção 6 são detalhados os resultados deste trabalho. Por fim, na Seção 7 são feitas as considerações finais.

2 TRABALHOS RELACIONADOS

A seguir, serão apresentados alguns trabalhos que se relacionam ao contexto deste trabalho do conhecimento dos autores. Neste sentido, serão expostos brevemente tais estudos, bem como suas semelhanças e suas diferenças em relação ao trabalho aqui proposto.

Em Rodrigues (2016), foi realizado o processo de mineração de textos em dados do Twitter com ênfase na dinamicidade dos assuntos abordados em redes sociais em um determinado período de tempo. Primeiramente, Rodrigues (2016) coletou *tweets* sobre os assuntos relacionados com os protestos que ocorreram no Brasil em 2014, como a Parada Gay. Em seguida, realizou a etapa de pré-processamento dos *tweets* e executou o algoritmo de clusterização de dados DBSCAN utilizando-se das medidas de similaridade de Jaccard e Fading Lee, Lakshmanan e Milios (2014). Além disso, foram aplicadas técnicas de evolução de *cluster* a fim de identificar a evolução de conjuntos de assuntos repercutidos no Twitter em um determinado período de tempo. Rodrigues (2016) apresentou os resultados do seu trabalho utilizando nuvens de palavras.

A proposta deste trabalho é bastante semelhante à apresentada em Rodrigues (2016), levando em consideração que ambas aplicaram mineração de textos em dados coletados do Twitter. Além disso, é válido ressaltar que os dois trabalhos possuem uma metodologia extremamente semelhante, pois o trabalho aqui proposto também realizou para o estudo de caso uma coleta de texto no Twitter, aplicou o pré-processamento dos dados utilizando processamento de linguagem natural (PLN) com o auxilio da biblioteca NLTK[1], utilizou o algoritmo DBSCAN para analisar os dados e apresentou os resultados com o auxilio de nuvens de palavras.

Entretanto, este trabalho se diferencia ao de Rodrigues (2016), pois os dados coletados distinguem-se em tema e em época no processo de coleta dos tweets. Adicionalmente, o trabalho aqui proposto criou um framework capaz de ajudar futuros trabalhos com objetivo semelhante ao de analisar textos de redes sociais. Contudo, este trabalho não utilizou de técnicas de evolução de *cluster* como em Rodrigues (2016). Outro ponto a se destacar é o fato da disponibilização e compartilhamento dos resultados na web, visto que apenas neste trabalho foi criado um sistema web para tal finalidade.

Filho (2014) e Viana (2014) também realizaram mineração de texto em seus trabalhos. Ambos realizaram o processo de coleta de informação utilizando a API do Twitter[2], logo

[1] nltk.org

[2] https://dev.twitter.com/rest/public

após aplicaram processamento de linguagem natural, em seguida a análise e visualização dos resultados usando nuvens de palavras. Ambos os trabalhos de Filho (2014) e Viana (2014) possuem enfoque em análise de sentimentos. Os *tweets* analisados foram categorizados como: positivo, negativo, neutro e ambíguo. Porém, Filho (2014) realizou sua pesquisa com dados referentes a copa do mundo de 2014, enquanto que Viana (2014) aplicou seu trabalho em *tweets* sobre as eleições presidenciais de 2014.

As propostas de Filho (2014) e Viana (2014) assemelham-se a este trabalho pois, mais uma vez, compartilham de processos de desenvolvimento similares: Coleta de dados do Twitter, pré-processamento, e apresentação dos resultados utilizando nuvens de palavras.

No entanto, Filho (2014) e Viana (2014) utilizaram de uma técnica de mineração de dados chamada classificação, o que se difere do estudo de caso deste trabalho. No estudo de caso, foi utilizada a técnica de mineração de dados chamada de clusterização. Os dados coletados neste trabalho são sobre o *impeachment* da ex presidente Dilma Rouseff no ano de 2016. No entanto, Filho (2014) e Viana (2014) trabalharam com dados sobre a copa de 2014 e as eleições presidenciais de 2014, respectivamente.

O estudo de Leite (2016) consiste na construção de um modelo capaz de classificar, em categorias, os dados coletados da rede social Twitter. Os *tweets* foram classificados em: economia, esporte, religião, política e outros. O trabalho de Leite (2016) não difere dos demais trabalhos aqui citados quando se trata do processo de mineração de texto. Leite (2016) também realizou a coleta de dados utilizando a API do Twitter, aplicou o processo de mineração de dados de forma semelhante aos demais trabalhos, utilizando um algoritmo de classificação para analisar os dados, e por fim, fez uso de nuvem de palavras para ajudar na apresentação dos resultados.

Fica claro mais uma vez, a extrema semelhança entre os processos dos trabalhos aqui apresentados, incluindo a necessidade de implementação de *scripts* para a coleta e pré-processamento de dados.

Vale a pena ressaltar uma grande diferença entre o trabalho aqui proposto com o apresentado em Leite (2016). Os resultados obtidos pelo estudo de Leite (2016), bem como os resultados de Rodrigues (2016), Filho (2014) e Viana (2014), não foram disponibilizados e compartilhados de forma ampla para a sociedade. Um dos diferenciais deste trabalho é exatamente a criação de um sistema web capaz de disseminar o conhecimento produzido por esse tipo de trabalho.

Realizando um estudo nos procedimentos metodológicos dos trabalhos de Rodrigues

(2016), Filho (2014), Viana (2014) e Leite (2016), foi possível identificar, para cada passo do processo de execução do trabalho, se o autor do trabalho implementou seu próprio mecanismo para executar o passo, ou se apenas utilizou de implementações de terceiros. As informações dessa análise estão destacadas de forma simples na tabela 1. Utilizamos a palavra "pelo autor" se o autor do trabalho teve de fazer sua própria implementação. Já a palavra "ferramenta" foi usada quando o autor apenas utilizou de uma ferramenta já existente e utilizamos "NÃO" quando o trabalho não realizou tal passo.

Quadro 1 – Análise da metodologia dos trabalhos relacionados

Trabalho	Coleta	Pré-processamento	Análise	Nuvem de palavras	Publicação web
Leite (2016)	pelo autor	pelo autor	pelo autor	ferramenta	NÃO
Rodrigues (2016)	pelo autor	pelo autor	pelo autor	ferramenta	NÃO
Filho (2014)	pelo autor	pelo autor	ferramenta	ferramenta	NÃO
Viana (2014)	pelo autor	pelo autor	ferramenta	ferramenta	NÃO
Este Trabalho	ferramenta	ferramenta	pelo autor	ferramenta	pelo autor

Fonte: Elaborado pelo autor

3 FUNDAMENTAÇÃO TEÓRICA

Nessa seção são expostos os conceitos que fundamentam este trabalho. Primeiramente, é explicado o conceito de clusterização. Após, destaca-se o algoritmo de clusterização que foi utilizado por esse trabalho, que é o DBSCAN. Além disso, é apresentado o conceito de medida de similaridade que foi utilizado conjuntamente com o algoritmo de clusterização. Por último, apresentamos brevemente um tópico sobre a rede social Twitter.

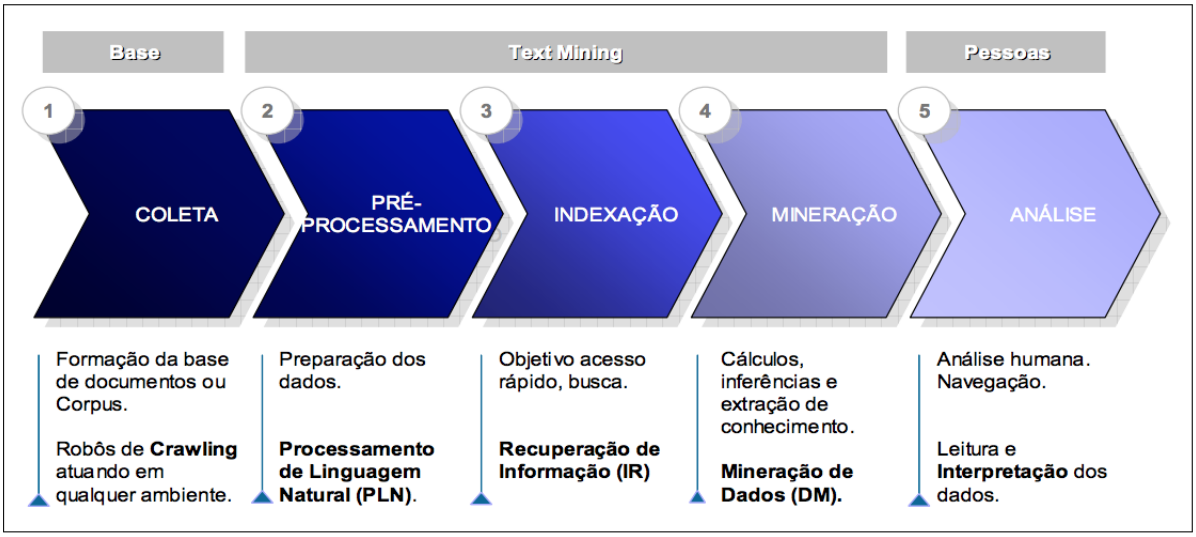
3.1 Mineração de texto

O processo de minerar dados é composto por um conjunto de técnicas baseadas em modelos capazes de encontrar padrões, sumarizar dados, extrair novos conhecimentos ou realizar predições com o objetivo de descobrir informações com base em grandes volumes de dados (SILVA et al., 2013).

Mineração de textos, ou em em inglês *TextMining*, pode ser considerada como uma extensão ou sub-área da Mineração de Dados (RODRIGUES, 2016). Segundo Tan et al. (1999), refere-se a um processo de extração de conhecimento, padrões interessantes e não triviais de documentos de texto.

A Figura 1 apresenta segundo Aranha, Vellasco e Passos (2007), um modelo de processo de Mineração de Textos, do início ao fim, muito comum entre vários trabalhos da literatura. Tal processo é semelhante com os apresentados na seção anterior.

Figura 1 – Processo de Mineração de Texto



Fonte: Aranha, Vellasco e Passos (2007)

Mathiak e Eckstein (2004), detalha cada uma dessas etapas. A etapa de coleta é basicamente onde são coletados os dados que serão analisados. Esses dados podem ser coletados em sites, sistemas corporativos, redes sociais entre outros. O pré-processamento tem por finalidade melhorar a qualidade dos dados já disponíveis e organizá-los. A indexação é a fase onde são extraídos os conceitos dos documentos por meio da análise de seu conteúdo. Na fase de mineração é onde são aplicadas técnicas para a extração do conhecimento, tal como clusterização. Por fim, é realizada a análise e interpretação dos dados pela pessoa responsável, procurando padrões e/ou analisando de forma manual os resultados da etapa de mineração.

3.2 Clusterização

Clusterização é uma técnica de mineração de dados que consiste na divisão dos dados em grupos de objetos com características semelhantes, ou que estejam próximos uns dos outros. A ideia é que objetos que estão no mesmo grupo sejam mais similares em comparação a objetos de grupos diferentes. Entre os principais algoritmos de clusterização estão: K-means (WU, 2008) e o DBSCAN (ESTER et al., 1996).

A técnica de clusterização pode ser usada em diversos contextos, como, por exemplo: a descoberta de perfis de clientes de uma empresa. Cada perfil pode ser um *cluster*. Os perfis podem ser formados por clientes de uma mesma faixa etária, e que costumam comprar o mesmo tipo de produto, por exemplo. A partir desse conhecimento uma empresa pode realizar campanhas publicitárias e/ou promoções voltadas para um grupo específico de clientes.

3.2.1 DBSCAN

O DBSCAN é um algoritmo baseado em densidade, não sendo necessário que o usuário defina a priori quantos clusters irá ter como resultado, o próprio algoritmo que define os clusters agrupando os dados que estiverem mais próximos para formarem um cluster. Cada região de densidade maior ou igual a um *threshold* dado de entrada se tornará um *cluster*(ESTER et al., 1996).

Para explicar o funcionamento do DBSCAN é necessário conhecer algumas definições:

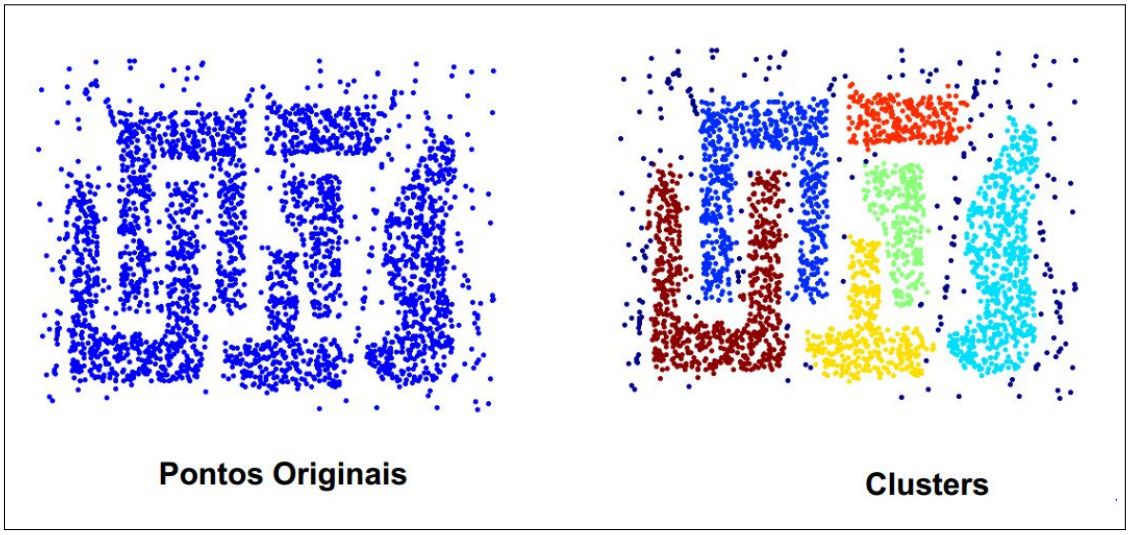
- **eps**: A distância máxima entre dois pontos para serem considerados vizinhos.
- **minPts**: O número minimo de vizinhos que um ponto deve ter para ser

considerado um corePoint.

- **corePoint**: É um ponto onde seu número de vizinhos é maior ou igual a minPts.
- **borderPoint**: Possui o número de vizinhos menor que minPts, mas é vizinho de um corePoint.
- **noisePoint**: Possui o número de vizinhos menor que minPts, e não possui um corePoint em sua vizinhança.

O Algoritmo inicia escolhendo um ponto aleatório X, analisa sua vizinhança e se possuir uma vizinhança maior que *minPts*, um novo *cluster* C é criado. Depois, o algoritmo percorre todos os pontos vizinhos de X, se algum desses pontos tiver uma vizinhança de pelo menos *minPts* esses pontos da vizinhança são inseridos no *cluster* C, e esse processo se repete até que não tenha mais pontos à adicionar neste *cluster* C. Depois de finalizado um *cluster*, o algoritmo pega um ponto que ainda não foi visitado e recomeça o processo. A Figura 2 mostra um exemplo de como o DBSCAN divide os dados em *clusters*.

Figura 2 – Divisão dos dados em *clusters*



Fonte: Tan et al. (2006)

3.3 Medida de Similaridade

KASZNAR e GONÇALVES (2009) falam que o processo de clusterização requer medidas de “proximidade” ou “similaridade”. Medidas de similaridade são métricas capazes de analisar um conjunto de características com o objetivo de comparar objetos (RODRIGUES, 2016).

A técnica utilizada neste trabalho é a similaridade de Jaccard, pois é uma medida simples e que pode ser aplicada no contexto desse trabalho. Tal medida já foi utilizada em trabalhos similares como em Rodrigues (2016) e Yin et al. (2012). Segundo Russell (2013), a similaridade de Jaccard expressa a similaridade entre dois conjuntos como a interseção dos conjuntos dividida pela união deles. Matematicamente, a similaridade de Jaccard é escrita como:

$$J(p^L, b^L) = \frac{\left|p^L \bigcap b^L\right|}{\left|p^L \bigcup b^L\right|}$$

A similaridade de Jaccard resulta em um valor entre 0 e 1. Quanto mais próximo a 1 o resultado de uma comparação, mais semelhantes são os conjuntos. Sendo assim, considerando um conjunto A com 6 elementos, e um conjunto B com 8 elementos, e considerando que eles apresentam 4 elementos em comum, a similaridade de Jaccard(A,B) = 4/10.

3.4 Twitter

Russell (2013) apresenta o Twitter como um serviço de microblog que permite que pessoas compartilhem ideias e pensamentos de forma rápida e gratuita utilizando-se de apenas 140 caracteres. Em Recuero e Zago (2016) o Twitter é apresentado como um site de rede social, pois permite aos seus usuários a construção de perfis públicos em um espaço da web, bem como prover uma estrutura para a conexão de tais perfis. Em 2013, o Twitter possuía mais de 500 milhões de usuários cadastrados e mais de 100 milhões de usuários ativos em todo o mundo Russell (2013).

O uso frequente desse tipo de serviço web gera uma quantidade enorme de dados, não apenas o twitter, mas também outras redes sociais tais como Facebook, Linkedin, Google+, se tornam uma fonte útil de dados para analisar e compreender as tendências populares.

No contexto desse trabalho, o Twitter será utilizado como base de dados para a aplicação de técnicas de mineração de dados para extrair conhecimento de um assunto específico, no caso, o *impeachment* da ex presidente Dilma Rousseff. O Twitter foi escolhido por ser uma Rede Social que possui uma API para coleta de informações de fácil utilização e por ser uma Rede Social que não necessita da aprovação dos usuários para a coleta dos conteúdos postados nessa plataforma.

3.5 Framework

Para Mattsson e Bosch (2000), um framework é uma arquitetura desenvolvida com o objetivo de atingir a máxima reutilização, representada como um conjunto de classes abstratas e concretas, com grande potencial de especialização. Já segundo Johnson e Russo (1991), um framework é um conjunto de objetos que colaboram com o objetivo de atender a um objetivo de responsabilidades para uma aplicação específica ou um domínio de aplicação.

No contexto deste trabalho, foi utilizado o termo framework para formalizar o conjunto de ferramentas que foram agrupadas com o objetivo de auxiliar o processo de mineração de textos em Redes Sociais. Cada uma das ferramentas que formam o framework proposto por esse trabalho facilita uma das etapas do processo de mineração de textos em Redes Sociais.

4 OBJETIVOS

4.1 Objetivo Geral

Reconhecendo a importância de entender e acompanhar a dinâmica dos assuntos em redes sociais, o trabalho aqui apresentado teve como objetivo a criação de um *framework* para facilitar o processo de mineração de textos em redes sociais, bem como a realização de um estudo de caso utilizando o *framework* criado.

4.2 Objetivos Específicos

- Realizar um levantamento dos trabalhos que vêm sendo desenvolvidos para mineração de dados de redes sociais;
- Verificar as equivalências e diferenças nos trabalhos levantados com respeito aos procedimentos metodológicos;
- Criar um framework para facilitar a replicação do processo de análise de texto no estudo de caso realizado neste trabalho e em futuros trabalhos;
- Realizar a coleta de uma base de dados com assuntos pertinentes tratados no Twitter para o estudo de caso;
- Implementar um algoritmo de clusterização com a utilização da medida de similaridade de *Jaccard* para o estudo de caso;
- Utilizar de Processamento de Linguagem Natural para organizar, selecionar e tratar os dados da base como fase de Pré-processamento para o estudo de caso;
- Executar o processo de clusterização dos dados utilizando o algoritmo de clusterização implementado para o estudo de caso;
- Analisar os resultados obtidos do estudo de caso.

5 PROCEDIMENTOS METODOLÓGICOS

A metodologia deste trabalho se divide em:

- Criação do framework;
- Coleta dos *tweets*;
- Pré-processamento da base de dados;
- Clusterização dos dados com DBSCAN;
- Análise dos resultados.

As subseções a seguir mostram de forma detalhada e clara como cada um dos passos necessários para a execução desse trabalho foi realizado. É explica a criação do framework, bem como a execução do estudo de caso.

5.1 Criação do framework

A primeira etapa deste trabalho foi a identificação das metodologias dos trabalhos realizados sobre mineração de textos em redes sociais no Campus em Quixadá da Universidade Federal do Ceará, bem como, as semelhanças que esses trabalhos possuem em comum.

Em seguida, foi reunido um conjunto de ferramentas capazes de solucionar cada uma das etapas do processo de mineração de textos em redes sociais. Foram utilizadas ferramentas criadas por Filho (2014) para as fases de coleta de dados do Twitter e para a etapa de pré-processamento. Para a fase de análise, foi implementada uma versão do algoritmo de clusterização DBSCAN com 3 medidas de similaridade, Euclidiana (BARROSO; ARTES, 2003), *Fading* (LEE; LAKSHMANAN; MILIOS, 2014) e *Jaccard* que foi utilizado no estudo de caso e melhor detalhada na fundamentação teórica deste trabalho. Para a geração das nuvens de palavras foi selecionada a ferramenta Tagul[1], por ser uma ferramenta de simples utilização e por permitir uma grande customização da nuvem, como formato da nuvem, fonte do texto, cores entre outras opções. Por fim, foi criado e adicionado ao framework um sistema web com a finalidade de disponibilizar o conhecimento gerado por projetos de mineração de texto em redes sociais para a comunidade.

Além disso, pensando na disponibilização do framework para futuros interessados, foi criado um repositório público no Github onde todas essas aplicações, *script* para coleta dos dados no Twitter, *script* para pré-processamento, implementação do DBSCAN e o sistema web,

[1] https://tagul.com/

foram disponibilizadas para a comunidade acadêmica[2].

5.2 Coleta dos dados

A segunda etapa deste trabalho foi a criação da base de dados necessária para o estudo. Portanto, foi realizada uma coleta de *tweets* para a criação dessa base no período entre agosto e setembro de 2016. Tal coleta foi realizada utilizando a API de streaming do Twitter através de uma aplicação feita em JAVA por Filho (2014), que coleta em tempo real os *tweets* relacionados e cria um arquivo de saída em formato JSON[3].

A coleta foi feita fundamentalmente em *tweets* sobre o assunto do processo de *Impeachment* presidencial em agosto de 2016. O processo iniciou-se com a aceitação, em 2 de dezembro de 2015, pelo ex-Presidente da Câmara dos Deputados, Eduardo Cunha. A ex-presidente Dilma Rousseff foi acusada de crime de responsabilidade. O julgamento iniciou-se no dia 25 de agosto de 2016 e terminou no dia 31 de agosto de 2016 com a aprovação do pedido de *impeachment* da ex-Presidente Dilma Rousseff.

Para tal, foram coletados *tweets* que continham *hashtags* relacionadas ao assunto, tais como: impeachment, PelaDemocracia, ImpeachmentDay, golpistasday, naovaitergolpe, ForaDilma, ForaTemer, Dilma, Temer, DilmaCoraçãoValente, ForaPT.

Para cada tweet coletado foi extraído um conjunto de atributos relacionado com o tweet. Foram eles:

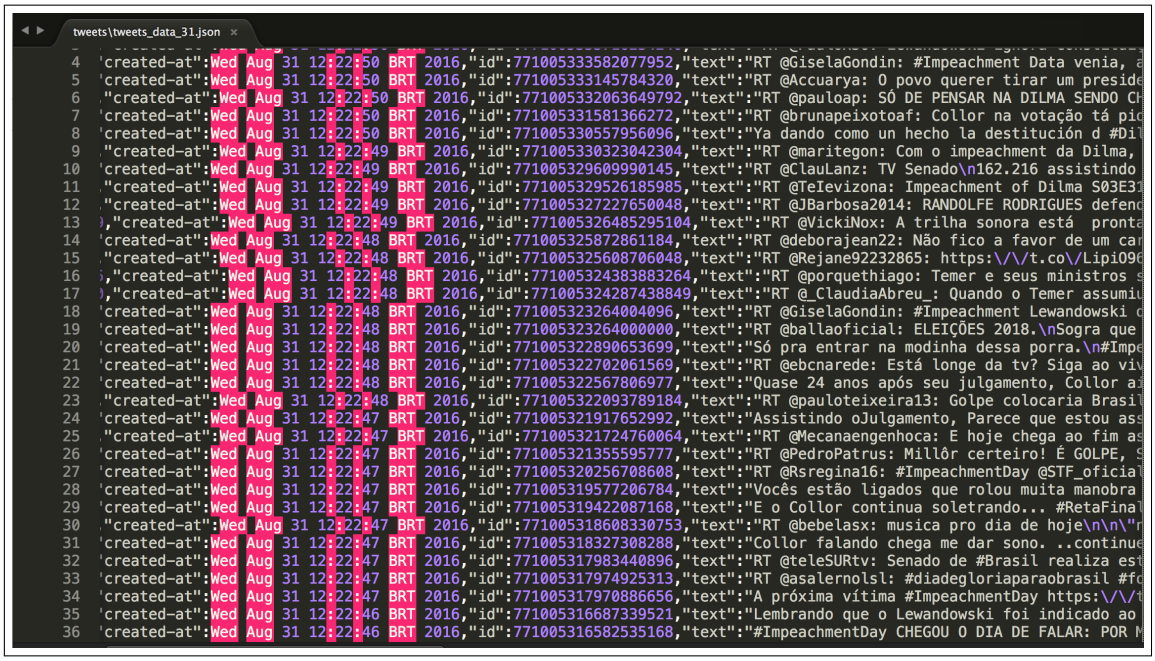
- id;
- text;
- created-at;
- username;
- retweetsNum;
- favourite;
- lang;
- latitude;
- longitude;

O atributo id é um valor inteiro que identifica de forma única um *tweet* no banco de dados do twitter. $'Text'$ é um atributo de texto UTF-8 com a mensagem do *tweet*. O campo

[2] https://github.com/mineracaoUFC
[3] http://www.json.org/

created-at armazena a data e hora da criação do *tweet*, *username* guarda o nome de usuário que criou o *tweet*, *retweetsNum* é o atributo que armazena a quantidade de vezes que esse *tweet* foi foi retuitado, $favourite$ indica aproximadamente a quantidade vezes que um *tweet* foi curtido, *lang* representa a linguagem em que o *tweet* foi escrito e por fim, latitude e longitude armazenam a localização em que o *tweet* foi criado caso o usuário tenha dado permissão para o twitter coletar tal informação.

Figura 3 – Amostra dos *tweets* coletados



Fonte: Elaborado pelo autor

Para o experimento deste trabalho foram apenas utilizados o texto do *tweet* e o id, visto que esses foram as entradas necessárias para os algoritmos de clusterização. Contudo, os demais campos foram coletados visando trabalhos futuros a este estudo, e/ou utilização por outros trabalhos que necessitem de tais informações.

Esta base de dados servirá como entrada para os algoritmos de clusterização na subseção 5.3. Porém, as informações coletadas passarão primeiro pela etapa de pré-processamento, detalhada na subseção 5.2.

5.3 Pré-processamento da base de dados

Esta fase possui a finalidade de processar os dados coletados no passo anterior, a fim de prepara-los para a fase de mineração de dados.

Desta forma, foi realizado o pré-processamento dos textos dos *tweets* coletados. Foram removidos os acentos das palavras, links, caracteres especiais, nomes de usuários. Além disso, foi realizado o processo de retirada das *stopwords* que são palavras que não possuem muito valor semântico para uma análise. Palavras como: os artigos, preposições, conjunções, dentre outros foram removidas.

Por exemplo, uma entrada como: "#Dilma detonou o verbo na @tvsenado #VoltaDemocraciaBrasileira #ForaTemer fraco, fujão de uuuuu", têm como resultado "dilma detonou verbo voltademocraciabrasileira foratemer fraco fujao". Neste exemplo, foram retirados os acentos, os símbolos de *hashtag*, a sequencia de caracteres "uuuuu", vírgulas e o nome de usuário do twitter "@tvsenado".

Foi utilizado o *script* criado por Filho (2014) para a realização do pré-processamento dos dados. O *script* recebe um arquivo em formato .txt, cuja formatação necessária pode ser visualizada no arquivo "README" do projeto disponível no Apêndice A deste trabalho. O arquivo de saída deste *script* é um arquivo em formato .tsv, que contém em cada linha do arquivo um id de um *tweet* e o texto pré-processado, separados por tabulações.

Com a finalização desta etapa, os dados estão prontos para serem minerados pelo algoritmo de clusterização detalhado na próxima Seção.

5.4 Clusterização dos dados com DBSCAN

Nesta etapa, foi realizada a implementação do algoritmo de clusterização DBSCAN com o objetivo de melhor entender o funcionamento do algoritmo, bem como utilizar tal implementação para clusterizar a base de dados.

O desenvolvimento do algoritmo foi realizado utilizando a linguagem de programação Python. Python foi escolhida por ser uma das linguagens de código aberto mais comuns e populares atualmente[4]. Além disso, é de fácil implementação, a manipulação de arquivos e textos é simples e Python faz parte do domínio de conhecimento do autor desse trabalho.

Após a implementação do algoritmo, foram realizados alguns testes para validar se a codificação do DBSCAN estava feita de forma correta. Para tal, foi utilizado a plataforma web "DBSCAN Data Points Plotter"[5]. Esta plataforma gera um conjunto de pontos e mostra o

[4] http://www.codingdojo.com/blog/9-most-in-demand-programming-languages-of-2016/
[5] http://people.cs.nctu.edu.tw/ rsliang/dbscan/index.html

resultado da clusterização com o DBSCAN dada uma entrada de parâmetros. Como: *minPoints*, *eps*, quantidade de *clusters*. Foi realizado um comparativo entre as saídas geradas por essa plataforma e a implementação feita por este trabalho. Comparamos os *clusters* gerados, bem como os *outliers*.

Por fim, após ter certeza da corretude da implementação, foi aplicado o algoritmo implementado aos dados pré-processados. Arquivos .tsv serviram como entrada do algoritmo, sendo aplicado um arquivo como entrada por vez. Cada arquivo contêm *tweets* coletados de apenas um dia. Como resultado, o algoritmo gera um arquivo para cada *cluster* gerado durante a execução contendo os *tweets* que fazem parte do mesmo.

Os *clusters* encontrados nessa etapa de clusterização dos *tweets* serão analisados e melhor explicados na seção posterior.

5.5 Analise dos resultados

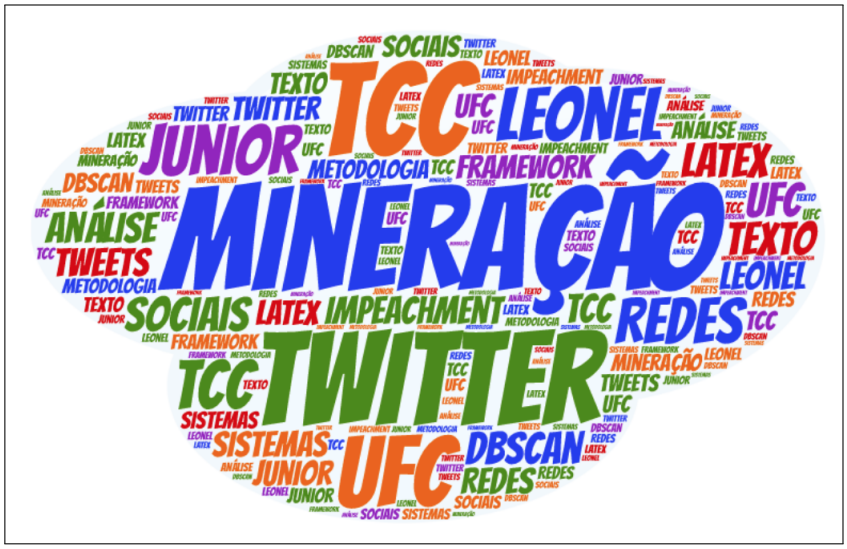
Esta é a etapa final do estudo de caso, onde foi realizada a análise dos *clusters* gerados na seção anterior. Para tal, o autor desse projeto confrontou os *clusters* criados com notícias da época do *impeachment*. Desta forma, é possível validar os resultados do estudo de caso.

Para uma melhor visualização dos resultados, esse trabalho utilizou o auxílio de nuvens de palavras. Foi criada uma nuvem de palavras para cada um dos *clusters* criados pelo processo de clusterização. Desta forma, a visualização das palavras mais utilizadas é feita de forma mais simples. Um exemplo de uma nuvem de palavras pode ser visualizado na Figura 4.

Para a criação das nuvens de palavras, foi utilizada uma ferramenta chamada Tagul[6]. O Tagul foi escolhido por ser de simples utilização e permitir uma grande quantidade de customização das nuvens. Tal como: cor das palavras, formato da nuvem de palavras, fonte das palavras, entre outras opções.

[6] https://tagul.com/

Figura 4 – Exemplo de nuvem de palavras.



Fonte: Elaborado pelo autor

6 RESULTADOS

Esta seção tem como finalidade relatar, detalhadamente, acerca da realização dos procedimentos citados anteriormente.

6.1 Consolidação do framework

O primeiro passo do processo de mineração de textos em redes sociais é a coleta de dados. A implementação do *script* de coleta de *tweets* criado por Filho (2014) foi a escolhida para fazer parte do framework[1]. Essa implementação foi escolhida, pois foi criada em um dos trabalhos relacionados e executa o processo de coleta de modo eficaz. Foi criado um passo a passo de como utilizar esse *script* com o propósito de facilitar a utilização do mesmo. O passo a passo pode ser encontrado no Apêndice A deste trabalho.

Seguindo o processo, o *script* utilizado pelo framework para o pré-processamento de dados que envolve as fases de Processamento de Linguagem Natural foi desenvolvido por Filho (2014). Foi criado também um passo a passo, incluindo alguns exemplos, para ajudar na utilização do *script*. O passo a passo para a execução do *script* de pré-processamento está disponível no Apêndice B.

Para a fase de análise de informações, foi implementado um *script* em linguagem Python do algoritmo DBSCAN. É possível usar como entrada, arquivos em formato JSON ou arquivos ".txt" com dados separados por algum caractere pré-configurado no *script*. Como saída do programa, é gerado um arquivo ".txt" para cada *cluster* criado pelo algoritmo.

Por fim, foi desenvolvida uma aplicação web utilizando o framework JavaScript Angular. Angular foi escolhido por fazer parte do conhecimento dos autores, e por ser de fácil implementação. O layout do site foi criado com base no padrão *materialdesigner*[2], com a intenção de criar uma aplicação agradável aos usuários.

Essa aplicação pode ser dividida em 3 principais funcionalidades. A primeira é a apresentação das nuvens de palavras em *cards* com título e uma pequena descrição. A segunda parte da aplicação é formada por um conjunto de *cards* informativos sobre os dados da pesquisa. Tais como: total de *tweets* coletados, *clusters* criados com a análise, alguns gráficos sobre os dados, entre outras informações. A terceira parte é uma área onde a aplicação permite o download de todos os dados(*tweets*) coletados.

[1] https://github.com/AdailCarvalho/twutils
[2] https://material.google.com/

Na Figura 5, podemos ver a página inicial do aplicativo.

Figura 5 – Sistema para disponibilização dos resultados do estudo de caso.

Fonte: Elaborado pelo autor

6.2 Coleta de dados

A etapa de coleta de informações ocorreu entre os dias 29 de agosto à 18 de setembro de 2016. Foi selecionada a implementação de Filho (2014) para ser o *script* de coleta de dados do Twitter a fazer parte do *framework*. Tal implementação foi escolhida porque realiza a coleta de forma satisfatória. Além disso, é uma aplicação livre e disponível no Github[3]. Contudo, fez-se necessário elaborar a documentação explicando o passo a passo o uso dessa ferramenta em trabalhos futuros. Esse texto foi inserido no arquivo "README.md" do projeto no Github e disponibilizado no Apêndice A deste trabalho. Foi realizado um *fork* da implementação original do Github para o repositório público criado para unir as aplicações que formam o *framework*.

Para a identificação dos *tweets* relacionados com o assunto tratado, o *script* recebeu como entrada uma lista de *hashtags* relacionadas ao assunto.

A lista de *hashtags* foi selecionada de forma manual observando a plataforma web Trends24[4], que disponibiliza a lista com as 10 *hastags* mais comentadas a cada hora. Então, sempre que uma *hashtag* relacionada ao assunto aqui estudado ficava entre as 10 mais

[3] https://github.com/AdailCarvalho/twutils
[4] http://trends24.in

faladas do Brasil, tal *hashtag* entrava na lista utilizada para a coleta dos *tweets* para esse trabalho. Ao final a lista de *hashtags* era composta por: #impeachment, #Golpe, #naovaitergolpe, #vemprarua, #tchauquerida, #golpistasday, #ForaDilma, #ForaTemer, #Dilma, #Temer, #Dilmãe, #PelaDemocracia, #Gilma, #DilmaCoraçãoValente, #ForaPT, #impeachmentDay.

Como resultado dessa coleta, foi obtida uma grande quantidade de *tweets* relacionados ao assunto de interesse deste trabalho, no caso, o *impeachment* da ex-presidente Dilma Rousseff. No Quadro 2, são apresentados os dados finais da coleta.

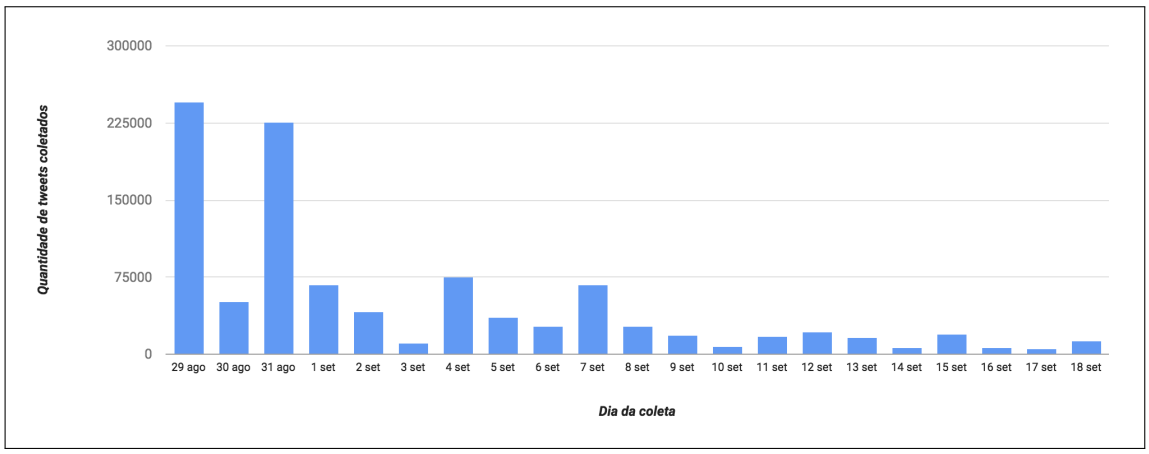
Quadro 2 – Resultado da Coleta de Dados

Dias de coleta	21 dias
Tweets coletados	1.008.290 *tweets*
Tamanho total em disco (arquivos JSON)	413 Mb

Fonte: Elaborado pelo autor

O estudo de caso foi aplicado usando apenas os dados do dia 29 de agosto à 2 de setembro de 2016, pois foi esse o período de maior concentração de *tweets* coletados. Foram coletados 630.975 *tweets* apenas nesses 5 dias. A Figura 6 apresenta a quantidade de *tweets* coletados por dia de coleta.

Figura 6 – *Tweets* coletados por dia



Fonte: Elaborado pelo autor

6.3 Pré-processamento

Nesta fase, foi aplicado o pré-processamento de dados apenas nos *tweets* que foram selecionados para fazer parte do estudo de caso. Foram removidos links, nomes de usuários,

caracteres especiais, acentos e as *stopwords*.

Durante o pré-processamento dos *tweets* foi realizada também a remoção de atributos que não eram necessários para o algoritmo DBSCAN. Portanto, atributos como: longitude, latitude, número de *retweets* foram removidos.

Para essa etapa do desenvolvimento do projeto, foi escolhido o *script* criado por Filho (2014) para fazer parte do framework[5]. Esse algoritmo realiza todos os requisitos do pré-processamento utilizando Processamento de Linguagem Natural (PLN). No entanto, o *script* foi aprimorado para aceitar um número maior de tipos de arquivos de entrada. Até então, a implementação só aceitava, como entrada, arquivos do tipo *JSON*. Devido a este trabalho, é possível executar o *script* usando qualquer arquivo do tipo DSV (delimiter-separated values), separado por um determinado delimitador. Um delimitador pode ser vírgula, espaço, tab, etc. Para tal, o usuário apenas necessita listar os divisores em uma função do *script*.

Para facilitar a utilização, foi criado também um passo a passo de como utilizar o *script*, bem como a exposição de alguns exemplos de arquivos de entrada suportados. Esse passo a passo está dentro do projeto no Github no arquivo "README"e disponível no apêndice B.

6.4 Clusterização

Após a implementação do algoritmo DBSCAN, foi realizada uma validação da implementação utilizando a plataforma web "DBSCAN Data Points Plotter"[6]. Tal plataforma gera um conjunto de coordenadas para a entrada no DBSCAN, bem como o resultado após a aplicação do algoritmo. A plataforma recebe como entrada um valor para *eps*, *minPoints*, número de *clusters*, quantidade de *outiliers* e o espaço mínimo entre os *clusters*. Após isso, comparamos *clusters* e os *outliers* gerados pela plataforma com os resultados obtidos pela implementação do DBSCAN no *framework* proposto neste trabalho, garantindo assim a corretude da implementação do algoritmo de mineração.

Para a realização do estudo de caso foi utilizado um parâmetro *minPts* com o valor de 500, ou seja, um *tweet* só será considerado *core* se tiver pelo menos 500 *tweets* similares. Foi usado um $Eps = 0{,}3$ para medir a similaridade. Isso implica dizer que um *tweet* precisa ser pelo menos 30% igual a outro *tweet* para serem considerados similares.

O quadro 3 mostra os resultados de forma geral.

[5] https://github.com/AdailCarvalho/textprocessor
[6] http://people.cs.nctu.edu.tw/ rsliang/dbscan/index.html

Quadro 3 – Resultados da clusterização

Dia	Qtd clusters	Tweets nos clusters	outliers
29 de Agosto	1	77.251	168.370
30 de agosto	3	8.105	27.607
31 de agosto	3	74.572	150.713
1 de setembro	4	32.914	34.590
2 de setembro	4	19.877	27.872

Fonte: Elaborado pelo autor

Neste estudo de caso, foi utilizada a similaridade de Jaccard, conforme já foi afirmado, apesar de o *framework* oferecer suporte a qualquer função de similaridade na implementação do DBSCAN. As medidas de similaridade de Jacard, Fading e Euclidiana já foram implementadas.

Na próxima seção será explicado de forma mais detalhada cada um dos *clusters* gerados nesta etapa. Para facilitar a visualização dos *clusters*, serão usadas nuvens de palavras.

6.5 Análise dos dados

Após a clusterização dos *tweets* realizada no passo anterior, fez-se necessária a realização de uma análise dos *clusters* gerados. Através de uma análise subjetiva com base em notícias e fontes informativas, a existência dos *clusters* foi justificada.

Para os dados coletados no dia 29 de agosto de 2016, foi criado apenas um cluster. As palavras mais citadas nos tweets foram "PelaDemocracia", "Golpe", "ForaTemer" e "Impeachment". Apesar das palavras "PelaDemocracia" e "Impeachment" serem palavras imparciais em relação ao *impeachment*, fica evidenciado o apoio à ex-presidente Dilma com as palavras "Golpe" e "ForaTemer". A frequência do aparecimento dessas palavras no dia em questão é justificada pelo fato de este ter sido o dia em que a ex-presidente discursou no Senado Federal[7]. Além disso, foram realizados protestos contra o *impeachment* em onze estados e no Distrito Federal[8]. A nuvem de palavras criada pelos dados do *cluster* do dia 29 de agosto na Figura 7.

Já no dia 30 de agosto de 2016, foram gerados 3 *clusters*. O primeiro deles contém as palavras "Impeachment", "Paschoal", "Janaina" e "Gigante" como as palavras que mais se repetiram dentro do *cluster*. É válido destacar aqui a aparição das palavras "Paschoal"e "Janaina",

[7] http://g1.globo.com/jornal-nacional/noticia/2016/08/dilma-se-defende-pessoalmente-em-longa-sessao-no-senado.html

[8] http://g1.globo.com/jornal-nacional/noticia/2016/08/onze-estados-e-o-df-tem-protestos-contra-o-impeachment.html

Figura 7 – *Cluster* do dia 29 de agosto de 2016

Fonte: Elaborado pelo autor

pois fazem menção a professora de Direito Constitucional Janaína Paschoal. Janaína fez parte da acusação à ex presidente Dilma e foi a primeira a falar no debate entre acusação e defesa no dia 30[9]. Na Figura 8 é possível ver a nuvem de palavras do primeiro *cluster* do dia 30 de agosto.

Já no segundo *cluster*, criado com os *tweets* do dia 30 de agosto, Além da palavra "Paschoal", podemos notar o aparecimento de duas novas palavras em destaque: "Gleisi"e "Hofman", que fazem referência a Senadora Gleisi Hofman. A senadora Gleisi Hofman apareceu em destaque, pois fez parte da defesa da ex presidente Dilma Rousseff e realizou um discurso no dia 30[10]. A segunda nuvem de palavras do dia 30 de agosto é apresentada na Figura 9.

O último *cluster* gerado no dia 30 de Agosto é composto apenas por uma mensagem que foi "retweetada"diversas vezes. O estudo de caso realizado neste trabalho considera um *re-tweet* como uma mensagem original. O texto do *tweet* original é "dilma chegando no senado ao som de *crazy in love appreciation* video foratemer", esse *tweet* foi postado em conjunto com um vídeo e obteve bastante repercussão[11]. A Figura 10 mostra a nuvem de palavras desse *cluster*.

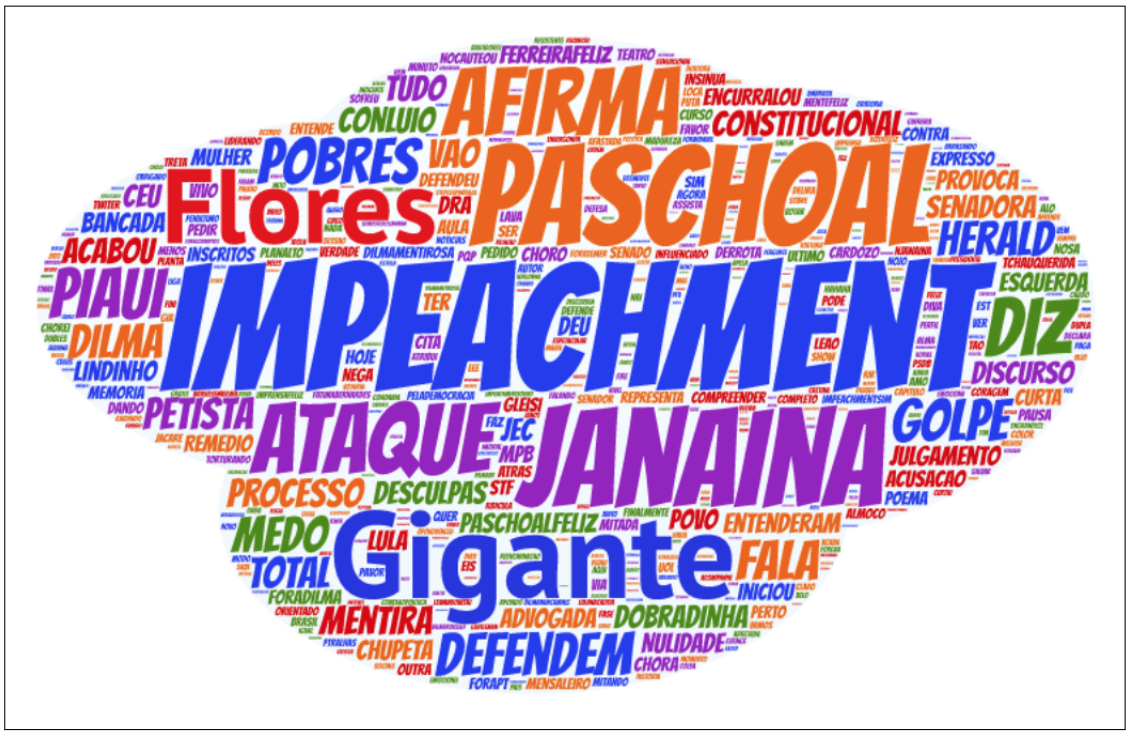
Para os dados coletados no dia 31 de agosto de 2016 foram gerados 3 *clusters*.

[9] http://g1.globo.com/jornal-nacional/noticia/2016/08/acusacao-apresenta-argumentos-finais-no-julgamento-do-impeachment.html

[10] http://g1.globo.com/politica/processo-de-impeachment-de-dilma/noticia/2016/08/impeachment-no-senado-discurso-final-de-gleisi-hoffmann-pt-pr.html

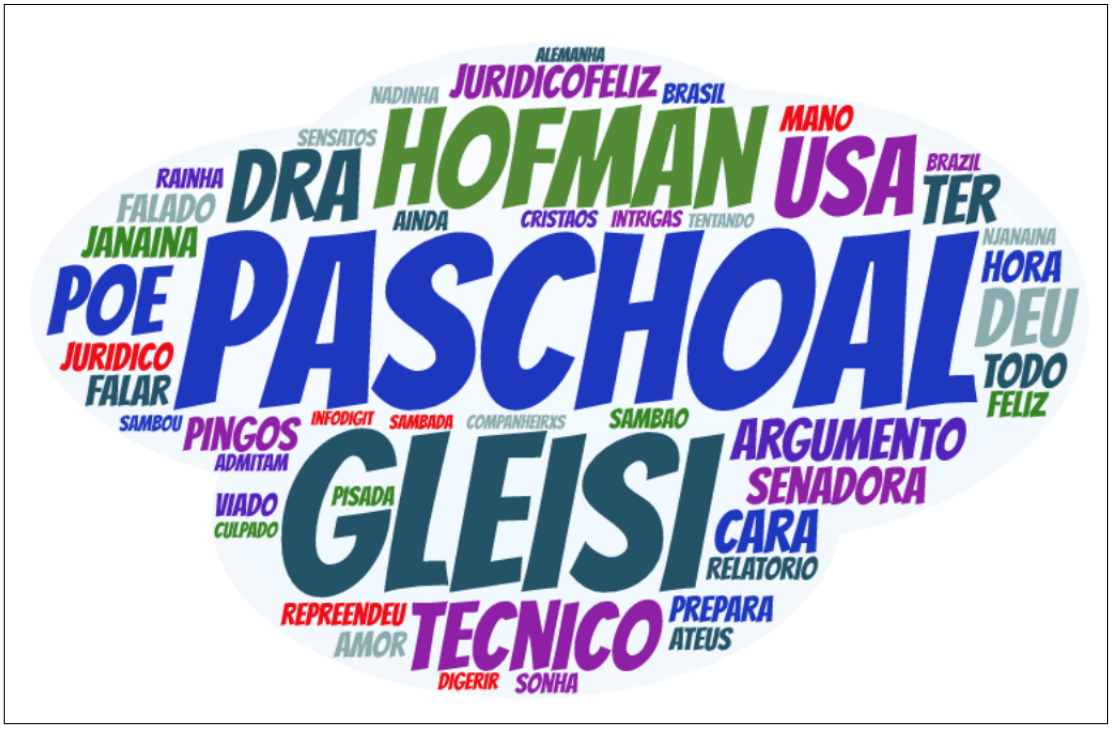
[11] https://mobile.twitter.com/imdressinupfor/status/770255984589672449

Figura 8 – Primeiro *cluster* do dia 30 de agosto de 2016



Fonte: Elaborado pelo autor

Figura 9 – Segundo *cluster* do dia 30 de agosto de 2016



Fonte: Elaborado pelo autor

Figura 10 – Terceiro *cluster* do dia 30 de agosto de 2016

Fonte: Elaborado pelo autor

O primeiro *cluster* criado possui as palavras "ImpeachmentDay", "ForaTemer", "Tchau" e "Querida" como palavras mais mencionadas. "ImpeachmentDay" porque dia 31 de agosto de 2016 foi o último dia no processo do *impeachment* presidencial. Podemos destacar que neste *cluster* existem palavras de apoio à ex presidente Dilma Rousseff, como por exemplo: "ForaTemer". Porém, as palavras "Tchau" e "Querida" foram comumente utilizadas por pessoas favoráveis ao *impeachment*, visto que a frase "Tchau Querida" foi utilizada pelo Deputado Federal Eduardo Cunha quando a ex presidente Dilma foi afastada do cargo[12]. Veja a Figura 11 para visualizar a nuvem de palavras desse primeiro *cluster* encontrado no dia 31 de Agosto.

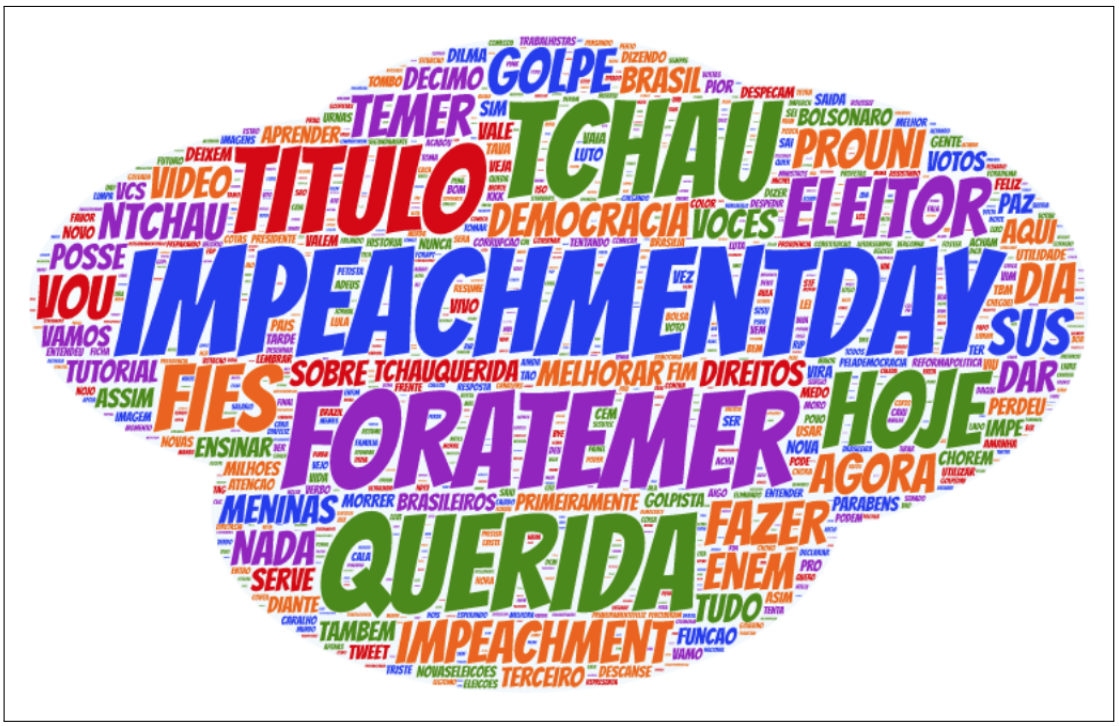
O segundo *cluster* criado foi formado por centenas de *retweets*. O *tweet* original é formado pelo seguinte texto: "Duas rupturas democráticas na História. ImpeachmentDay"[13]. Além do texto, esse *tweet* possui uma imagem que compara a posse do presidente Temer em 2016 à posse do general Castelo Branco em 1964, fazendo menção ao golpe militar de 1964[14]. Por conta disso, muitas pessoas contra o *impeachment* compartilharam esse *tweet*. A Figura 12 apresenta a nuvem de palavras deste *cluster*.

[12] http://zh.clicrbs.com.br/rs/noticias/politica/noticia/2016/05/cunha-rebate-entrevista-de-dilma-apenas-uma-frase-tchau-querida-5820089.html

[13] https://twitter.com/jandira$_feghali/status/770980550907686912$

[14] http://www.infoescola.com/historia/golpe-militar-de-1964/

Figura 11 – Primeiro *cluster* do dia 31 de agosto de 2016



Fonte: Elaborado pelo autor

Figura 12 – Segundo *cluster* do dia 31 de agosto de 2016

Fonte: Elaborado pelo autor

Semelhante ao *cluster* anterior, o terceiro *cluster* do dia 31 também foi formado apenas por *retweets*. Esse *tweet* foi muito compartilhado por pessoas contrárias ao *impeachment*, pois o texto do *tweet* original provoca usuários do PROUNI e FIES que eram a favor do *impeachment*[15]. A nuvem de palavras pode ser vista na Figura 13.

Figura 13 – Terceiro *cluster* do dia 31 de agosto de 2016

Fonte: Elaborado pelo autor

A clusterização dos *tweets* coletados no dia 1 de setembro de 2016, um dia após o *impeachment* da ex presidente Dilma Rousseff, gerou 4 *clusters* distintos. O primeiro deles obteve a maior quantidade de *tweets* e em sua grande maioria de *tweets* contra o *impeachment*. Palavras como "ForaTemer", "Golpista" e "Golpe" estão entre as mais comuns do *cluster*. Isso se espelha nas ruas, onde protestos ocorreram em pelo menos 6 capitais[16]. A nuvem de palavras desse *cluster* é mostrado na Figura 14

O segundo e terceiro *clusters* gerados nesse dia se contrastam bastante. O *cluster* representado pela Figura 15 foi criado a partir de *retweets* de um texto postado por Adolfo Pérez Esquivel[17], ganhador do Prêmio Nobel da paz em 1980, em apoio à Dilma. Contudo, o terceiro

15 https://twitter.com/pilllowzjm/status/771058833485619202
16 http://g1.globo.com/jornal-nacional/noticia/2016/09/estudante-sofre-lesao-no-olho-em-protesto-contra-impeachment-em-sp.html
17 https://twitter.com/prensapesquivel/status/771042540715991040

Figura 14 – Primeiro *cluster* do dia 1 de setembro de 2016

Fonte: Elaborado pelo autor

cluster, representado pela Figura 16, é claramente de apoio ao *impeachment*, pois a palavra mais usada foi "Tchau Querida".

O quarto *cluster* gerado no dia 1 de setembro contém uma característica interessante. Ele foi criado a partir da repercussão nas mídias de uma matéria sobre um juiz que ao expedir um alvará de soltura de um camelô em Belo Horizonte, protestou contra o *impeachment*[18]. Em seu alvará de soltura, o Juiz escreveu “...os bandidos deste País, que deveriam estar presos, estão soltos dando golpe na Democracia”. Consequentemente, várias pessoas compartilharam *tweets* sobre o assunto. A Figura 17 representa esse *cluster*.

O último dia analisado foi o dia 2 de setembro. Os dados coletados nesse dia resultaram na geração de 4 *clusters*. Observando os 4 *clusters* pode-se destacar que todos são contrários ao mandato do então presidente Temer. Os *clusters* representados pelas Figura 18 e Figura 19 mostram de forma clara a rejeição ao então presidente, pois as palavras "ForaTemer", "Golpista", "TemerGolpista" e "Golpe" estão entre as palavras mais mencionadas nesses *clusters*.

Não é coincidência que todos os *clusters* mostram um sentimento que vai contra o *impeachment*. Esse fato é apenas um reflexo dos acontecimentos do dia. A população de 10

[18] http://politica.estadao.com.br/blogs/coluna-do-estadao/juiz-protesta-contra-impeachment-de-dilma-em-alvara-de-soltura/

Figura 15 – Segundo *cluster* do dia 1 de setembro de 2016

Fonte: Elaborado pelo autor

Figura 16 – Terceiro *cluster* do dia 1 de setembro de 2016

Fonte: Elaborado pelo autor

Figura 17 – Quarto *cluster* do dia 1 de setembro de 2016

Fonte: Elaborado pelo autor

estados brasileiros realizou protestos contra o então presidente Michel Temer[19]. Além disso, os Senadores votaram a favor da flexibilização de créditos suplementares sem autorização do Congresso Nacional[20]. Essa atitude do Senado repercutiu de forma bastante negativa nas redes sociais para o governo em exercício.

Já o *cluster* representado pela Figura 20 é formado apenas por *retweets* de uma postagem que discorda da afirmativa de que todos que desaprovavam o governo Temer eram supostamente petistas[21]. O texto do *tweet* original é "Ta engraçado quem chama os ForaTemer de petista. Migos, Dilma ja foi. Agora eh soh Fora Temer mesmo. Lembram? 1o ela depois o resto?".

O último *cluster* gerado com dados coletado desse dia foi formado especificamente por *tweets* que tratavam do assunto sobre a sanção das pedaladas fiscais pelo governo Temer, apenas dois dias após o *impeachment*[22]. Dois dias após *impeachment*, governo Temer sanciona lei que autoriza pedaladas fiscais. A Figura 21 apresenta a nuvem de palavras desse *cluster*.

19 http://g1.globo.com/jornal-nacional/edicoes/2016/09/02.html
20 http://economia.ig.com.br/2016-09-02/lei-orcamento.html
21 https://twitter.com/urgh/status/771506400660971521
22 https://www.brasildefato.com.br/2016/09/02/dois-dias-apos-golpe-governo-temer-sanciona-lei-que-autoriza-pedaladas-fiscais/

Figura 18 – Primeiro *cluster* do dia 2 de setembro de 2016

Fonte: Elaborado pelo autor

Figura 19 – Segundo *cluster* do dia 2 de setembro de 2016

Fonte: Elaborado pelo autor

Figura 20 – Terceiro *cluster* do dia 2 de setembro de 2016

Fonte: Elaborado pelo autor

Figura 21 – Quarto *cluster* do dia 2 de setembro de 2016

Fonte: Elaborado pelo autor

7 CONSIDERAÇÕES FINAIS

Este trabalho investigou o processo de Mineração de Textos em redes sociais em trabalhos recentes realizados no Campus em Quixadá da Universidade Federal do Ceará. Bem como, identificou grandes semelhanças nos procedimentos metodológicos entre todos os trabalhos estudados.

Dessa forma, levando em consideração o retrabalho necessário para replicar o procedimento metodológico, este trabalho criou um framework que possui ferramentas que buscam auxiliar em cada um dos passos do processo. Estas etapas do procedimento metodológico constituem desde a coleta utilizando a API do Twitter, pré-processamento utilizando PLN, análise com o algoritmo DBSCAN, até a disponibilização do estudo de forma ampla através de um sistema web.

Foi realizado ainda um estudo de caso visando experimentar o framework proposto. O estudo de caso foi realizado utilizando como base os dados do Twitter referentes ao processo de *impeachment* da ex-presidente Dilma Rousseff. Após a coleta, pré-processamento, e aplicação do algoritmo de mineração de dados, foi realizada uma análise dos resultados obtidos utilizando nuvens de palavras. Por fim, a disponibilização do estudo para a comunidade foi realizada através de um sistema web incluso no framework.

Como trabalhos futuros, a intenção é que o framework seja expandido, consequentemente, útil para mais pesquisadores. Para cada passo do processo de mineração pode ser adicionado melhorias. Discutimos algumas possibilidades nos parágrafos seguintes.

Em relação a coleta de dados, seria interessante que o framework fosse capaz de realizar a coleta em diferentes redes sociais. Tais como, Facebook, Linkedin, Googgle+, entre outros. A criação de uma interface gráfica para o *script* existente também é uma futura contribuição para este trabalho.

Para a fase de pré-processamento, podem ser adicionadas outras técnicas para limpeza e organização dos dados. Bem como, a criação de uma interface gráfica para facilitar a utilização do *script*.

A fase de análise é onde existe mais oportunidades de trabalhos futuros. É interessante adicionar novas medidas de similaridades, além das já existentes no framework. Adicionalmente, é importante incluir outros algoritmos de Mineração de Dados, expandindo assim o campo de utilização do framework.

O sistema web disponibilizado para o compartilhamento das pesquisas deve ser

expandido. O mesmo deve possuir outras funções, tais como: avaliação dos *outliers*, identificação da evolução de *clusters*, geração automática de nuvens de palavras, entre outros. Tudo isso com o objetivo de melhor apresentar resultados de outros tipos de pesquisa. Além disso, outras formas de compartilhamento do conhecimento podem ser consideradas.

Além disso, os dados coletados sobre o *impeachment* da ex-presidente Dilma Rousseff podem ser analisados por outras técnicas e/ou utilizados por outros pesquisadores. Tais dados estão disponíveis no sistema web.

Em relação ao estudo de caso, é importante realizar uma análise mais crítica dos *clusters* encontrados após a aplicação do DBSCAN. Além disso, é interessante aplicar o DBSCAN com diferentes valores para os parâmetros *eps* e *minPoints*, com o objetivo de diminuir a quantidade de *outliers* encontrados no estudo. Dessa forma, novas informações relevantes podem ser extraídas desses dados.

Vale ressaltar, que o desejo principal deste trabalho foi contribuir para o desenvolvimento das pesquisas relacionadas à Mineração de Dados em redes sociais. O desejo é que os pesquisadores usem menos tempo nas fases intermediárias da pesquisa, podendo assim, focar mais tempo nas fases de análise e interpretação dos dados.

REFERÊNCIAS

AGGARWAL, C. C.; ZHAI, C. **Mining text data**. [S.l.]: Springer Science & Business Media, 2012.

ARANHA, C. N.; VELLASCO, M.; PASSOS, E. 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, RJ**, 2007.

BARROSO, L. P.; ARTES, R. Análise multivariada. **Lavras: Ufla**, 2003.

ESTER, M.; KRIEGEL, H.-P.; SANDER, J.; XU, X. et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: **Kdd**. [S.l.: s.n.], 1996. v. 96, n. 34, p. 226–231.

FILHO, J. A. C. **Mineração de textos: análise de sentimento utilizando tweets referentes à copa do mundo 2014**. TCC (Graduação em Sistemas de Informação) - Universidade Federal do Ceará, 2014.

JOHNSON, R. E.; RUSSO, V. **Reusing object-oriented designs**. [S.l.]: Department of Computer Science, University of Illinois at Urbana-Champaign, 1991.

LEE, P.; LAKSHMANAN, L. V.; MILIOS, E. E. Incremental cluster evolution tracking from highly dynamic network data. In: IEEE. **2014 IEEE 30th International Conference on Data Engineering**. [S.l.], 2014. p. 3–14.

LEITE, J. L. A. **Mineração de textos do twitter utilizando técnicas de classificação**. TCC (Graduação em Sistemas de Informação) - Universidade Federal do Ceará, 2016.

LEONEL JUNIOR, R. d. A.; JUNIOR, J. H. F.; JORGE da S. T.; SILVA, T. L. da; MAGALHãES, R. P. **Mineração em Dados Abertos In: IV Jornada Científica de Sistemas de Informação**. [S.l.]: Parnaíba, PI, 2014.

MATHIAK, B.; ECKSTEIN, S. Five steps to text mining in biomedical literature. In: **Proceedings of the second European workshop on data mining and text mining in bioinformatics**. [S.l.: s.n.], 2004. p. 43–46.

MATTSSON, M.; BOSCH, J. Stability assessment of evolving industrial object-oriented frameworks. **Journal of Software Maintenance: Research and Practice**, Wiley Online Library, v. 12, n. 2, p. 79–102, 2000.

RECUERO, R.; ZAGO, G. Em busca das "redes que importam": redes sociais e capital social no twitter. **LÍBERO. ISSN impresso: 1517-3283/ISSN online: 2525-3166**, n. 24, p. 81–94, 2016.

RODRIGUES, P. R. F. **Dinamica de temas abordados no twitter via evoluca de clusters**. TCC (Graduação em Sistemas de Informação) - Universidade Federal do Ceará, 2016.

RUSSELL, M. A. **Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Google+, GitHub, and More**. [S.l.]: "O'Reilly Media, Inc.", 2013.

SILVA, T. L. da; SOUSA, F. R.; MACÊDO, J. A. F. de; MACHADO, J. C.; CAVALCANTE, A. A. **Análise em Big Data e um Estudo de Caso utilizando Ambientes de Computação em Nuvem**. [S.l.]: Quixadá, 2013.

SIMOS, G. C. **How Much Data Is Generated Every Minute On Social Media?** 2015. Disponível em: <http://wersm.com/how-much-data-is-generated-every-minute-on-social-media/>.

TAN, A.-H. et al. Text mining: The state of the art and the challenges. In: **Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases**. [S.l.: s.n.], 1999. v. 8, p. 65–70.

TAN, P.-N. et al. **Introduction to data mining**. [S.l.]: Pearson Education India, 2006.

VIANA, Z. L. **Mineração de textos: análise de sentimento utilizando tweets referentes às eleições presidenciais 2014**. TCC (Graduação em Sistemas de Informação) - Universidade Federal do Ceará, 2014.

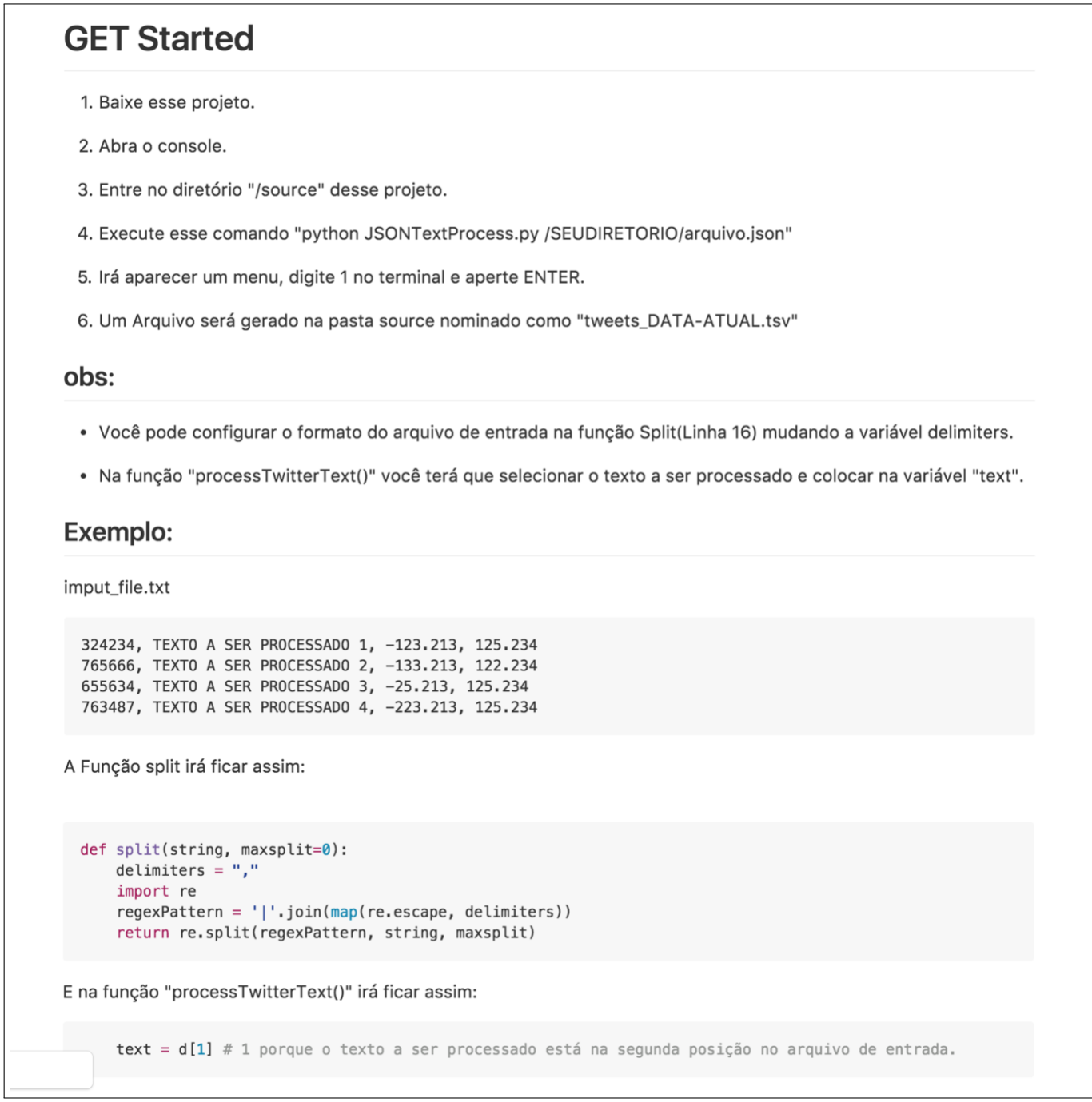
WU, F.-x. Genetic weighted k-means algorithm for clustering large-scale gene expression data. **BMC bioinformatics**, BioMed Central, v. 9, n. 6, p. 1, 2008.

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

YIN, J.; LAMPERT, A.; CAMERON, M.; ROBINSON, B.; POWER, R. Using social media to enhance emergency situation awareness. **IEEE Intelligent Systems**, v. 27, n. 6, p. 52–59, 2012.

APÊNDICE A – GET STARTED DO SCRIPT DE COLETA DE DADOS DO TWITTER

Figura 22 – *Get Started* do algoritmo de coleta de dados do Twitter.



GET Started

1. Baixe esse projeto.
2. Abra o console.
3. Entre no diretório "/source" desse projeto.
4. Execute esse comando "python JSONTextProcess.py /SEUDIRETORIO/arquivo.json"
5. Irá aparecer um menu, digite 1 no terminal e aperte ENTER.
6. Um Arquivo será gerado na pasta source nominado como "tweets_DATA-ATUAL.tsv"

obs:

- Você pode configurar o formato do arquivo de entrada na função Split(Linha 16) mudando a variável delimiters.
- Na função "processTwitterText()" você terá que selecionar o texto a ser processado e colocar na variável "text".

Exemplo:

imput_file.txt

```
324234, TEXTO A SER PROCESSADO 1, -123.213, 125.234
765666, TEXTO A SER PROCESSADO 2, -133.213, 122.234
655634, TEXTO A SER PROCESSADO 3, -25.213, 125.234
763487, TEXTO A SER PROCESSADO 4, -223.213, 125.234
```

A Função split irá ficar assim:

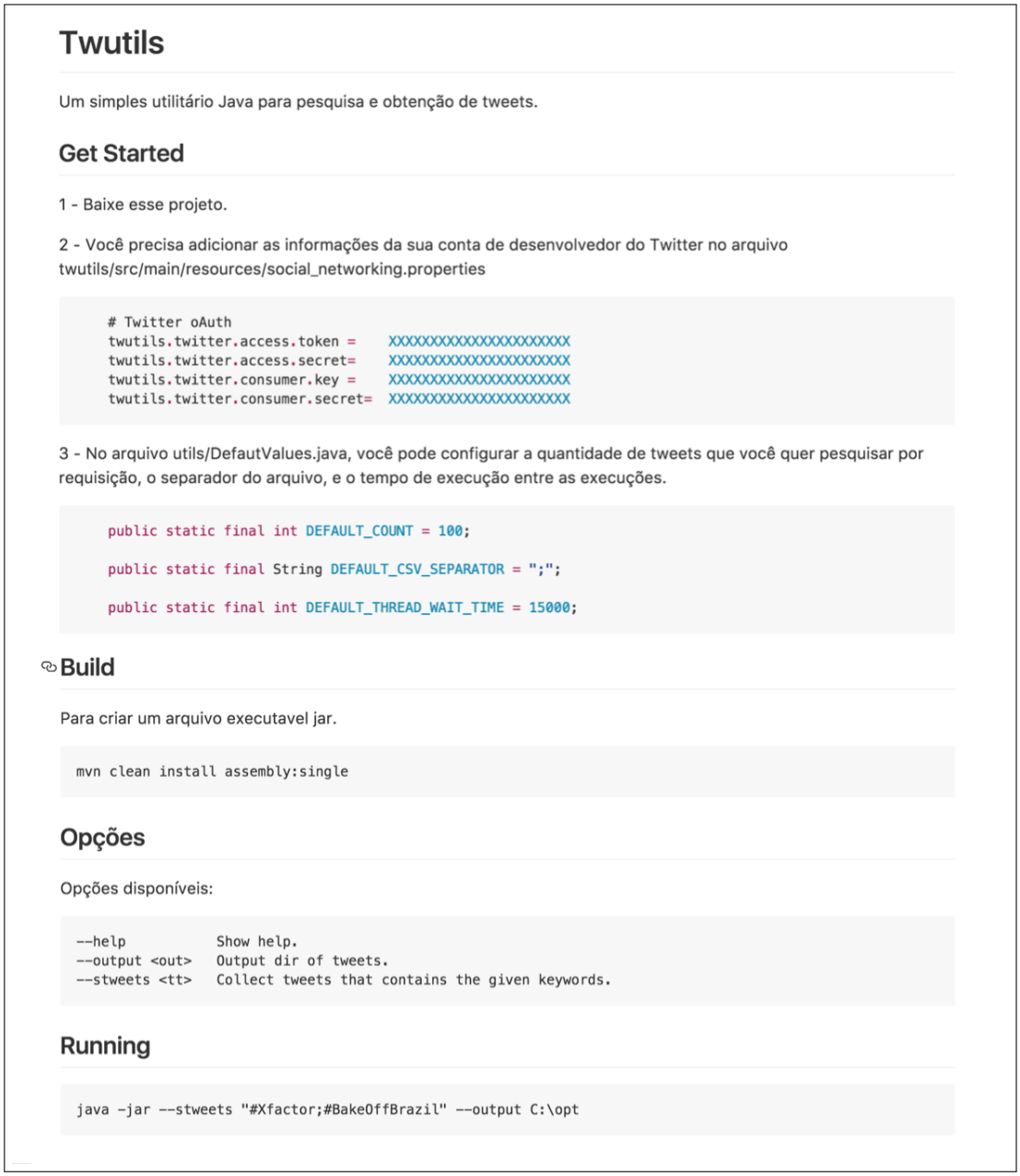
```
def split(string, maxsplit=0):
    delimiters = ","
    import re
    regexPattern = '|'.join(map(re.escape, delimiters))
    return re.split(regexPattern, string, maxsplit)
```

E na função "processTwitterText()" irá ficar assim:

```
    text = d[1] # 1 porque o texto a ser processado está na segunda posição no arquivo de entrada.
```

APÊNDICE B – GET STARTED DO SCRIPT DE PRÉ-PROCESSAMENTO DOS DADOS

Figura 23 – *Get Started* do algoritmo de pré-processamento dos dados.



Twutils

Um simples utilitário Java para pesquisa e obtenção de tweets.

Get Started

1 - Baixe esse projeto.

2 - Você precisa adicionar as informações da sua conta de desenvolvedor do Twitter no arquivo twutils/src/main/resources/social_networking.properties

```
# Twitter oAuth
twutils.twitter.access.token =    XXXXXXXXXXXXXXXXXXXXX
twutils.twitter.access.secret=    XXXXXXXXXXXXXXXXXXXXX
twutils.twitter.consumer.key =    XXXXXXXXXXXXXXXXXXXXX
twutils.twitter.consumer.secret=  XXXXXXXXXXXXXXXXXXXXX
```

3 - No arquivo utils/DefautValues.java, você pode configurar a quantidade de tweets que você quer pesquisar por requisição, o separador do arquivo, e o tempo de execução entre as execuções.

```
public static final int DEFAULT_COUNT = 100;

public static final String DEFAULT_CSV_SEPARATOR = ";";

public static final int DEFAULT_THREAD_WAIT_TIME = 15000;
```

Build

Para criar um arquivo executavel jar.

```
mvn clean install assembly:single
```

Opções

Opções disponíveis:

```
--help           Show help.
--output <out>   Output dir of tweets.
--stweets <tt>   Collect tweets that contains the given keywords.
```

Running

```
java -jar --stweets "#Xfactor;#BakeOffBrazil" --output C:\opt
```

APÊNDICE C – GET STARTED DO SCRIPT DO DBSCAN

Figura 24 – *Get Started* do *script* de implementação de DBSCAN.

DBScan

Implementação do algoritmo de mineração de dados DBScan. Incluindo 3 medidas de similaridade: Euclidiana, Jaccard e Fading.

Get started

1 - Baixe este projeto.

2 - Coloque o caminho do seu arquivo de entrada.

```
    with open('tweets_30.tsv') as json_data:
```

3 - Escolha o formato do seu arquivo de entrada(JSON ou algum arquivo .txt dividido por algum caracter)

```
    # A linha abaixo le um arquivo em formato JSON
    # points = json.load(json_data)

    # O codigo abaixo le um arquivo txt(tabulado)
    points = {}
    points['tweets'] = []
    for line in json_data:
        data = line.split('\t')
```

4 - Configure os parâmetros do algoritmo (eps e minPts)

```
    # A funcao DBSCAN recebe um array de pontos, eps e minPoints.
    dbScan(points['tweets'], 0.3, 500)
```

5 - Coloque o caminho do seu arquivo de saída.

```
    # Caminho do arquivo de saida dos clusters
    output_file = open('/Users/CAMINHO/DO/SEU/ARQUIVO/file.txt','a')
```

6 - Salve as edições.

7 - Abra o console e execute "python DBSCAN.py"

Observações

- O código está todo comentádo, a fim de facilitar a edição do mesmo.
- Esse projeto possui 3 implementações de medida de similaridade. Euclidiana, Fading e de Jaccard.